University of Málaga

Doctoral Thesis

Context-Guided Computational Methods for the Consensus Inference of Gene Regulatory Networks and the Detection of Co-expression Patterns

Author

Adrián Segura Ortiz

Supervisor

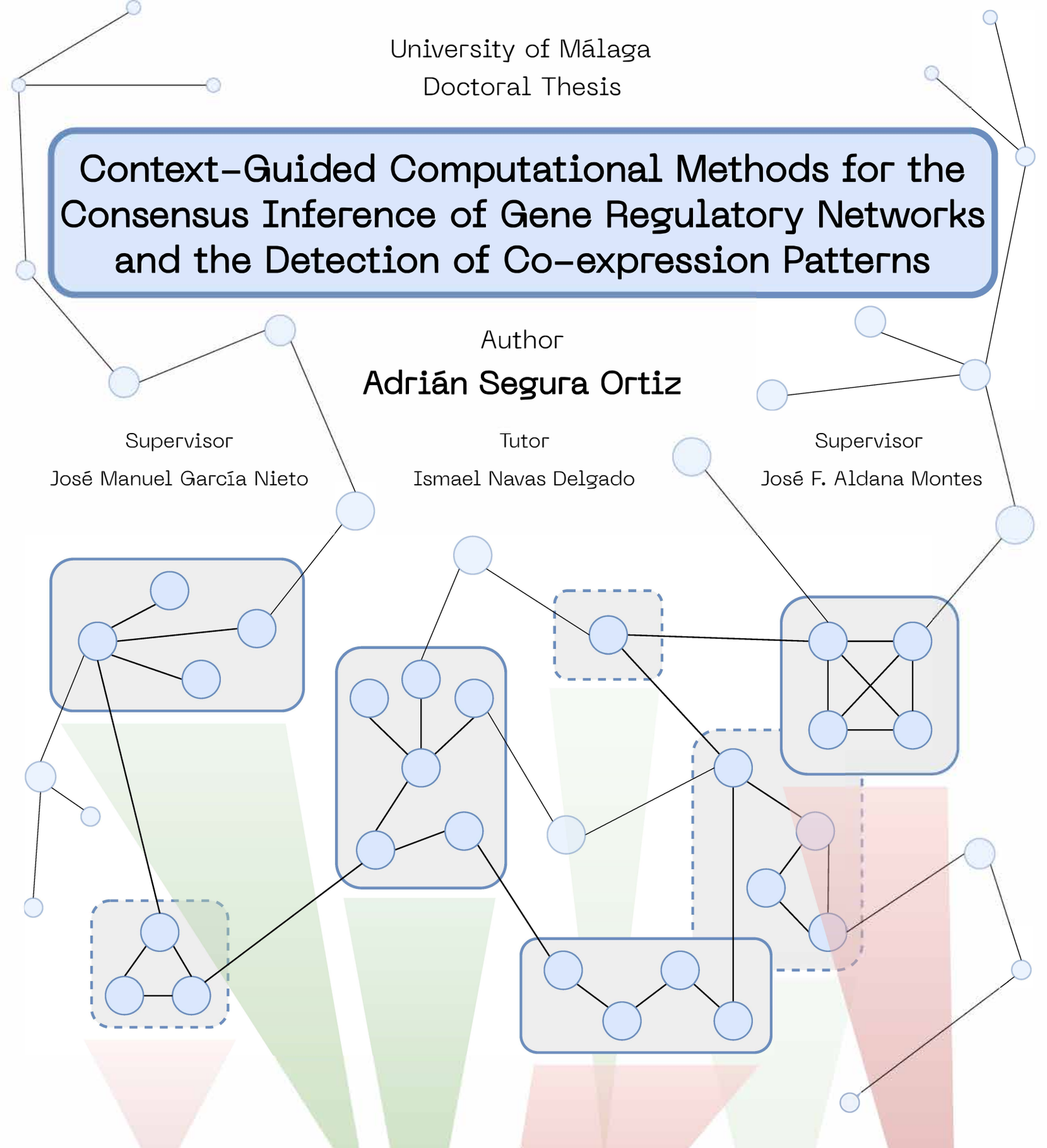
José Manuel García Nieto

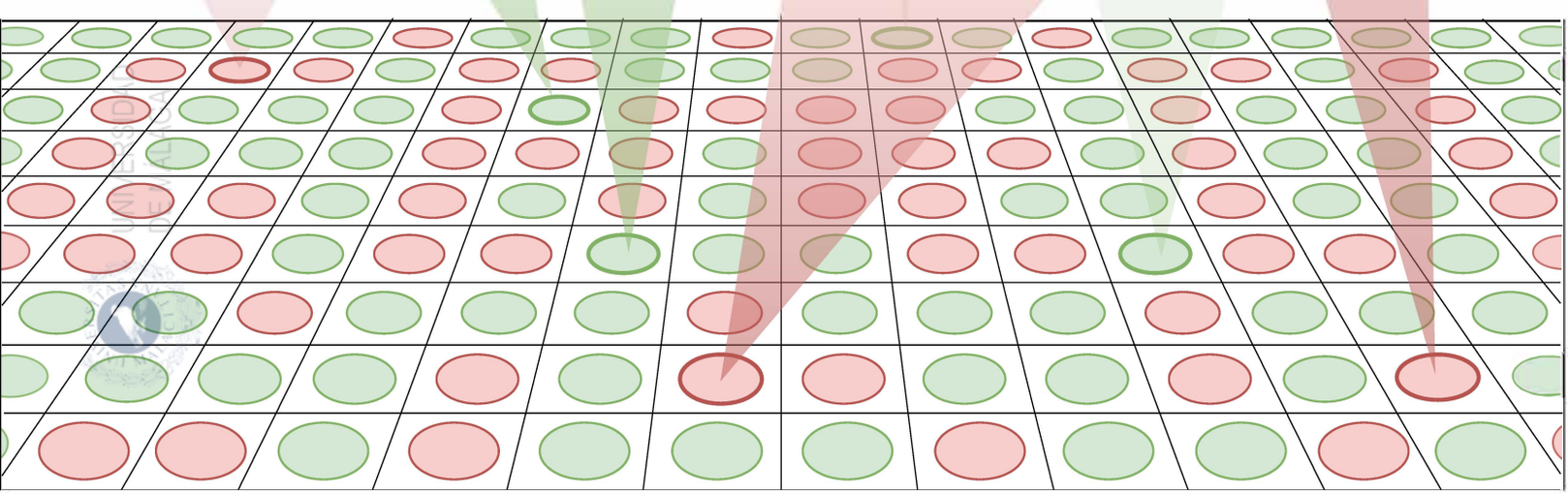
Tutor

Ismael Navas Delgado

Supervisor

José F. Aldana Montes





UNIVERSIDAD DE MÁLAGA

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

DOCTORAL THESIS
COMPUTER TECHNOLOGIES

Context-Guided Computational Methods for the Consensus Inference of Gene Regulatory Networks and the Detection of Co-expression Patterns

E.T.S.I. Informática
R.D. 99/2011

Author

Adrián Segura Ortiz
Khaos Research Group
ITIS Software
University of Málaga

Supervisor

Dr. José Manuel García Nieto
ITIS Software
Department of Computer Science and Programming Languages
University of Málaga

Tutor

Dr. Ismael Navas Delgado
ITIS Software
Department of Computer Science and Programming Languages
University of Málaga

Supervisor

Dr. José Francisco Aldana Montes
ITIS Software
Department of Computer Science and Programming Languages
University of Málaga

July, 2025

AUTOR: Adrián Segura Ortiz

http://orcid.org/0000-0003-2149-5754

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

Context-Guided Computational Methods for the Consensus Inference of Gene Regulatory Networks and the Detection of Co-expression Patterns

First hardcover edition July 2025, self-published.

Front and back cover design by Adrián Segura Ortiz.

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D./Dña ADRIÁN SEGURA ORTIZ
Estudiante del programa de doctorado TECNOLOGÍAS INFORMÁTICAS de la Universidad de Málaga, autor/a de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: CONTEXT-GUIDED COMPUTATIONAL METHODS FOR THE CONSENSUS INFERENCE OF GENE REGULATORY NETWORKS AND THE DETECTION OF CO-EXPRESSION PATTERNS.

Realizada bajo la tutorización de ISMAEL NAVAS DELGADO y dirección de JOSÉ FRANCISCO ALDANA MONTES Y JOSÉ MANUEL GARCÍA NIETO (si tuviera varios directores deberá hacer constar el nombre de todos)

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 25 de JULIO de 2025

Fdo.: ADRIÁN SEGURA ORTIZ Doctorando/a	Fdo.: ISMAEL NAVAS DELGADO Tutor/a

Edificio Pabellón de Gobierno. Campus El Ejido. 29071
Tel.: 952 13 10 28 / 952 13 14 61 / 952 13 71 10
E-mail: doctorado@uma.es

Fdo.: JOSÉ FRANCISCO ALDANA MONTES Y JOSÉ MANUEL GARCÍA NIETO
Director/es de tesis

DECLARACIÓN DE DIRECCIÓN Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D. JOSÉ FRANCISCO ALDANA MONTES y D. JOSÉ MANUEL GARCÍA NIETO, profesores doctores del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga

DECLARAN QUE:

D. ADRIÁN SEGURA ORTIZ, estudiante del programa de doctorado TECNOLOGÍAS INFORMÁTICAS ha realizado bajo su dirección la tesis presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: CONTEXT-GUIDED COMPUTATIONAL METHODS FOR THE CONSENSUS INFERENCE OF GENE REGULATORY NETWORKS AND THE DETECTION OF CO-EXPRESSION PATTERNS.

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo. Así mismo, las publicaciones en coautoría que avalan dicha tesis no forman parte de otra tesis doctoral en la Universidad de Málaga ni en ninguna otra universidad.

En Málaga, a 25 de JULIO de 2025

Fdo.: JOSÉ FRANCISCO ALDANA MONTES y D. JOSÉ MANUEL GARCÍA NIETO
Director/es de tesis

Edificio Pabellón de Gobierno. Campus El Ejido. 29071
Tel.: 952 13 10 28 / 952 13 14 61 / 952 13 71 10
E-mail: doctorado@uma.es

DECLARACIÓN DE TUTORIZACIÓN Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D. ISMAEL NAVAS DELGADO, profesor doctor del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga

DECLARA QUE:

D. ADRIÁN SEGURA ORTIZ, estudiante del programa de doctorado TECNOLOGÍAS INFORMÁTICAS ha realizado bajo su tutorización la tesis presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: CONTEXT-GUIDED COMPUTATIONAL METHODS FOR THE CONSENSUS INFERENCE OF GENE REGULATORY NETWORKS AND THE DETECTION OF CO-EXPRESSION PATTERNS.

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo. Así mismo, las publicaciones en coautoría que avalan dicha tesis no forman parte de otra tesis doctoral en la Universidad de Málaga ni en ninguna otra universidad.

En Málaga, a 25 de JULIO de 2025

Fdo.: ISMAEL NAVAS DELGADO
Tutor de tesis

Acknowledgements

I would like to express my gratitude to my thesis supervisory team, composed of my supervisors, José Francisco Aldana Montes and José Manuel García Nieto, and my tutor, Ismael Navas Delgado, for their continuous support and guidance throughout these years. In particular, I wish to highlight the involvement of José Manuel, who has supervised every step of this thesis with patience and dedication, becoming a key reference in my academic journey.

My sincere thanks also go to Laetitia Jourdan for her warm welcome at the University of Lille during my predoctoral stay, which was an enriching and highly valuable experience for my professional development. I would also like to thank Professor Elisa Oltra from the University of Valencia for trusting me and sharing clinical data from her research studies, as well as her PhD student Karen Giménez for helping me interpret the results.

I am truly grateful to the Khaos research team, where I have had the opportunity to grow and learn, with a special mention to Antonio Benítez for his technical advice, which was fundamental in moving this thesis forward, and to Jorge Rodríguez for always being there to support and listen to me whenever I needed it. I also wish to thank the members of the ORKAD group in Lille, especially Adán José García, for working alongside me every day and sharing his knowledge, and Julie Jacques for her help in supervising my work.

I would also like to thank those who have given me the opportunity to share and disseminate my research beyond the academic community. In particular, I am grateful to the scientist and influencer Nathaly D. for spreading my work through her Instagram platform; to the IBIMA Institute for promoting my research both on social media and in the press; to the Scientific Communication Service of the University of Málaga for their support in publishing news articles; and to the Fundación Descubre of the Junta de Andalucía for their help in organising interviews and disseminating my work at a broader regional level.

Finally, I would like to thank my parents and my sister Andrea for their un-

conditional support throughout this journey, and my friends, especially Rosana, Sara, Fernando, and Salva, for always expressing their pride in me and being by my side during the most difficult moments. I am also deeply grateful to my psychologist, Ruxandra Vasilescu, for helping me understand myself better and for guiding me with empathy and wisdom during the most emotionally challenging times.

Contents

Resumen **3**

I Introduction and context **17**

1 Introduction **19**
1.1 Motivation 20
1.2 Objectives 21
1.3 Scientific contributions 23
1.3.1 Publications 25
1.3.2 Software 27
1.3.3 Global Overview 29
1.4 Thesis organization 31

2 Context and fundamentals **33**
2.1 Gene Expression and Regulatory Mechanisms 34
2.1.1 Fundamentals of Gene Expression 34
2.1.2 Gene Regulatory Networks (GRNs) 35
2.1.3 Gene Co-Expression 36
2.2 Computational Foundations for Optimization 38
2.2.1 Fundamentals of Optimization 38
2.2.2 Computational Methods for Optimization 39
2.2.3 Multi-Objective Optimization 41
2.2.4 Evolutionary Ensemble Learning 44
2.3 Computational Methods for Gene Expression Analysis 46
2.3.1 Inference of Gene Regulatory Networks 46
2.3.2 Co-Expression Detection via Biclustering 48

3 State of the art **55**
3.1 Gene Regulatory Networks Inference 56
3.1.1 Individual techniques 56

3.1.2 Consensus approaches 66
3.2 Co-expression Biclustering 69
3.2.1 Conventional methods 69
3.2.2 Methods based on evolutionary algorithms 73

4 Benchmark datasets 79
4.1 Gene Regulatory Networks Inference 80
4.1.1 Simulated expression data 80
4.1.2 Real-world expression data 87
4.2 Co-expression Biclustering 89
4.2.1 Simulated data 90
4.2.2 Real-world expression data 91

II Methodology, analysis and results 93

5 GENECI: Baseline consensus inference through mono-objective evolution 95
5.1 Algorithmic Proposal 95
5.1.1 Solution Representation 97
5.1.2 Evaluation 99
5.1.3 Selection 104
5.1.4 Crossover 105
5.1.5 Mutation 105
5.1.6 Output 106
5.2 Experimentation 106
5.2.1 Parameter Settings 106
5.2.2 Experimental Procedure 108
5.3 Results and Discussion 109
5.3.1 Benchmarks 110
5.3.2 Real-World: MELANOMA 123
5.3.3 Time Complexity 124

6 Memetic Inference: Incorporating prior knowledge into the evolutionary process 127
6.1 Proposed Approach 128
6.2 Experimentation 131
6.2.1 Parameter settings 132
6.3 Results and discussion 135

7 MO-GENECI: Multi-objective consensus guided by biological context 139
7.1 Algorithmic Proposal 140

7.1.1 Solution Representation 144
7.1.2 Crossover 144
7.1.3 Mutation 144
7.1.4 Evaluation 146
7.2 Experimentation 157
7.2.1 Parameter Settings 158
7.2.2 Experimental Procedure 159
7.3 Results and Discussion 160
7.3.1 MO-GENECI internal behavior 161
7.3.2 Experimental comparisons 166
7.3.3 Statistical Significance 171
7.3.4 Computational Complexity 179
7.3.5 Real-World Experimentation 181

8 PBEvoGen: Guiding GRN inference with expert-driven preference articulation 185
8.1 Proposed architecture and expert interaction 186
8.2 Experimentation 188
8.3 Results and discussion 191
8.3.1 Solution quality improvement 191
8.3.2 Spatial improvement 193
8.3.3 Performance improvement 194

9 BIO-INSIGHT: Maximizing biological coverage through many-objective optimization 197
9.1 Algorithmic Proposal 199
9.1.1 Objective 1: Quality 202
9.1.2 Objective 2: Motifs 203
9.1.3 Objective 3: Eigen Vector Distribution 204
9.1.4 Objective 4: Reduce Non-Essentials Interactions 205
9.1.5 Objective 5: Degree Distribution 207
9.1.6 Objective 6: Dynamicity 207
9.2 Experimentation 209
9.3 Results and Discussion 212
9.3.1 Performance Analysis 213
9.3.2 Objective Function Ablation Study 225
9.3.3 Real-world clinical application 226

10 MOEBA-BIO: Flexible framework for self-constructing evolutionary biclustering in biological domains 231
10.1 Methods 232

10.1.1 Representation 237
10.1.2 Objectives 239
10.1.3 Parameter autoconfigurator 243
10.1.4 Methodological Comparison 246
10.2 Experimentation 248
10.3 Results and discussion 250

11 MOEBA-BIO-CoExp: Context-guided evolutionary biclustering for gene co-expression analysis 257
11.1 Methods 257
11.1.1 New objective: Regulatory coherence 259
11.2 Experimentation 260
11.3 Results and discussion 264
11.3.1 Autoconfiguration 264
11.3.2 Candidates comparison: autoconfigurator validation . . . 267
11.3.3 Algorithmic comparison 268
11.3.4 Functional enrichment comparison 270
11.3.5 Scalability study 273

III Final observations 275

12 Conclusions 277
12.1 Gene regulatory networks consensus inference 277
12.2 Biclustering for biomedical domains 280

13 Future works 283
13.1 Gene regulatory network consensus inference 283
13.2 Biclustering for biomedical domains 284

List of tables 291

List of figures 291

List of algorithms 309

A Gene Regulatory Network Consensus Inference User Guide 311
Prerequisites 311
Installation 311
Output 311
Example procedure 312
Step 1: Obtain simulated expression data and their respective gold standards 312

Step 2: Inference and consensus of networks for the selected expression data . . . 313
Step 3: Representation of inferred networks . . . 315
Step 4: Validation of the inferred gene network . . . 316
Step 5: Binarization of the inferred gene network . . . 317
Additional commands and resources . . . 317

B Evolutionary Biclustering Algorithm for Expression Data User Guide 319
Prerequisites . . . 319
Basic Execution Command . . . 320
Input Files . . . 320
Representation Strategies . . . 321
Extending MOEBA-BIO with New Encoding . . . 322
Operator Support . . . 323
Defining Optimization Objectives . . . 323
Categories of Objective Functions . . . 323
Parameterized Objectives . . . 325
Implementing a Custom Objective . . . 325
Choosing the Optimization Algorithm . . . 326
Customizing Crossover and Mutation Operators . . . 327
Observers and Cache Management . . . 327
Parallel Execution and Output Files . . . 329
Example Execution . . . 329
Advanced Usage and Customization . . . 330

Bibliography 331

Resumen

Este resumen ofrece una visión general de los principales campos explorados en esta tesis doctoral, presentando la motivación que impulsa esta investigación, así como sus metas y objetivos. Además, se destacan las contribuciones más relevantes de la tesis en su ámbito de estudio. Finalmente, se resumen los principales hallazgos, se analizan sus implicaciones y limitaciones, y se proponen líneas de investigación futura basadas en los resultados obtenidos.

Introducción

El avance de la biología molecular y la disponibilidad creciente de datos transcriptómicos han generado nuevas oportunidades para comprender los mecanismos subyacentes a la regulación génica [1]. La capacidad para modelar con precisión las interacciones entre genes no solo permite avanzar en el conocimiento biológico, sino que también facilita el desarrollo de nuevas estrategias diagnósticas y terapéuticas en el ámbito biomédico [2, 3, 4, 5]. No obstante, el análisis e interpretación de estos datos sigue presentando importantes desafíos debido a su complejidad, su alta dimensionalidad y la presencia de ruido experimental [6, 7, 8].

En respuesta a estos retos, ha emergido un ecosistema muy diverso de metodologías computacionales diseñadas para abordar tareas como la inferencia de redes de regulación génica [9, 10, 11] o la detección de patrones de coexpresión [12, 13, 14]. Este abanico de enfoques, que incluye desde modelos estadísticos hasta técnicas avanzadas de inteligencia artificial, ha propiciado una intensa actividad investigadora en el campo, reflejo del interés por construir herramientas más precisas, eficientes y aplicables a distintos contextos biológicos.

Esta tesis propone un marco metodológico basado en el desarrollo de algoritmos híbridos guiados por el contexto, con una percepción holística del problema que permita mejorar la precisión y la aplicabilidad de las soluciones generadas. A

través de una combinación de técnicas de computación evolutiva, estrategias de consenso, funciones objetivo específicas y herramientas de evaluación, se persigue avanzar hacia soluciones más robustas e interpretables en el análisis computacional de la expresión génica.

Motivación

Tal como se ha expuesto en la introducción, el análisis de expresión génica ha propiciado el desarrollo de un amplio espectro de metodologías computacionales, tanto en la tarea de inferir redes de regulación génica (en inglés, Gene Regulatory Networks, en adelante GRNs) como en la identificación de patrones de coexpresión. Estas metodologías abordan los problemas desde enfoques muy diversos, lo que pone de manifiesto la vitalidad del campo, pero también evidencia la necesidad de un análisis crítico sobre cómo integrar eficazmente sus contribuciones y superar los retos aún no resueltos.

Entre los problemas más comunes se encuentra la elevada disparidad de resultados que ofrecen las diferentes técnicas [15]. Esta variabilidad además provoca que algunas propuestas obtengan un rendimiento sobresaliente en ciertos escenarios, pero que se degraden notablemente en otros, dificultando la selección metodológica y comprometiendo la generalización de los resultados [16]. Además, muchas de estas técnicas se centran exclusivamente en la optimización de criterios matemáticos o estadísticos, sin tener en cuenta información procedente del dominio biológico [17]. La ausencia de una guía basada en el contexto puede limitar la relevancia funcional de las soluciones y dificultar su interpretación experimental.

A estas carencias comunes se suman limitaciones particulares asociadas a cada tarea. En el caso de la inferencia de GRNs, la sensibilidad al ruido presente en los datos puede dar lugar a relaciones espurias o a la omisión de conexiones relevantes [18]. Por otro lado, en el análisis de coexpresión, las restricciones impuestas por las codificaciones internas de muchos algoritmos dificultan una visión global del problema, lo que ha producido una desconexión directa entre la solución algorítmica y la solución real del problema [19].

Ante esta situación, se hace necesario replantear el diseño de los métodos computacionales desde una perspectiva más integradora y contextualizada, que permita combinar el potencial técnico de las distintas aproximaciones con el conocimiento acumulado en el dominio biológico. En base a ello, esta tesis doctoral se sustenta en la siguiente hipótesis:

Hipótesis

La incorporación explícita de conocimiento biológico del dominio en algoritmos de computación evolutiva (mediante el diseño de objetivos biológicamente fundamentados, el uso de información previa y la consideración de las preferencias del experto), combinada con una percepción holística del problema y el consenso de enfoques metodológicos complementarios, permite una optimización multiobjetivo guiada por el contexto en la inferencia de redes de regulación génica y el biclustering, que mejora la precisión, la robustez y la coherencia biológica de las soluciones generadas.

Objetivos

El objetivo principal de esta tesis es el diseño y desarrollo de algoritmos híbridos guiados por el contexto cuya percepción holística aporte calidad y precisión en diversos problemas bioinformáticos. Este objetivo se divide en varios objetivos específicos de la siguiente manera:

Objetivo 1: Diseño de una estrategia de consensuado guiada por el contexto biológico de los datos cuya percepción holística aporte precisión a la inferencia de GRNs

1.1: Definir el conjunto de técnicas pertinente formado por propuestas bien consolidadas en el estado del arte actual y facilitar su ejecución mediante el desarrollo de un orquestador centralizado.

1.2: Selección del enfoque computacional más adecuado para la estrategia de consensuado, considerando diversas opciones dentro del ámbito de la inteligencia artificial.

1.3: Análisis del contexto biológico de los datos para identificar características relevantes que orienten el diseño.

1.4: Implementación modular y desacoplada del código, priorizando la reutilización y la durabilidad mediante el aislamiento de dependencias que faciliten su mantenimiento y adaptación futura.

Objetivo 2: Construcción de un benchmark académico extenso para la validación técnica de los métodos desarrollados.

2.1: Recopilación de redes de carácter académico con una amplia diversidad en tamaño, origen y naturaleza para garantizar la cobertura de distintos dominios de especialización.

2.2: Generación de datos sintéticos mediante el uso de distintos simuladores conocidos que permitan evaluar de manera sistemática el comportamiento y la robustez de los métodos desarrollados.

2.3: Implementación de métricas de validación para cuantificar la precisión y calidad de los resultados obtenidos en los experimentos, así como el desarrollo de herramientas de visualización y comparación algorítmica.

Objetivo 3: Exploración de la inyección de conocimiento como medio para reforzar la orientación basada en el contexto.

3.1: Diseño de funciones de cobertura biológica que permitan integrar aspectos relacionados con la estructura, topología y funcionalidad de las redes.

3.2: Implementación de funciones de búsqueda local que utilicen información previa conocida de las redes para guiar la optimización y mejorar la calidad de las soluciones.

3.3: Desarrollo de mecanismos de interacción con expertos que permitan la incorporación activa de su conocimiento en el proceso de inferencia.

Objetivo 4: Evaluación y mejora de la robustez y escalabilidad de las soluciones propuestas.

4.1: Análisis del impacto del tamaño de los datos en el rendimiento del software desarrollado, considerando datasets de diferente magnitud y complejidad.

4.2: Optimización del uso de recursos computacionales mediante estrategias de paralelización y aprovechamiento de arquitecturas de alto rendimiento.

4.3: Evaluación de la estabilidad de los resultados obtenidos ante perturbaciones en los datos y en los parámetros de los algoritmos.

Objetivo 5: Extrapolación del enfoque holístico y el software desarrollado a otros problemas bioinformáticos.

5.1: Evaluación de la aplicabilidad de la orientación basada en el contexto en otros problemas bioinformáticos de interés.

5.2: Identificación de oportunidades para nutrir problemas afines con la visión holística obtenida, promoviendo la transferencia de conoci-

miento y metodologías.

5.3: Adaptación del software implementado para facilitar su reutilización y extensión en contextos similares dentro del ámbito de la bioinformática.

Objetivo 6: Validación de la aplicabilidad clínica del software desarrollado empleando datos biológicos reales.

6.1: Identificación de recursos que contengan datos de expresión génica reales adecuados para evaluar la aplicabilidad del software.

6.2: Establecimiento de colaboraciones con investigadores del ámbito biomédico para llevar a cabo estudios interdisciplinares que validen los resultados obtenidos.

Asimismo, se establece como objetivo no funcional la publicación de implementaciones de código abierto para todas las contribuciones derivadas de esta tesis doctoral. Para ello, se seguirán las mejores prácticas en desarrollo de software y se garantizará una documentación detallada y de alta calidad para cada proyecto. Este enfoque busca facilitar el acceso y la reutilización de estas herramientas, tanto por la comunidad científica como por otros posibles usuarios.

Contribuciones científicas

Las principales contribuciones de esta tesis doctoral están relacionadas con los objetivos descritos en la sección anterior de la siguiente manera:

- En el Capítulo 5 se presenta `GENECI` [20], la primera propuesta algorítmica de esta tesis que establece un punto de partida inicial para abordar la inferencia consenso de GRNs. Esta propuesta propone un orquestador simple que permite ejecutar 12 técnicas de inferencia individuales (primera aproximación a **Objetivo 1.1**) y una estrategia de consenso sencilla basada en un algoritmo genético mono-objetivo (primera propuesta de los **Objetivos 1.2, 1.3** y **1.4**). Para su experimentación se han recolectado 28 redes procedentes de diferentes ediciones de los retos DREAM [21] y las dos versiones de la red de levadura de IRMA [22] (benchmark preliminar en **Objetivo 2.1**). Además, la calidad de los resultados ha sido cuantificada a través de métricas de precisión y verificada mediante representaciones gráficas estáticas sencillas (primera aproximación a **Objetivo 2.3**). Por último, se ha utilizado un conjunto de datos reales de expresión de pacientes de melanoma para validar la eficacia de la propuesta en entornos clínicos reales (**Objetivo 6.1**).

- El Capítulo 6 explora los beneficios de atender a interacciones génicas previamente conocidas a la estrategia anterior (**Objetivo 3.2**). `Memetic Inference` [23] extiende el núcleo evolutivo de `GENECI` incorporando una fase de búsqueda local destinada a minimizar la distancia entre las redes consenso y la pequeña muestra de interacciones etiquetadas. Los resultados muestran una clara mejora significativa de precisión respecto a las redes inferidas por la metodología original.

- El Capítulo 7 presenta `MO-GENECI` [24], una visión multi-objetivo del problema donde se refuerza la orientación basada en el contexto contemplando características más especificas de las redes biológicas, como la topología y patrones funcionales. Esta nueva visión se acompaña de un sólido refinamiento de muchos de los objetivos preliminarmente abordados en la propuesta inicial mono-objetivo: se mejora el orquestador integrando un total de 26 técnicas de inferencia (**Objetivo 1.1**), se sustituye el algoritmo genético inicial por un modelo evolutivo más sofisticado que incluye operadores de nuevo diseño adaptados al problema (**Objetivos 1.2, 1.3** y **1.4**), se extiende el benchmark académico a un total de 106 redes de regulación génica empleando más de 10 fuentes de datos distintas incluyendo varios simuladores (**Objetivos 2.1** y **2.2**), se mejora el proceso y visualización de comparación entre diferentes propuestas metodológicas (**Objetivo 2.3**) y se lleva a cabo un análisis más profundo de los datos de expresión de pacientes de melanoma (**Objetivo 6.1**).

- El Capítulo 8 propone un nuevo mecanismo destinado a la participación del experto del dominio en el sistema (**Objetivo 3.3**). Hasta ahora su papel se reducía a la elección final del individuo en el frente aproximado de Pareto obtenido. No obstante, la información conocida por este experto puede aportar también gran valor durante la optimización del consenso. Para aprovechar esto, la selección por preferencia implementada en `PBEvoGen` permite acotar la búsqueda a regiones específicas que el experto considere de alto interés biológico. Los resultados han demostrado no solo mejorar la precisión de las redes inferidas, sino sustituir convenientemente el esfuerzo de exploración por una mayor explotación de la región de interés permitiendo obtener el mismo nivel de optimización que el algoritmo original empleando la mitad de evaluaciones.

- El Capítulo 9 presenta `BIO-INSIGHT` [25], cuyo núcleo algorítmico representa la propuesta definitiva de esta tesis para optimizar el consenso de un amplio conjunto de técnicas individuales de inferencia de GRNs. Sus principales aportaciones son: la implementación de hasta 6 funciones de aptitud contrapuestos que aseguran una completa cobertura biológica del proble-

ma (**Objetivo 3.1**) y una refactorización exhaustiva destinada a mantener la factibilidad computacional de la propuesta en redes de gran tamaño (**Objetivo 4**). Su modelo asíncrono paralelo permite una evaluación simultánea entre individuos de incluso distintas generaciones (refinamiento del **Objetivo 1**) aprovechando al máximo todos los recursos computacionales del entorno de ejecución. Además, cabe mencionar que esta propuesta ha sido validada en un entorno clínico real gracias a la colaboración establecida con investigadoras de la Universidad de Valencia (**Objetivo 6.2**). En concreto, `BIO-INSIGHT` se ha empleado para inferir las redes génicas de pacientes de fibromialgia, encefalomielitis miálgica y otros con un codiagnóstico de ambas. Los resultados han permitido descubrir interacciones génicas presentes en ciertas patologías que no se han detectado en las muestras de control.

- El Capítulo 10 explora la extrapolación del enfoque abordado en esta tesis al problema del biclustering aplicado sobre datos biomédicos. En este capítulo se presenta `MOEBA-BIO` [26], un framework evolutivo de biclustering que permite la autoconstrucción de algoritmos en base al dominio biomédico de aplicación. En particular, se presenta una nueva codificación de individuos de perspectiva global (**Objetivo 5.2**) que permite la autodeterminación del número de biclusters como parte del aprendizaje y la implementación de objetivos directamente relacionados con el contexto (**Objetivo 5.1**). Los resultados de `MOEBA-BIO` sobre datos simulados demuestran una clara reducción de redundancia y un aumento significativo de calidad respecto a la codificación tradicional.

- El Capítulo 11 constituye una de las contribuciones más importantes de esta tesis, al representar el punto culminante en el que confluyen de forma integrada los avances desarrollados en las dos líneas de investigación principales: el biclustering sobre datos biomédicos y la inferencia de redes de regulación génica. Fruto de esta convergencia, se presenta un algoritmo evolutivo de biclustering desarrollado a partir de `MOEBA-BIO`, diseñado específicamente para la detección de coexpresión génica. La principal función de aptitud de esta contribución reutiliza el software implementado para la inferencia consenso de GRNs (**Objetivo 5.3**), permitiendo integrar de manera efectiva la relación entre las redes de regulación génica y la coexpresión génica. Los biclusters generados por `MOEBA-BIO-CoExp` han demostrado mejoras significativas en diversas métricas de precisión sobre datos simulados en comparación con otras metodologías de biclustering del estado del arte. Además, en conjuntos de datos reales de expresión génica en levadura (**Objetivo 6.1**), estos biclusters han mostrado un mayor

enriquecimiento funcional respecto a dichas metodologías, consolidando la eficacia real del enfoque de dirección basado en el contexto.

En la Figura 1 se presenta una visión global de las contribuciones desarrolladas a lo largo de esta tesis, organizadas en torno a tres áreas principales: la inferencia de redes de regulación génica, el biclustering aplicado a datos biomédicos y el diseño algorítmico. Las propuestas se ubican en las intersecciones correspondientes de estas áreas, reflejando su carácter transversal y complementario.

La línea de investigación centrada en la inferencia de GRNs parte de `GENECI`, una primera aproximación evolutiva al consenso de técnicas de inferencia, publicada en la revista Computers in Biology and Medicine (Q1) [20]. A partir de esta propuesta surgen dos desarrollos: por un lado, `Memetic Inference`, como una escisión que incorpora una fase de búsqueda local basada en conocimiento previo de interacciones génicas, y que fue presentada en el congreso internacional International Conference on Computational Science (ICCS) (IC) [23]; y por otro, `MO-GENECI`, que continúa la línea principal extendiendo el modelo hacia un enfoque multiobjetivo que considera propiedades topológicas y biológicas de las redes inferidas, también publicado en Computers in Biology and Medicine (Q1) [24]. Desde `MO-GENECI` se ramifica `PBEvoGen`, una propuesta centrada en la integración de conocimiento experto mediante la selección guiada por preferencias, actualmente en revisión en la revista Computational Biology and Chemistry (Q2), lo que establece un vínculo entre la optimización evolutiva y la interpretación humana. Finalmente, esta línea principal culmina en `BIO-INSIGHT`, que generaliza el marco de `MO-GENECI` para incorporar múltiples objetivos biológicamente informados y demostrar su aplicabilidad en contextos clínicos reales, también difundido en Computers in Biology and Medicine (Q1) [25]. Una recapitulación general de estas propuestas fue además presentada en el congreso regional Jornadas Andaluzas de Bioinformática (JABI) (RC).

En paralelo, `MOEBA-BIO` y `MOEBA-BIO-CoExp` abordan el problema del biclustering sobre datos biomédicos desde una perspectiva evolutiva genérica, basada en una codificación completa de soluciones y mecanismos de autodeterminación del número de patrones y autoconfiguración adaptativa. Ambas propuestas fueron presentadas de forma conjunta en una publicación en Computer Methods and Programs in Biomedicine (Q1) [26], con el objetivo de validar la utilidad del framework general (MOEBA-BIO) y demostrar su especialización eficaz en tareas de coexpresión génica (MOEBA-BIO-CoExp), incorporando para ello objetivos innovadores como la coherencia regulatoria.

En su conjunto, estas contribuciones representan un avance significativo en

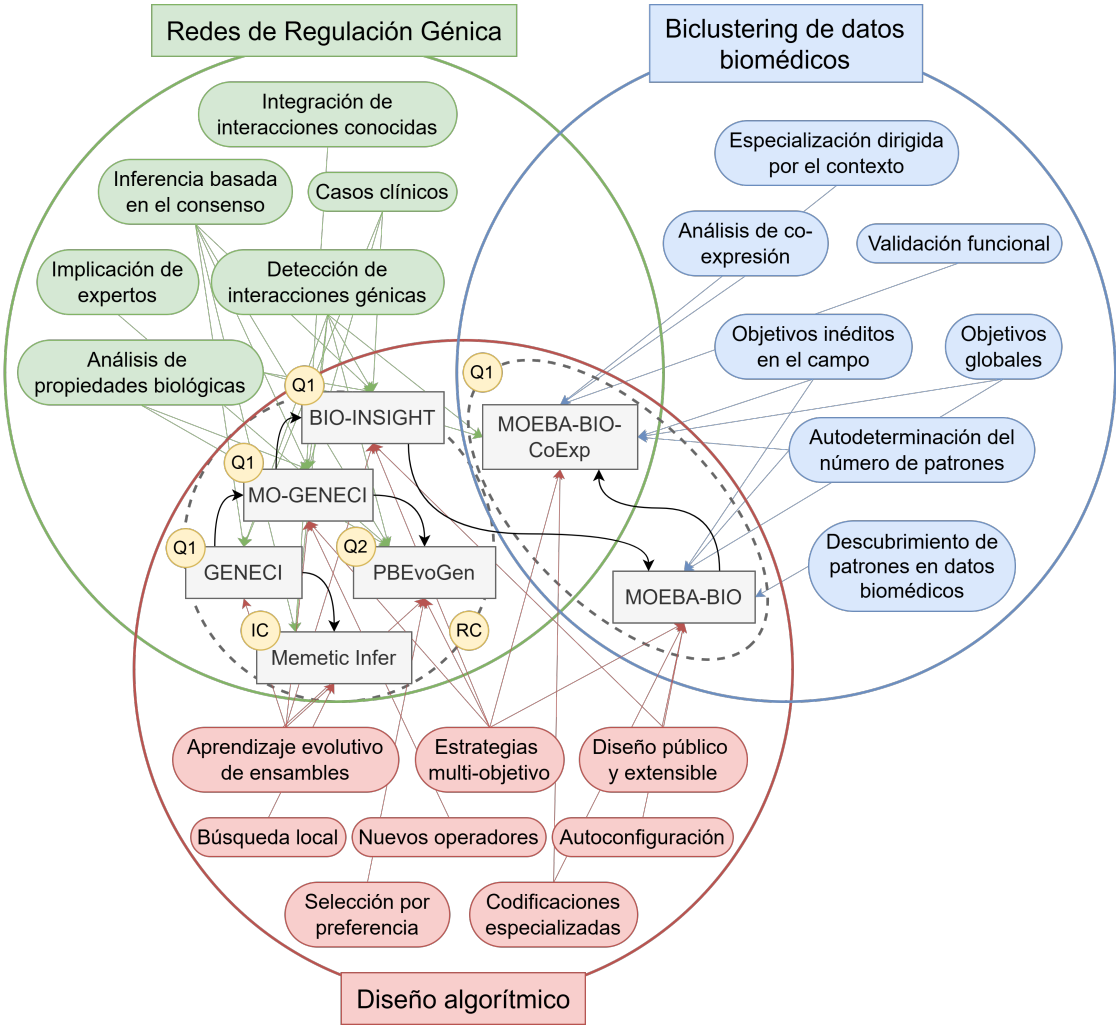


Figura 1: Representación gráfica de las contribuciones desarrolladas en esta tesis, organizadas en torno a tres áreas principales: inferencia de redes de regulación génica (verde), biclustering aplicado a datos biomédicos (azul) y diseño algorítmico (rojo). Las flechas de colores indican la vinculación de cada trabajo con los conceptos metodológicos o aplicados dentro de cada área, mientras que las flechas negras reflejan evolución directa o influencia entre propuestas. Las contribuciones se representan mediante recuadros grises ubicados en las intersecciones temáticas que abordan. En la esquina superior izquierda de cada recuadro se especifica su tipo de difusión: Q1 y Q2 indican publicación en revistas del primer o segundo cuartil en categorías de Ciencias Computacionales y Biología; IC señala difusión en congreso internacional; y RC, en congreso de ámbito regional.

ambas líneas de investigación, tanto en la inferencia de redes de regulación génica como en el biclustering evolutivo aplicado a datos biomédicos. Un avance

que ha sido claramente respaldado por la comunidad científica internacional mediante su difusión en revistas de alto impacto y congresos de referencia.

Conclusiones

Inferencia consensuada de redes de regulación génica

A lo largo de esta tesis se ha abordado progresivamente el problema de la inferencia por consenso de redes reguladoras de genes, superando las limitaciones de los enfoques tradicionales. La propuesta inicial, **GENECI**, introducida en el Capítulo 5, sentó las bases de un marco evolutivo capaz de combinar múltiples técnicas mediante un sistema de votación ponderada, sin depender de conocimiento previo. Sus evaluaciones en benchmarks estándares y en datos clínicos de melanoma demostraron una gran capacidad de generalización y una asignación inteligente de pesos, consolidándola como un punto de partida robusto.

Sobre esta base, **Memetic Inference**, presentada en el Capítulo 6, incorporó por primera vez la búsqueda local guiada con conocimiento experto, mejorando significativamente los resultados incluso con un uso limitado de conocimiento previo. Su capacidad para equilibrar la inclusión de información y evitar sesgos reforzó su versatilidad, especialmente en redes de gran tamaño.

Posteriormente, **MO-GENECI**, del Capítulo 7, superó la necesidad de agrupar múltiples criterios en una única función, empleando un enfoque evolutivo multiobjetivo que permitió optimizar simultáneamente la coherencia entre técnicas, la estructura topológica y los motivos regulatorios. Gracias a operadores evolutivos especializados y una rigurosa validación estadística, mostró una superioridad notable frente a 26 técnicas individuales en un amplio benchmark de 106 redes de todos los tamaños, y validó su aplicabilidad clínica redescubriendo interacciones relevantes y hallando otras nuevas en datos reales de melanoma.

A continuación, **PBEvoGen**, presentado en el Capítulo 8, aportó un mecanismo de selección basado en preferencias, otorgando a los expertos un papel activo al guiar la evolución mediante puntos de referencia en el espacio de soluciones. Esto mejoró considerablemente la precisión de las redes inferidas, superando incluso a `MO-GENECI`, y permitió reducir a la mitad el coste computacional en redes grandes sin sacrificar la calidad de los resultados.

Finalmente, **BIO-INSIGHT**, detallado en el Capítulo 9, constituye la propuesta más reciente y completa de esta tesis para la inferencia por consenso de redes reguladoras de genes. Su desarrollo abordó un amplio conjunto de preguntas de investigación, consolidando una estrategia evolutiva que mejora la precisión de

los enfoques anteriores gracias a un marco analítico extenso y a una perspectiva más rica y coherente biológicamente. En primer lugar, BIO-INSIGHT demostró que es posible guiar la inferencia hacia soluciones más robustas e interpretables, con estructuras y topologías acordes a patrones biológicos conocidos, mediante un espacio objetivo novedoso y fundamentado en compromisos biológicos. A nivel algorítmico, se confirmó que la estrategia propuesta no se solapa con el aprendizaje proporcionado por las técnicas individuales, sino que aporta conocimiento complementario, reforzando así el carácter integral e innovador de la propuesta. Además, el estudio de ablación evidenció que la optimización conjunta de los objetivos planteados genera redes de mayor calidad que aquellas obtenidas mediante la optimización individual de cada aspecto, justificando la naturaleza multifacética del modelo. Finalmente, la aplicabilidad de BIO-INSIGHT se validó en un contexto clínico real, mediante su aplicación a datos de expresión génica de pacientes con fibromialgia, encefalomielitis miálgica y diagnósticos conjuntos, donde se identificaron interacciones diferenciales específicas de cada patología, respaldadas por la literatura científica o evidencia experimental. Estos hallazgos no solo refuerzan la robustez de la propuesta, sino que también demuestran su utilidad como herramienta de apoyo para la comprensión de enfermedades complejas sin biomarcadores validados, sentando las bases para futuras aplicaciones biomédicas con impacto clínico.

Aplicación de biclustering a dominios biomédicos

Además de la línea principal de esta tesis sobre la inferencia consenso de redes de regulación génica, se ha explorado una línea complementaria centrada en el biclustering para contextos biomédicos. Para ello, se presentó **MOEBA-BIO** en el Capítulo 10, un marco evolutivo flexible que introduce una representación global en la que cada individuo codifica una solución completa, lo que permite evaluar conjuntamente todos los biclusters y superar la fragmentación típica de las representaciones tradicionales. Gracias a esta representación, se han propuesto nuevos objetivos de evaluación que aprovechan las relaciones entre biclusters y valoran propiedades globales como la diferenciación y la coherencia estructural, además de permitir la autodeterminación del número de biclusters.

En la fase inicial de validación, MOEBA-BIO superó a las arquitecturas evolutivas tradicionales con codificaciones parciales, obteniendo soluciones más coherentes y mejor alineadas con la estructura latente de los datos. Además, se ha demostrado la adaptabilidad del autoconfigurador de MOEBA-BIO a problemas biomédicos específicos, como la detección de grupos de genes coexpresados, mediante la especialización **MOEBA-BIO-CoExp** presentada en el Capítulo 11. Esta versión incorpora un nuevo objetivo global basado en la coherencia regulatoria,

lo que ha permitido obtener biclusters con alta coherencia funcional, validada en datos simulados y reales.

La capacidad de autoconfiguración, autodeterminación del número de biclusters y especialización por dominio de este enfoque, abre nuevas posibilidades para aplicar el biclustering evolutivo a problemas biomédicos complejos, proporcionando soluciones más realistas, explicables y acordes al contexto.

Trabajos futuros

A partir de los resultados y metodologías desarrollados en esta tesis, se han identificado varias líneas de investigación futura para consolidar y ampliar su aplicabilidad, tanto en la inferencia por consenso de redes génicas como en el biclustering en dominios biomédicos.

Inferencia por consenso de redes génicas

En el área de inferencia por consenso de redes génicas, se proponen las siguientes extensiones:

- Diseñar nuevas codificaciones para representar de forma más rica y flexible las interacciones génicas.
- Introducir un preprocesamiento basado en el agrupamiento estructural de las redes, con estrategias de optimización híbridas.
- Desarrollar un autoconfigurador supervisado guiado por métricas de calidad (AUROC y AUPR), para adaptar la parametrización del algoritmo a las propiedades estructurales de la red.
- Ampliar `PBEvoGen` para que los expertos puedan guiar el proceso de optimización de manera interactiva.
- Entrenar modelos de aprendizaje automático para ayudar a seleccionar soluciones prometedoras del frente de Pareto.
- Integrar mecanismos de IA explicable que permitan rastrear cada enlace regulador hasta los métodos que lo respaldan, facilitando la validación biológica y la detección de sesgos o limitaciones en las técnicas.

Biclustering para dominios biomédicos

En cuanto al biclustering para dominios biomédicos, se proponen estas direcciones futuras:

- Aplicar el autoconfigurador `MOEBA-BIO` a nuevos problemas biomédicos, como la detección de patrones epigenómicos o de respuesta a fármacos.
- Desarrollar funciones objetivo para datos heterogéneos, incluyendo atributos numéricos, categóricos y ordinales.
- Explorar codificaciones alternativas que permitan solapamiento tanto en filas como en columnas, mejorando la captura de patrones biológicos complejos.

Part I
Introduction and context

Chapter 1

Introduction

The advancement of molecular biology and the increasing availability of transcriptomic data have created new opportunities to understand the underlying mechanisms of gene regulation [1]. The ability to accurately model interactions between genes not only advances biological knowledge, but also facilitates the development of new diagnostic and therapeutic strategies in the biomedical field [2, 3, 4, 5]. Nevertheless, the analysis and interpretation of these data continue to present significant challenges due to their complexity, high dimensionality, and the presence of experimental noise [6, 7, 8].

In response to these challenges, a highly diverse ecosystem of computational methodologies has emerged, designed to address tasks such as the inference of gene regulatory networks [9, 10, 11] or the detection of co-expression patterns [12, 13, 14]. This range of approaches, which includes everything from statistical models to advanced artificial intelligence techniques, has spurred intense research activity in the field, reflecting the interest in developing more accurate, efficient, and applicable tools for various biological contexts.

This PhD thesis proposes a methodological framework based on the development of context-guided hybrid algorithms, with a holistic perception of the problem that enables improved accuracy and applicability of the generated solutions. Through a combination of evolutionary computation techniques, consensus strategies, specific objective functions, and evaluation tools, the aim is to advance toward more robust and interpretable solutions in the computational analysis of gene expression.

1.1 Motivation

As noted in the introduction, gene expression analysis has led to the development of a wide spectrum of computational methodologies for both inferring Gene Regulatory Networks (GRNs) and identifying co-expression patterns. These methods approach the problems from very diverse perspectives, highlighting the field's vitality but also underscoring the need for a critical analysis of how to effectively integrate their contributions and address the challenges that remain unresolved.

One of the most common issues is the high variability of results produced by different techniques [15]. This variability also causes some approaches to achieve outstanding performance in certain scenarios but degrade significantly in others, making methodological selection difficult and compromising the generalizability of the results [16]. Moreover, many of these techniques focus exclusively on the optimization of mathematical or statistical criteria, without taking into account information from the biological domain [17]. The absence of a context-based guide can limit the functional relevance of the solutions and hinder their experimental interpretation.

In addition to these common shortcomings, there are specific limitations associated with each task. For GRN inference, sensitivity to noise present in data can lead to spurious relationships or the omission of relevant connections [18]. In co-expression analysis, the constraints imposed by internal encodings of many algorithms hinder a global view of the problem, resulting in a direct disconnection between the algorithmic solution and the real solution to the problem [19].

In this situation, it is necessary to rethink the design of computational methods from a more integrative and contextualized perspective, enabling the combination of the technical potential of the different approaches with the accumulated knowledge in the biological domain. Based on this, the main hypothesis of this thesis is as follows:

Hipothesis

The explicit integration of domain-specific biological knowledge into evolutionary computation algorithms (through the design of biologically grounded objectives, the use of prior information, and the consideration of expert preferences) combined with a holistic understanding of the problem and the consensus of complementary methodological approaches, enables context-guided multi-objective optimization in gene regulatory network inference and biclustering, enhancing the accuracy, robustness, and biological coherence of the resulting solutions.

1.2 Objectives

The main objective of this thesis is the design and development of context-guided hybrid algorithms whose holistic perspective enhances the quality and accuracy in various bioinformatics problems. This objective is divided into several specific goals as follows:

Objective 1: Design of a consensus strategy guided by the biological context of the data, whose holistic perception improves the accuracy of GRN inference

1.1: Define a relevant set of techniques composed of well-established proposals from the current state of the art and facilitate their execution through the development of a centralized orchestrator.

1.2: Select the most suitable computational approach for the consensus strategy, considering options within the field of artificial intelligence.

1.3: Analyze the biological context of the data to identify relevant features that guide the design.

1.4: Implement modular code, prioritizing reusability and durability through dependency isolation to ease future maintenance and adaptation.

Objective 2: Build an extensive academic benchmark for the technical validation of the developed methods

2.1: Collect academic networks with a wide diversity in size, origin, and nature to ensure coverage of different domains of specialization.

2.2: Generate synthetic data using well-known simulators to systematically assess the behavior and robustness of the developed methods.

2.3: Implement validation metrics to quantify the accuracy and quality of the experimental results, along with the development of visualization and algorithm comparison tools.

Objective 3: Exploration of knowledge injection as a means to reinforce context-based guidance

3.1: Design biological coverage functions that integrate aspects related to the structure, topology, and functionality of the networks.

3.2: Implement local search functions that exploit prior network knowledge to guide optimization and improve solution quality.

3.3: Develop interaction mechanisms with experts to enable the active incorporation of their knowledge into the inference process.

Objective 4: Evaluation and improvement of the robustness and scalability of the proposed solutions

4.1: Analyze the impact of dataset size on software performance, considering different magnitudes and complexities.

4.2: Optimize the use of computational resources through parallelization strategies and the use of high-performance architectures.

4.3: Evaluate the stability of the obtained results in response to data perturbations and algorithm parameter variations.

Objective 5: Extrapolation of the holistic approach and the developed software to other bioinformatics problems

5.1: Evaluate the applicability of context-based guidance in other relevant bioinformatics problems.

5.2: Identify opportunities to enrich related problems with the holistic perspective obtained, promoting knowledge and methodology transfer.

5.3: Adapt the implemented software to facilitate its reuse and extension in similar contexts within the field of bioinformatics.

Objective 6: Validation of the clinical applicability of the developed software using real-world biological data

6.1: Identify resources containing real-world gene expression data suitable for evaluating the applicability of the software.

6.2: Establish collaborations with researchers from the biomedical domain to carry out interdisciplinary studies that validate the results obtained.

Additionally, a non-functional objective is set: the publication of open-source implementations for all contributions derived from this doctoral thesis. To this end, best practices in software development will be followed, and each project will include detailed and high-quality documentation. This approach aims to facilitate access and reuse of these tools by both the scientific community and other potential users.

1.3 Scientific contributions

The main contributions of this doctoral thesis are related to the objectives described in the previous section as follows:

- Chapter 5 presents `GENECI` [20], the first software package of this thesis that establishes an initial starting point to address the consensus inference of GRNs. This proposal introduces a simple orchestrator that enables the execution of 12 individual inference techniques (first approach to **Objective 1.1**) and a straightforward consensus strategy based on a single-objective genetic algorithm (first proposal for **Objectives 1.2, 1.3** and **1.4**). For its experimentation, 28 networks have been collected from different editions of the DREAM challenges [21] and the two versions of the yeast IRMA network [22] (preliminary benchmark in **Objective 2.1**). In addition, the quality of the results has been quantified through precision metrics and verified using simple static graphical representations (first approach to **Objective 2.3**). Finally, a set of real-world gene expression data from melanoma patients has been used to validate the effectiveness of the proposal in real clinical settings (**Objective 6.1**).

- Chapter 6 explores the benefits of incorporating previously known gene interactions into the previous strategy (**Objective 3.2**). `Memetic Inference` [23] extends the evolutionary core of `GENECI` by incorporating a local search phase aimed at minimizing the distance between the consensus networks and the small sample of labeled interactions. The results show a significant precision improvement compared to the original methodology.

- Chapter 7 presents `MO-GENECI` [24], a multi-objective vision of the problem that reinforces the context-based guidance by considering more specific features of biological networks, such as topology and functional motifs. This perspective is accompanied by a solid refinement of many goals from the initial single-objective proposal: the orchestrator is improved by integrating a total of 26 inference techniques (**Objective 1.1**), the initial genetic algorithm is replaced by a more sophisticated evolutionary model that includes newly designed operators tailored to the problem (**Objectives 1.2, 1.3** and **1.4**), the academic benchmark is expanded to a total of 106 gene regulatory networks using more than 10 different data sources including various simulators (**Objectives 2.1** and **2.2**), the process and visualization of comparisons between different methodological proposals are improved (**Objective 2.3**), and a deeper analysis of the expression data from melanoma patients is carried out (**Objective 6.1**).

- Chapter 8 proposes a new mechanism aimed at involving the domain expert in the system (**Objective 3.3**). Until now, their role was limited to the final selection in the approximated Pareto front, but their knowledge can also add value during consensus optimization. To leverage this, the preference-based selection in `PBEvoGen` confines the search to regions deemed of high biological interest by the expert. The results have shown not only an improvement in the precision of the inferred networks but also a convenient substitution of the exploration effort by increased exploitation of the region of interest, enabling the same level of optimization as the original algorithm while using only half of the evaluations.

- Chapter 9 presents `BIO-INSIGHT` [25], whose algorithmic core represents the final proposal of this thesis for optimizing the consensus of a broad set of individual GRN inference techniques. Its main contributions include: the implementation of up to six opposing fitness functions that ensure comprehensive biological coverage of the problem (**Objective 3.1**) and an exhaustive refactoring aimed at maintaining the computational feasibility of the proposal on large-scale networks (**Objective 4**). Its asynchronous parallel model allows simultaneous evaluation of individuals, even across different generations (refinement of **Objective 1**), maximizing the use of all computational resources of the execution environment. Furthermore, it is worth mentioning that this proposal has been validated in a real clinical setting thanks to the collaboration established with researchers from the University of Valencia (**Objective 6.2**). Specifically, `BIO-INSIGHT` was used to infer gene networks of patients with fibromyalgia, myalgic encephalomyelitis, and co-diagnosis, uncovering interactions absent in control samples.

- Chapter 10 explores the extrapolation of the approach addressed in this thesis to the problem of biclustering applied to biomedical data. In this chapter, `MOEBA-BIO` is presented [26], an evolutionary biclustering framework for self-constructing algorithms tailored to the biomedical domain. In particular, it introduces a new global perspective encoding of individuals (**Objective 5.2**) that allows the number of biclusters to be self-determined as part of the learning process and the implementation of objectives directly related to the context (**Objective 5.1**). The results of `MOEBA-BIO` on simulated data show a clear reduction in redundancy and a significant quality improvement compared to the traditional encoding.

- Chapter 11 presents an evolutionary biclustering algorithm developed from `MOEBA-BIO`, specifically designed for the detection of gene co-expression. The main fitness function reuses the software for GRN consensus inference (**Objective 5.3**), integrating the relationship between gene regulation and

co-expression. Biclusters from `MOEBA-BIO-CoExp` [26] showed significant precision gains on simulated data over other state-of-the-art methods. Furthermore, in real-world gene expression datasets from yeast (**Objective 6.1**), these biclusters have shown higher functional enrichment compared to those methodologies, consolidating the real-world effectiveness of the context-based guidance approach.

1.3.1 Publications

The development of this PhD thesis has led to the publication of 4 scientific articles in prestigious journals and 1 conference. Below is a detailed list of these publications, including their quality and impact metrics.

- Adrián Segura-Ortiz, José García-Nieto, José F Aldana-Montes, Ismael Navas-Delgado. **"GENECI: a novel evolutionary machine learning consensus-based approach for the inference of gene regulatory networks"**. In: *Computers in Biology and Medicine* **155** (2023), p. 106653, DOI: `https://doi.org/10.1016/j.compbiomed.2023.106653`, Impact Factor: 7.0.
  - Category of *Computer Science, Interdisciplinary Applications* (Q1, Rank: 19/170, Perc.: 89.1)
  - Category of *Biology* (Q1, Rank: 7/109, Perc.: 94.0)
  - Category of *Engineering, Biomedical* (Q1, Rank: 16/123, Perc.: 87.4)
  - Category of *Mathematical & Computational Biology* (Q1, Rank: 2/66, Perc.: 97.7)
- Adrián Segura-Ortiz, José García-Nieto, José F Aldana-Montes, Ismael Navas-Delgado. **"Multi-objective context-guided consensus of a massive array of techniques for the inference of Gene Regulatory Networks"**. In: *Computers in Biology and Medicine* **179** (2024), p. 108850, DOI: `https://doi.org/10.1016/j.compbiomed.2024.108850`, Impact Factor: 6.3.
  - Category of *Computer Science, Interdisciplinary Applications* (Q1, Rank: 26/175, Perc.: 85.4)
  - Category of *Biology* (Q1, Rank: 7/107, Perc.: 93.9)
  - Category of *Engineering, Biomedical* (Q1, Rank: 22/124, Perc.: 82.7)
  - Category of *Mathematical & Computational Biology* (Q1, Rank: 4/67, Perc.: 94.8)

- Adrián Segura-Ortiz, Adán José-García, Laetitia Jourdan, José García-Nieto. **"Exhaustive biclustering driven by self-learning evolutionary approach for biomedical data"**. In: *Computer Methods and Programs in Biomedicine* **2025**, p. 108846, DOI: https://doi.org/10.1016/j.cmpb.2025.108846, Impact Factor: 4.8.
  - Category of *Computer Science, Theory & Methods* (Q1, Rank: 20/147, Perc.: 86.7)
  - Category of *Medical Informatics* (Q1, Rank: 11/48, Perc.: 78.1)
  - Category of *Engineering, Biomedical* (Q2, Rank: 35/124, Perc.: 72.2)
  - Category of *Computer Science, Interdisciplinary Applications* (Q2, Rank: 47/175, Perc.: 73.4)

  Note: This work was carried out during a three-month predoctoral stay at the University of Lille within the ORKAD group, resulting in a publication that supports the **international mention** of this PhD thesis.

- Adrián Segura-Ortiz, Karen Giménez-Orenga, José García-Nieto, Elisa Oltra, José F. Aldana-Montes. **"Multifaceted evolution focused on maximal exploitation of domain knowledge for the consensus inference of Gene Regulatory Networks"**. In: *Computers in Biology and Medicine* **196** (2025), p. 110632, DOI: https://doi.org/10.1016/j.compbiomed.2025.110632, Impact Factor: 6.3.
  - Category of *Computer Science, Interdisciplinary Applications* (Q1, Rank: 26/175, Perc.: 85.4)
  - Category of *Biology* (Q1, Rank: 7/107, Perc.: 93.9)
  - Category of *Engineering, Biomedical* (Q1, Rank: 22/124, Perc.: 82.7)
  - Category of *Mathematical & Computational Biology* (Q1, Rank: 4/67, Perc.: 94.8)

- Adrián Segura-Ortiz, José García-Nieto, José F Aldana-Montes. **"Exploiting medical-expert knowledge via a novel memetic algorithm for the inference of gene regulatory networks"**. In: *International Conference on Computational Science (ICCS)*. Springer, 2024, pp. 3–17, DOI: https://doi.org/10.1007/978-3-031-63772-8_1.
  - Conference Quality (CORE2023): Rank Multiconference, Field of Research: 4601 - Applied computing.

– Conference Quality (CORE2021): Rank A, Field of Research: 4601 - Applied computing.

It is also worth noting that the proposal `PBEvoGen` is currently under review in the journal *Computational Biology and Chemistry*. Moreover, all contributions developed in this PhD thesis concerning GRNs inference were jointly presented at the 2025 edition of the *Jornadas Andaluzas de Bioinformática (JABI 2025)*, providing a comprehensive overview of the research outcomes in this line.

In addition to the scientific recognition achieved through peer-reviewed publications, the impact and dissemination of the aforementioned research were effectively enhanced through press coverage and social media outreach. GENECI was featured in a press release by Europa Press[1], MO-GENECI was highlighted in Diario Sur[2], MOEBA-BIO was published in Cadena Ser [3] and BIO-INSIGHT was reported in Salud a Diario [4]. Social media played a crucial role in further amplifying the research, with dissemination occurring through profiles aimed at young researchers, such as biomedical influencer Nathaly's account, where a collaborative reel surpassed 35,000 views[5], as well as through institutional channels like IBIMA's Instagram account[6] (5,000 views), the Instagram account of the Ministry of University and Research of the Government of Andalusia [7] (12,000 views) and ITIS UMA's post on X (formerly Twitter)[8].

1.3.2 Software

In addition to its scientific contributions, this doctoral thesis emphasizes practical applicability by making all implementations and datasets publicly available through multiple open-source repositories, thereby fostering reproducibility and supporting their adoption by the scientific community. The associated projects are publicly available under the MIT license and include:

[1] https://www.europapress.es/esandalucia/malaga/noticia-desarrollan-algoritmo-util-comprension-procesos-celulares-implicados-distintas-enfermedades-20240829115250.html

[2] https://www.diariosur.es/malaga-capital/investigadores-ibima-plataforma-bionand-avanzan-comprension-enfermedad-20230819134019-nt.html

[3] https://cadenaser.com/andalucia/2025/07/09/moeba-bio-una-nueva-herramienta-desarrollada-en-la-uma-para-descubrir-estructuras-ocultas-en-datos-biomedicos-ser-malaga/

[4] https://www.saludadiario.es/investigacion/un-sistema-inteligente-analiza-los-genes-implicados-en-enfermedades-como-la-fatiga-cronica/

[5] https://www.instagram.com/reel/C_bri8BPcJc/?igsh=NHZoeTBpYzk1cHJp

[6] https://www.instagram.com/reel/C_chjeWIvhw/?igsh=N3ZnMDV2b3Zkc2N4

[7] https://www.instagram.com/reel/DMhqCVoMptb/?igsh=bjdxZ2EzNDE0Zmx3

[8] https://x.com/itis_uma/status/1830874768372810209?t=Z842pV_66iDI7vsqQ747fQ

- *GENECI*:
  - Git repository: `https://github.com/AdrianSeguraOrtiz/Single-GENECI`
  - PyPI: `https://pypi.org/project/GENECI/1.0.2/`
  - Docker: 24 images at `https://hub.docker.com/u/adriansegura99?page=1&search=geneci` (Tag 1.0.0).
- *Memetic Inference*:
  - Git repository: `https://github.com/AdrianSeguraOrtiz/Memetic-GENECI`
  - PyPI: `https://pypi.org/project/GENECI/1.5.2/`
  - Docker: 24 images at `https://hub.docker.com/u/adriansegura99?page=1&search=geneci` (Tag 1.5.1).
- *MO-GENECI*:
  - Git repository: `https://github.com/AdrianSeguraOrtiz/MO-GENECI`
  - PyPI: `https://pypi.org/project/GENECI/2.0.2/`
  - Docker: 38 images at `https://hub.docker.com/u/adriansegura99?page=1&search=geneci` (Tag 2.0.0).
- *PBEvoGen*:
  - Git repository: `https://github.com/AdrianSeguraOrtiz/PBEvoGen`
  - PyPI: `https://pypi.org/project/GENECI/2.5.1/`
  - Docker: 38 images at `https://hub.docker.com/u/adriansegura99?page=1&search=geneci` (Tag 2.5.1).
- *BIO-INSIGHT*:
  - Git repository: `https://github.com/AdrianSeguraOrtiz/BIO-INSIGHT`
  - PyPI: `https://pypi.org/project/GENECI/3.0.1/`
  - Docker: 38 images at `https://hub.docker.com/u/adriansegura99?page=1&search=geneci` (Tag 3.0.0).
- *MOEBA-BIO*:
  - Git repository: `https://github.com/AdrianSeguraOrtiz/MOEBA-BIO`

In addition, all the contributions related to consensus-based network inference (GENECI, Memetic Inference, MO-GENECI, PBEvoGen, and BIO-INSIGHT) have been unified into a single Git repository that integrates all functionalities into a cohesive framework, available at: https://github.com/AdrianSeguraOrtiz/GENECI. This repository is fully mirrored in the corresponding Python package hosted on PyPI, which has been progressively updated to include each of the developments and improvements introduced throughout the PhD thesis: https://pypi.org/project/GENECI. The impact and adoption of the package are further evidenced by its download statistics, accessible through the ClickPy dashboard (https://clickpy.clickhouse.com/dashboard/geneci), which report over 15,000 downloads to date. These metrics confirm that the software developed in this PhD thesis is not only accessible but also actively used, effectively fulfilling its purpose as a reusable and reproducible scientific resource.

1.3.3 Global Overview

Figure 1 provides a global overview of the contributions developed throughout this thesis, organized around three main areas: inference of gene regulatory networks, biclustering applied to biomedical data, and algorithm design. The proposals are positioned at the intersections of these areas, reflecting their transversal and complementary nature.

The research line focused on GRN inference begins with GENECI, an initial evolutionary approach to the consensus of inference techniques, published in the journal Computers in Biology and Medicine (Q1) [20]. From this work, two developments emerge: on the one hand, Memetic Inference, a spin-off that incorporates a local search phase based on prior knowledge of gene interactions, presented at the International Conference on Computational Science (ICCS) (IC) [23]; and on the other hand, MO-GENECI, which continues the main line by extending the model to a multi-objective approach that considers both topological and biological properties of the inferred networks, also published in Computers in Biology and Medicine (Q1) [24]. From MO-GENECI, PBEvoGen branches out, a proposal focused on integrating expert knowledge through preference-guided selection, currently under review in the journal Computational Biology and Chemistry (Q2), establishing a link between evolutionary optimization and human interpretability. Finally, this main line culminates in BIO-INSIGHT, which generalizes the MO-GENECI framework to incorporate multiple biologically informed objectives and demonstrate its applicability in real-world clinical contexts, also disseminated in Computers in Biology and Medicine (Q1) [25]. A general recap of these proposals was also presented at the regional conference Jornadas Andaluzas de Bioinformática (JABI) (RC).

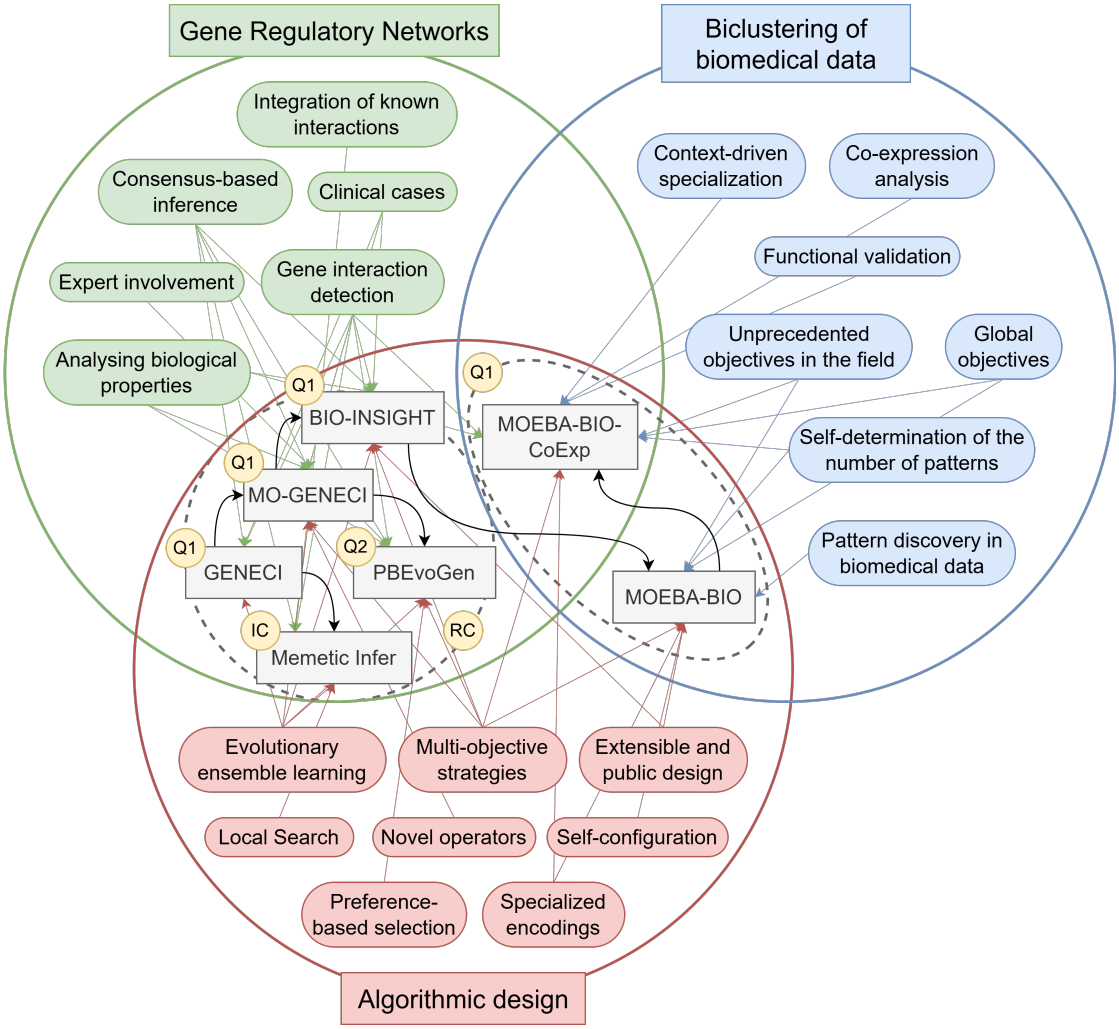


Figure 1.1: Graphical representation of the contributions developed in this thesis, organized around three main areas: inference of gene regulatory networks (green), biclustering applied to biomedical data (blue), and algorithm design (red). The colored arrows indicate the connection of each work with methodological or applied concepts within each area, while the black arrows reflect direct evolution or influence between proposals. The contributions are represented by gray boxes placed at the thematic intersections they address. The top-left corner of each box specifies its type of dissemination: Q1 and Q2 indicate publication in first- or second-quartile journals in the fields of Computer Science and Biology; IC refers to dissemination at an international conference; and RC, at a regional-level conference.

In parallel, `MOEBA-BIO` and `MOEBA-BIO-CoExp` address the problem of biclustering on biomedical data from a generic evolutionary perspective, based on a complete encoding of solutions and mechanisms for self-determination of the

number of patterns and adaptive self-configuration. Both proposals were jointly presented in a publication in Computer Methods and Programs in Biomedicine (Q1) [26], aiming to validate the utility of the general framework (MOEBA-BIO) and demonstrate its effective specialization in gene co-expression tasks (MOEBA-BIO-CoExp), incorporating innovative objectives such as regulatory coherence.

Altogether, these contributions represent a significant advance in both research lines, gene regulatory network inference and evolutionary biclustering applied to biomedical data. An advance that has been clearly supported by the international scientific community through its dissemination in high-impact journals and leading conferences.

1.4 Thesis organization

This thesis is structured in three main parts. Part I focuses on the contextualization and consists of four main chapters: Chapter 1 is the current chapter, in which a brief introduction, motivations, objectives, and the main scientific and computational contributions of this thesis are developed; in Chapter 2, the principles of all the concepts covered in the research are described, including the fundamentals of gene expression, computational bases of optimization, and fundamental methodologies for gene expression analysis; Chapter 3 presents the current state-of-the-art and an exhaustive review of the available computational proposals to address the two main bioinformatics challenges covered in this thesis, GRN inference and gene co-expression detection; finally, in Chapter 4, a compilation of all gene expression datasets, both synthetic and real, used to validate the different methodological contributions of this thesis within the biomedical context addressed is carried out.

Part II presents the methodology designed in this thesis to address the aforementioned bioinformatics challenges. This leads to a subdivision into seven chapters, each aimed at presenting a distinct computational contribution. This involves Chapters 5, 6, 7, 8, 10, and 11, whose content has been previously detailed in Section 1.3.

Part III contains the final observations and is composed of two chapters. In Chapter 12, the main conclusions drawn from the work developed throughout the thesis are presented, highlighting the scientific and computational contributions achieved. Finally, Chapter 13 outlines possible future research directions, both in improving the proposed methodologies and in their application to new biomedical scenarios and related computational problems.

Chapter 2

Context and fundamentals

This chapter provides the necessary foundations to understand the biological and computational context in which this thesis is framed. Firstly, it introduces the key concepts related to gene expression and its regulatory mechanisms, addressing both the formation of gene regulatory networks (GRNs) and the phenomenon of co-expression.

Subsequently, it presents the fundamental principles of computational optimization, with a special emphasis on evolutionary methods and multi-objective optimization. Finally, it describes the main computational strategies employed for the inference of GRNs and the detection of co-expression patterns through biclustering, thus establishing the theoretical framework necessary for the subsequent methodological development.

2.1 Gene Expression and Regulatory Mechanisms

The study of gene expression and its regulatory mechanisms is fundamental to understanding how cells control protein production and respond to various stimuli [27]. Gene expression does not occur in isolation; rather, it is regulated by complex interactions between multiple genes and transcription factors, giving rise to gene regulatory networks (GRNs) [28]. Moreover, genes that exhibit similar expression patterns under certain conditions may be functionally related, which has motivated the analysis of gene co-expression as a tool to identify functional modules in expression data [29]. This section presents the fundamental concepts of gene expression, GRNs, and co-expression, providing the necessary context to subsequently understand the computational approaches used in their study.

2.1.1 Fundamentals of Gene Expression

Gene expression is defined as the process by which the information contained in DNA is used to build proteins or to generate non-coding RNA molecules with other functionalities [27]. The expression of a gene can vary depending on the cell type [30], environmental conditions [31], and the stage of the cell cycle [32], making it a dynamic and highly complex mechanism.

Gene expression encompasses the central dogma of molecular biology [33]. Therefore, the first step in gene expression is the transcription process carried out by RNA polymerase, an enzyme that copies the DNA sequence into a messenger RNA (mRNA) molecule. This mRNA acts as an intermediary, transporting the genetic information from the nucleus to the cytoplasm. In the cytoplasm, the mRNA associates with ribosomes, where the translation process takes place. During translation, the mRNA is read in codons, sequences of three nucleotides that code for specific amino acids. Ribosomes facilitate the linking of these amino acids in the sequence determined by the mRNA, with the help of transfer RNA (tRNA), which carries the appropriate amino acids and incorporates them into the growing polypeptide chain. This process results in the synthesis of functional proteins, essential for the development and maintenance of the cell [34].

The regulation of this process is complex and enables cells to control which genes are activated or repressed at any given time. Although there are multiple regulatory mechanisms, one of the most relevant is the action of gene products on other genes [35]. Transcription factors, which are proteins encoded by specific genes, can bind to regulatory DNA sequences such as promoters and enhancers to activate or inhibit the transcription of other genes [36]. The fact

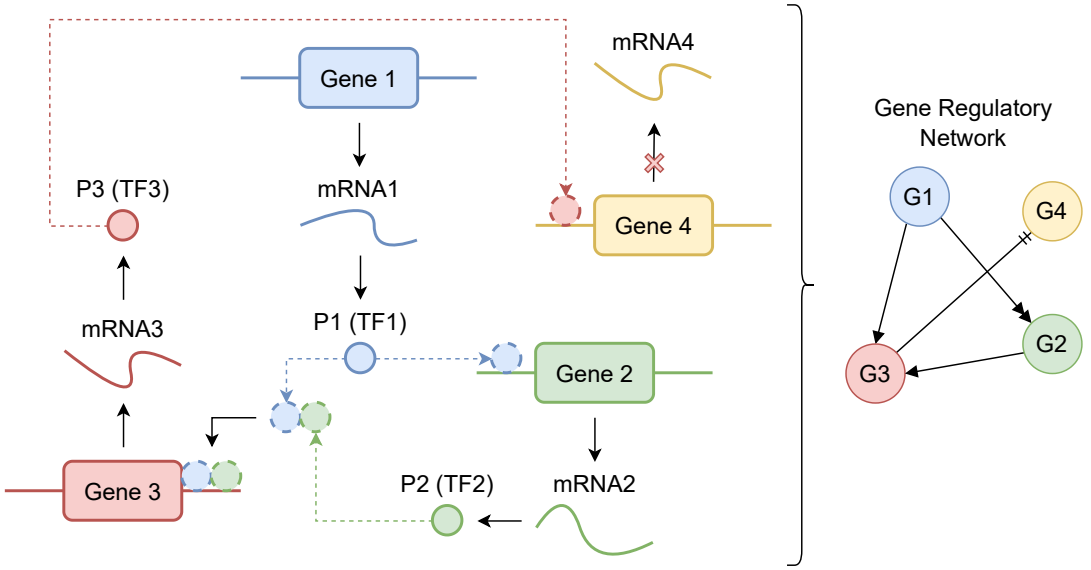


Figure 2.1: Biological basis of a gene regulatory network. The left part shows the molecular mechanisms that give rise to a gene regulatory network. Each gene (colored boxes) is transcribed into a messenger RNA (mRNA) molecule, which is then translated into a protein. Some of these proteins act as transcription factors (TFs), capable of binding to specific DNA sequences to modulate (activate or repress) the expression of other genes. The right part represents the resulting gene regulatory network, a computational abstraction where the nodes (G1–G4) correspond to genes and the directed edges indicate regulatory relationships between them.

that more than 5% of our genes are predicted to encode transcription factors underscores the importance of this family of proteins in biology [37]. Nevertheless, it is worth mentioning that gene expression can also be influenced by epigenetic modifications, which alter the accessibility of DNA without modifying its sequence [38].

Precise control over the production of RNA molecules and proteins is fundamental for development, cellular differentiation, and organismal homeostasis [27]. Alterations in this process can lead to pathologies such as cancer [39, 40] or other diseases [41], highlighting the importance of its study in biomedicine and biotechnology.

2.1.2 Gene Regulatory Networks (GRNs)

Gene regulatory networks (GRNs) [28] represent the set of functional interactions through which certain genes regulate the expression of others (see Figure

2.1). These networks enable the precise coordination of gene activity in fundamental processes such as development, cellular differentiation, and response to environmental stimuli [42].

A GRN consists of three key elements: regulator genes, target genes, and regulatory connections. A regulator gene (source) is one whose activity affects the expression of another gene (target), establishing an influence relationship within the network. The connections between them can be described through three fundamental properties:

- **Directionality**: The interactions are directed, which means that a regulator gene affects a target gene in a unidirectional relationship.
- **Sign**: The connections can be activating or repressing. An activator regulator promotes the transcription of the target gene, while a repressor regulator inhibits it.
- **Intensity**: The magnitude of regulation varies, reflecting the strength of the regulator's effect on its target gene. This intensity can depend on factors such as the concentration of the regulator, the binding affinity to the promoter region, and the presence of additional cellular signals.

GRNs are commonly represented as directed graphs, where the nodes correspond to genes and the edges indicate regulatory interactions [43]. In these graphs, the edges can carry additional labels to specify the sign (activation or repression) or be weighted to reflect the intensity of the regulation. Structural analysis of these networks has enabled the identification of topological features [44, 45] and regulatory motifs [46, 47] that play a key role in the stability and plasticity of gene expression [48].

Understanding GRNs is essential for unraveling the underlying molecular mechanisms in numerous biological processes. Alterations in these networks can lead to gene dysregulation, contributing to diseases such as cancer or metabolic disorders [49]. Thanks to advances in bioinformatics and network inference methods, it is now possible to model these interactions from gene expression data [9, 50, 11], providing key information for systems biology and personalized medicine [3, 4].

2.1.3 Gene Co-Expression

Gene co-expression refers to the phenomenon in which two or more genes exhibit similar expression patterns across different experimental conditions, tissues, or cellular states [51]. Unlike direct regulation in a gene regulatory net-

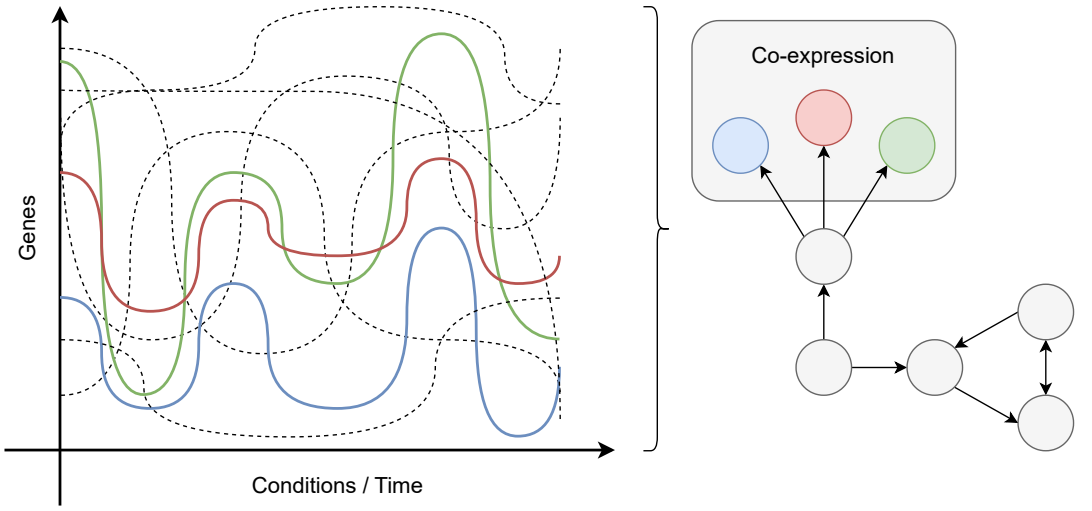


Figure 2.2: Conceptual representation of gene co-expression. On the left, the expression profiles of different genes are shown, captured under various experimental conditions or over time. When two or more genes exhibit similar patterns, such as a simultaneous increase or decrease in their activity, they are considered co-expressed. On the right, it is illustrated how these genes tend to be regulated by common factors, suggesting a possible functional coordination within the biological system.

work (GRN), where one gene actively regulates the expression of another, co-expression does not imply a causal relationship, but rather a correlation in the transcriptional activity of the involved genes (see Figure 2.2). This phenomenon can reflect the action of a shared regulator, participation in the same biological pathway, or a structural proximity within the chromatin [52].

The identification of functional modules from co-expression data has been widely used in systems biology [53, 54]. Grouping genes with similar expression profiles allows for the inference of functional relationships, the identification of new biological pathways, and the prediction of unknown gene functions [55]. This approach is particularly useful in the study of complex diseases, where disruptions in the co-expression of certain genes may be associated with pathologies [56]. Computational methods such as biclustering make it possible to detect groups of co-expressed genes within specific subsets of conditions, providing a more detailed view of the underlying biological processes [57, 58].

2.2 Computational Foundations for Optimization

The formulation of problems as optimization tasks has proven to be an effective strategy in a wide range of bioinformatics domains, from omics sciences to drug design or disease diagnosis [59]. In these contexts, the goal is to find solutions that maximize or minimize certain criteria, represented through objective functions. This approach has led to the development of numerous computational methods capable of addressing highly complex problems [60], especially when the solution space is large or presents constraints that are difficult to satisfy using analytical techniques. This section introduces the fundamentals of optimization, as well as the main methods used for its resolution, providing the necessary foundation to understand evolutionary approaches and their application to inference and analysis tasks in computational biology.

2.2.1 Fundamentals of Optimization

Optimization focuses on finding the global extreme (maximum or minimum) of a function defined mathematically within a region of interest [61]. Given a set of feasible solutions S and an objective function $f : S \to \mathbb{R}$ that assigns a scalar value to each solution, the optimization problem consists of finding an element $x^* \in S$ that optimizes $f(x)$.

In the case of minimization, this problem can be expressed as shown in Equation (2.1):

$$f(x^*) \leq f(x), \quad \forall x \in S, \tag{2.1}$$

whereas the maximization counterpart is formulated in Equation (2.2):

$$f(x^*) \geq f(x), \quad \forall x \in S. \tag{2.2}$$

The choice between the minimization or maximization formulation depends on the improvement criterion adopted. Moreover, in many practical scenarios the search for x^* is subject to additional constraints, which can be expressed as equality conditions $g_i(x) = 0$ or inequality conditions $h_j(x) \leq 0$. These constraints define a feasible subset $S' \subseteq S$ within which the optimal solution must be found.

The nature of the search space S allows optimization problems to be classified into different categories. If $S \subseteq \mathbb{R}^n$, it is a continuous optimization problem, where the variables can take any real value within the allowed domain. If $S \subseteq \mathbb{Z}^n$, the problem is an integer optimization problem, restricting solutions to discrete values. A particular case occurs when $S \subseteq \{0, 1\}^n$, which defines a

binary optimization problem, typical in selection and assignment decisions. Finally, in heterogeneous optimization, some variables are continuous and others are discrete, combining both domains in the search for a solution.

Depending on the structure of $f(x)$ and the imposed constraints, different solution methods can be applied, ranging from analytical techniques based on derivatives to numerical algorithms and heuristics designed to tackle problems of great computational complexity [62].

2.2.2 Computational Methods for Optimization

Optimization problems can be solved using different computational approaches, which differ in their ability to guarantee optimality, their computational cost, and their applicability to different types of problems [60]. In general terms, optimization methods can be classified into three main categories: exact methods, heuristic methods, and metaheuristic methods.

Exact methods [63] guarantee the attainment of an optimal solution when the problem is solvable within the imposed computational limits. These methods are based on well-defined mathematical principles and can be applied to problems with known structures. Representative examples include:

- **Linear programming** [64]: Solves problems in which the objective function and constraints are linear. Algorithms such as the simplex method and mixed-integer programming are widely used in this context.
- **Dynamic programming** [65]: Decomposes the problem into smaller subproblems, solving them recursively and storing intermediate results to avoid redundant calculations.
- **Enumeration-based methods** [66]: Strategies such as branch and bound systematically explore the solution space to identify the best possible alternative.

Despite their ability to find optimal solutions, exact methods can become computationally expensive in large-scale problems, where the number of feasible solutions grows exponentially.

Heuristic methods [67] seek approximate solutions within reduced computational times, without guaranteeing global optimality. Their goal is to find acceptable solutions through simplified search strategies. Some common approaches include:

- **Greedy algorithms** [68]: They build a solution iteratively by choosing, at

each step, the locally optimal option without considering the global effect of the decision.

- **Local search** [69]: Modifies an initial solution iteratively by exploring neighboring solutions to improve solution quality.

Heuristic methods are often effective in problems where exact methods are computationally infeasible, although they do not always guarantee an optimal solution.

Metaheuristic methods [70] extend the search capabilities of heuristics through advanced strategies for exploration and exploitation of the solution space. These methods are designed for complex optimization problems and are often inspired by natural processes or physical models. Some of the most commonly used approaches include:

- **Evolutionary algorithms** [71]: Based on the theory of evolution, they apply selection, recombination, and mutation operators on a population of solutions to iteratively improve their quality. Examples include genetic algorithms and differential evolution.

- **Simulated annealing** [72]: Introduces probabilistic elements into local search to avoid getting stuck in local optima, allowing transitions to worse solutions with a certain probability that decreases over time.

- **Particle swarm optimization** [73]: Models the search for solutions as the collective movement of particles in the search space, guided by global and local information.

- **Ant colony optimization** [74]: Simulates the behavior of ants in the search for optimal paths, using artificial pheromones to reinforce the best solutions found.

Unlike simple heuristic methods, metaheuristics offer a better balance between exploration and exploitation, enabling them to reach high-quality solutions in complex problems with large search spaces.

Each of these approaches presents advantages and limitations depending on the problem's structure, the dimensionality of the search space, and the computational requirements. The choice of the most suitable method depends on a trade-off between solution accuracy and the computational feasibility of obtaining it.

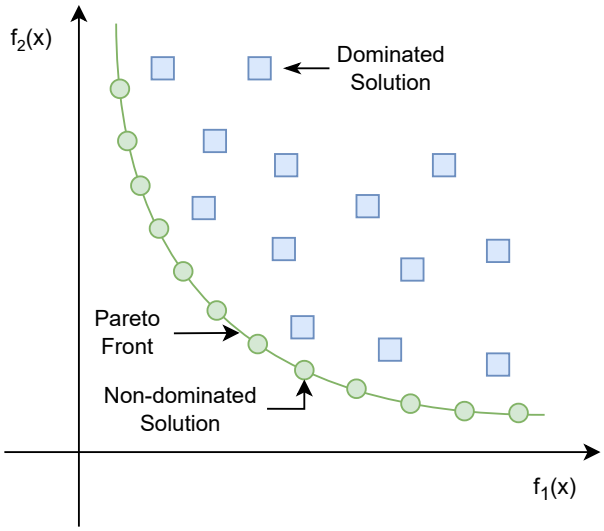


Figure 2.3: Pareto front in a two-dimensional multi-objective optimization problem. The figure exemplifies a case with two objectives to minimize, $f_1(x)$ and $f_2(x)$. The green points represent non-dominated solutions that form the Pareto front: none of them can be improved in one objective without worsening in the other. Conversely, the blue solutions are dominated, as there are other solutions better in at least one objective and no worse in the others.

2.2.3 Multi-Objective Optimization

In many optimization problems, the quality of a solution is not measured by a single criterion but by multiple objectives that may be in conflict with each other. In these cases, the goal is not to find a single optimal solution, but rather a set of solutions that represent different trade-offs among the competing objectives. This type of problem falls within the scope of multi-objective optimization [75], a natural extension of the computational optimization methods previously discussed.

Formally, a multi-objective optimization problem can be defined as shown in Equation (2.3):

$$\min_{x \in S} F(x) = (f_1(x), f_2(x), \ldots, f_m(x)), \tag{2.3}$$

where $F(x)$ is a vector of m objective functions that must be minimized simultaneously within the search space S. Unlike single-objective optimization, where there is a single optimal solution (or a small set of equivalent solutions), in multi-objective optimization there is not always a single solution that opti-

mizes all functions simultaneously. Instead, the concept of Pareto dominance is defined [76].

A solution $x^* \in S$ is said to be Pareto-optimal if there is no other solution $x \in S$ that dominates it. Formally, the dominance condition is expressed in Equation (2.4):

$$f_i(x) \leq f_i(x^*) \quad \forall i \in \{1, \ldots, m\}, \quad \text{and at least one inequality is strict}, \quad (2.4)$$

which means that x is at least as good as x^* in all objectives and strictly better in at least one.

The set of all Pareto-optimal solutions forms the Pareto front [75], which represents the boundary of solutions that cannot be improved in one objective without worsening at least another. The task in multi-objective optimization is not to select a single solution but to approximate this front with well-distributed solutions that offer different compromise alternatives (see Figure 2.3).

Computational Methods

To address these types of problems, different solution methods have been developed, which can be grouped into three main approaches [77]: aggregation-based methods, quality indicator-based methods, and Pareto dominance-based methods.

Aggregation-based methods [77] combine the objective functions into a single scalar function using weights or constraints, transforming the problem into a traditional single-objective optimization. However, this approach requires prior knowledge of the relative importance of each objective, which may be unfeasible or unrealistic in scenarios where the objectives are in conflict or the expert's preference is unknown.

Quality indicator-based methods [78] use preference functions to directly evaluate the contribution of each solution to the quality of the population set, avoiding the explicit use of dominance relations or the need for predefined weights. These approaches are particularly suitable for problems with a large number of objectives, where most solutions tend to be mutually non-dominated. A representative example is **IBEA (Indicator-Based Evolutionary Algorithm)** [78], which employs indicators such as the ε-indicator or the hypervolume to compare pairs of solutions and assign fitness values.

Finally, Pareto dominance-based methods [75] use dominance relations to establish a partial order among solutions, prioritizing those that are not dominated by any other in the population. Widely used examples include:

- **NSGA-II (Non-dominated Sorting Genetic Algorithm II)** [79] is an evolutionary algorithm based on non-dominated sorting. It is characterized by its elitist strategy, the preservation of diversity through crowding distance, and an efficient classification of solutions into Pareto fronts, reducing computational complexity compared to its predecessor.
- **NSGA-III (Non-dominated Sorting Genetic Algorithm III)** [80] is an extension of NSGA-II designed to handle problems with a large number of objectives. It introduces a reference point-based scheme, which allows for better distribution of solutions along the Pareto front and facilitates exploration in high-dimensional problems.
- **MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition)** [81] decomposes the multi-objective problem into multiple scalar subproblems, optimizing each independently but with information exchange between neighboring solutions. This strategy improves convergence and stability in the search for the Pareto front.
- **MOCell (Multi-Objective Cellular Genetic Algorithm)** [82] is an evolutionary algorithm based on a cellular structure where solutions evolve on a grid of defined neighbors. This approach improves the diversity of solutions and favors the exploration of the search space, avoiding premature convergence.
- **SPEA2 (Strength Pareto Evolutionary Algorithm 2)** [83] improves upon its predecessor through an advanced archiving mechanism and a dominance-based fitness assignment strategy. These improvements allow for the preservation of non-dominated solutions and maintain an adequate distribution on the Pareto front.

These methods make it possible to efficiently explore the search space in problems where no prior information is available about the relationship between the objectives, facilitating the identification of multiple optimal solutions that can be evaluated based on external criteria to the optimization process.

Quality Indicators

The evaluation of multi-objective optimization algorithms requires quality indicators that measure convergence to the true Pareto front, diversity of solutions, and coverage of the objective space:

- **Epsilon (EP)** [84]: Measures how much an approximation set needs to be translated in the objective space in order to weakly dominate a reference

Pareto front. It captures the worst-case approximation error and is Pareto compliant.

- **Generational Distance (GD)** [85]: Computes the average distance from each solution in the approximation set to the nearest point in the reference Pareto front. It quantifies how close the obtained solutions are to the optimal front, focusing on convergence.
- **PISAHypervolume (HV)** [86]: Calculates the volume of the objective space dominated by the approximation set and bounded by a reference point. It simultaneously reflects convergence and diversity, and is the only known unary Pareto-compliant indicator.
- **Inverted Generational Distance (IGD)** [85]: Computes the average distance from each point in a reference Pareto front to the closest solution in the approximation set. Unlike GD, it emphasizes both convergence and diversity, as all regions of the Pareto front must be well represented.
- **Inverted Generational Distance Plus (IGD+)** [87]: A modification of IGD that considers Pareto dominance when computing distances, avoiding misleading evaluations where dominated sets appear superior. IGD+ is weakly Pareto compliant and more reliable than IGD for performance comparisons.
- **Spread (SP)** [79]: Evaluates the extent of distribution of solutions along the Pareto front by measuring the range and uniformity of spacing. It complements convergence indicators by quantifying diversity.

These indicators provide complementary perspectives and are widely adopted in evolutionary multi-objective optimization, allowing a rigorous assessment of algorithmic performance.

2.2.4 Evolutionary Ensemble Learning

Ensemble learning is a machine learning technique that combines multiple base models with the goal of improving overall performance in classification, regression, or clustering tasks [88]. Instead of relying on a single model, an ensemble of models is trained and their predictions are aggregated using techniques such as majority voting, weighted averaging, or rule-based combination. This strategy often leads to more robust, generalizable, and accurate solutions, mitigating overfitting and leveraging the diversity among models.

When evolutionary algorithms are integrated into this context, the result is known as evolutionary ensemble learning, an approach that employs evolutionary processes to optimize different aspects of the ensemble, such as the selection

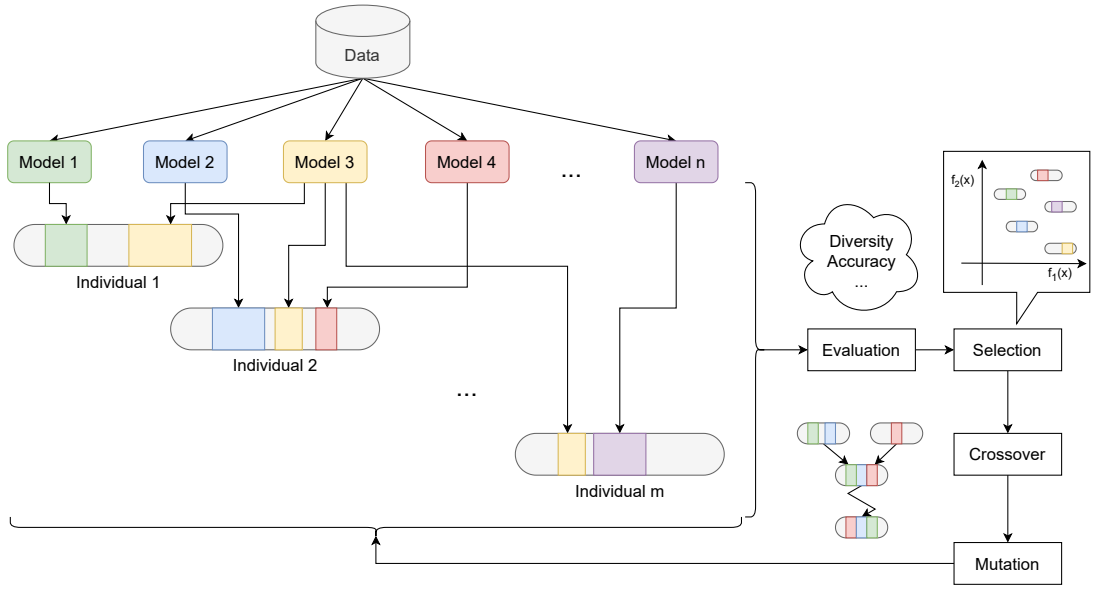


Figure 2.4: General schematic of an evolutionary ensemble learning approach. Starting from a dataset, multiple base models are trained and used as components to build the individuals of the population. Each individual represents a specific combination of models, combining them in different ways or even taking different configurations of the same models. These individuals are evaluated based on their performance and undergo an evolutionary process with selection, crossover, and mutation operators, which enables the generation of new combinations and progressively improves the quality of the ensemble.

of base models, weight assignment, generation of diversity, or combination of predictions [89]. Instead of building the ensemble statically, evolutionary algorithms allow the exploration of the space of possible model combinations, adapting the structure of the ensemble to the characteristics of the problem and the available data (see Figure 2.4).

This integration is especially advantageous in complex problems, where individual models exhibit heterogeneous behaviors and their performance varies significantly across different data subsets. In these contexts, there is no single model that consistently offers high performance, and the diversity of available approaches can be both an opportunity and a challenge. Choosing a specific model becomes a non-trivial task, particularly when the specialization of the various models is unknown and does not allow for prior identification of which one will be the most suitable for a given dataset. Evolutionary ensemble learning delegates this responsibility to an evolutionary process that adaptively selects, combines, and adjusts models, exploring multiple combinations and leveraging

those that best fit the particularities of the problem.

2.3 Computational Methods for Gene Expression Analysis

This section describes the computational methods developed to address two fundamental tasks in gene expression analysis: the inference of gene regulatory networks and the detection of co-expression patterns through biclustering. Both approaches enable the modeling of different aspects of the functional organization of the transcriptome based on expression data.

2.3.1 Inference of Gene Regulatory Networks

The inference of gene regulatory networks (GRNs) consists of reconstructing, from gene expression data, a model that describes the regulatory interactions between genes (see Section 2.1.1) [90]. Given a set of expression profiles measured under different conditions, tissues, or time points, the goal is to identify which genes regulate the activity of others, and under which mechanisms or dependencies these relationships manifest. This task enables the representation of the dynamics of the gene system as a directed network, where the nodes correspond to genes and the edges indicate regulatory interactions [43].

The inference process is based on the hypothesis that statistical dependencies between profiles may reflect underlying functional relationships [91]. Consequently, GRN inference is based on extracting dependency structures between variables from high-dimensional numerical data. These data usually represent the relative abundance of messenger RNA for each gene under different conditions and are typically organized in matrices where the rows correspond to genes and the columns to experimental conditions [92, 93].

Various computational approaches have been proposed for this task [17], among which three main approaches can be highlighted:

- **Correlation-based methods**: These methods infer regulatory relationships by analyzing the degree of association between gene expression profiles, using metrics such as Pearson correlation, Spearman correlation, or mutual information [94, 95]. Their main advantage lies in their conceptual simplicity and the low amount of data they require, which explains their widespread adoption. However, as they rely on symmetric measures, they tend to generate undirected networks and include indirect correlations, which can reduce the model's precision. To address this issue, strategies to

eliminate redundant or indirect relationships are commonly applied before building the final network [96].

- **Model-based methods**: This approach is based on defining a mathematical or probabilistic model that represents the dynamics of gene expression, such as Bayesian networks or differential equations [97, 98]. The parameters of the model are then fitted to infer the underlying regulatory network. The choice of model depends on the type of available data, and proper optimization is essential to obtain an accurate representation of gene interactions. These methods can incorporate temporal information and capture complex relationships between genes, although they often require more data and computational resources.

- **Machine learning-based methods**: These methods reformulate the inference of regulatory networks as a classification or regression problem [9, 50], using feature selection techniques to identify the most relevant regulatory genes for each target gene. Models are trained with the gene expression data to estimate the importance of each potential relationship, and then the network is built based on these scores. The choice of algorithm depends on the specific requirements, such as network directionality or computational complexity, and its accuracy depends both on the model assumptions and the implementation details.

The inference of GRNs presents various challenges that hinder the reliable reconstruction of networks. One of the main problems is the detection of indirect relationships, where two genes appear to be related due to a common regulator not explicitly considered [96]. Another significant challenge is the experimental noise present in expression data, which can arise from biological variability, measurement errors, or uncontrolled conditions, affecting the robustness of the methods [18]. Additionally, the high dimensionality typical of omics data, where the number of genes far exceeds the number of experimental conditions, introduces overfitting problems and necessitates the use of regularization or feature selection techniques [99].

Beyond these data-inherent challenges, there are also limitations associated with the inference tools themselves. On the one hand, the available methods often produce disparate results, making it difficult to validate the inferred networks and raising concerns about their reliability [15]. On the other hand, many techniques have domains of specialization, offering high performance only on certain datasets while their efficacy diminishes significantly on others [16]. This sensitivity is related to structural properties of the networks, such as their density, modularity, or degree distribution, which are unknown a priori in real scenar-

ios, complicating the initial choice of an appropriate method by the researcher. Finally, it should be noted that a large proportion of existing approaches focus on the mathematical optimization of the model, often neglecting the biological characteristics inherent to this type of network [17].

Despite these limitations, advances in computational methods and the development of robust ensemble-based approaches (see Section 2.2.4) promise significant progress in gene network inference. This makes GRN inference an active research field, in which experimental biological validation remains essential to confirm the in silico predictions obtained.

2.3.2 Co-Expression Detection via Biclustering

The analysis of gene co-expression aims to identify sets of genes that exhibit similar expression patterns under certain experimental conditions (see Section 2.1.3). One of the most widely used techniques for this purpose is biclustering, an extension of traditional clustering analysis that allows for the detection of local coherences in gene expression data [57].

Unlike traditional clustering, which groups only rows (genes) or columns (conditions) based on their global similarity, biclustering identifies subsets of genes that are correlated only under a specific subset of conditions [100, 101]. In other words, each bicluster represents a submatrix of the dataset where the included genes exhibit a coherent expression pattern limited to a particular subset of columns. This approach enables the capture of relationships that would not be detectable through classical clustering, especially when the correlations are contextual or conditional.

A fundamental aspect in biclustering is the type of pattern that one aims to identify [102]. The ability of an algorithm to identify one type of pattern or another depends on both the underlying mathematical model and the evaluation metric used. Algorithms can be oriented to detect:

- **Constant values**: all elements of the bicluster have similar values.
- **Constant values by row or by column**: variation is allowed in one of the two dimensions, while maintaining constancy in the other.
- **Coherent values**: coherence exists in both dimensions, according to models that are additive, multiplicative, or combined (additive-multiplicative).
- **Coherent evolutions**: these reflect shared trends (for example, genes that are activated or deactivated in parallel), without requiring exact matching

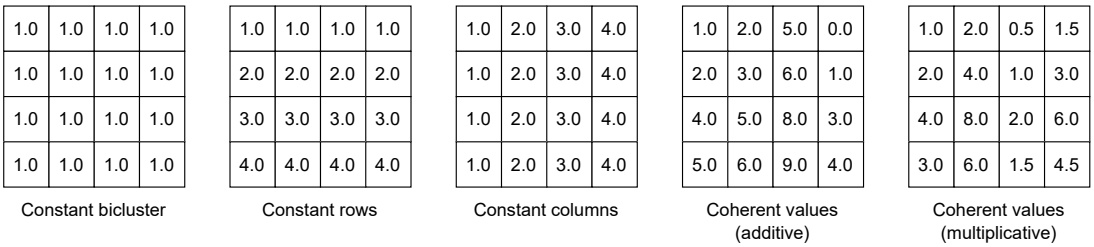


Figure 2.5: Examples of structural patterns detectable through biclustering. Each matrix shows a different type of pattern that can appear in subsets of rows and columns within a data matrix. From left to right: constant bicluster (all values are equal), constant rows (each row has a constant value), constant columns (each column maintains its constant value), additive coherence (values follow an additive pattern between rows and columns), and multiplicative coherence (values vary following a proportional relationship). These patterns reflect different types of relationships that biclustering algorithms aim to identify in the data.

of expression levels. This more flexible approach makes it possible to capture relevant biological relationships even when the magnitudes differ, as long as the relative trajectories are preserved.

Figure 2.5 illustrates some of the main patterns that conventional biclustering algorithms typically detect in data matrices.

Another key dimension in the design of biclustering algorithms is the structure of the biclusters with respect to the original matrix [57]. This can be analyzed from two main perspectives:

- **Overlap**:
  - **No overlap**: neither rows nor columns can belong to more than one bicluster.
  - **Row overlap**: a column can belong to only one bicluster, but rows may overlap.
  - **Column overlap**: a row can belong to only one bicluster, but columns may overlap.
  - **Total overlap**: both rows and columns can belong to multiple biclusters.
- **Coverage**:
  - **Partial coverage**: both rows and columns may remain outside any bicluster.

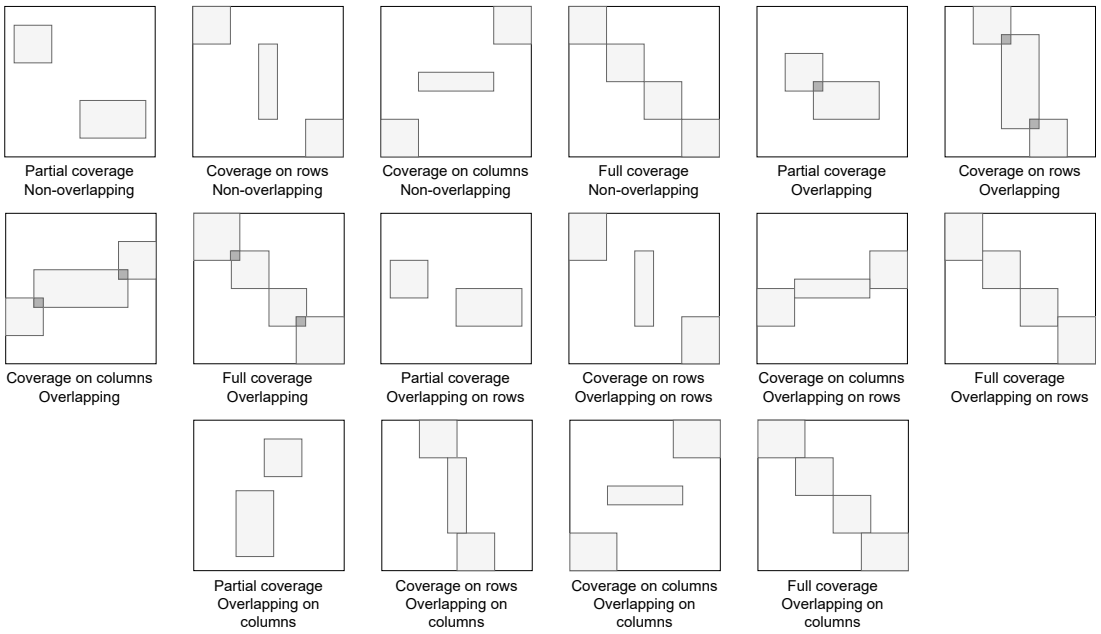


Figure 2.6: Possible combinations of coverage and overlap between biclusters. The figure shows the different scenarios that can arise in a biclustering solution according to the degree of coverage (partial or total) and the presence or absence of overlap between biclusters, whether in rows, columns, or both dimensions. These combinations determine the structural constraints that a biclustering algorithm may or may not allow, and directly affect the complexity of the problem and the interpretation of the results.

- **Row coverage**: all rows must be covered, but some columns may remain outside.
- **Column coverage**: all columns must be covered, but some rows may remain outside.
- **Total coverage**: all rows and columns must be covered by at least one bicluster.

Figure 2.6 illustrates the different structural configurations that biclusters can adopt based on their coverage and the degree of overlap allowed.

Some algorithms control overlap explicitly through penalties, thresholds, or selection mechanisms. Others allow free overlap or restrict it to a single dimension (for example, allowing overlap among conditions but not among genes). The assumed or allowed structure directly influences the type of solutions an algorithm can provide and must be considered when selecting the most suitable biclustering strategy.

From a computational standpoint, biclustering strategies can be organized into two broad categories depending on whether the search is guided by an explicit evaluation measure or not [57]:

- **Evaluation measure-based algorithms**: these methods use explicit objective functions that quantify the quality of the biclusters generated during the search process. Since the problem is NP-complete [102], these functions are usually optimized using heuristics or metaheuristics (see Section 2.2.2). The main subcategories are:
  - **Iterative greedy search**: incrementally build biclusters by applying locally optimal decisions. They use sequential coverage heuristics, selection based on marginal gain, or improvements in metrics such as variance or homogeneity. Although they do not guarantee optimality, they usually provide fast results with low computational cost [101, 103, 104].
  - **Stochastic search**: introduce random components into the iterative process to escape local optima. These strategies combine deterministic decisions with probabilistic moves, and may include perturbation phases, acceptance of worse solutions, or controlled random restarts [105, 106, 107].
  - **Nature-inspired metaheuristics**: simulate natural processes to efficiently explore the solution space. They include evolutionary algorithms, simulated annealing, particle swarm optimization, artificial immune systems, among others. These methods are particularly well-suited to finding multiple high-quality biclusters in complex and multimodal search spaces [108, 109, 110].
  - **Clustering-based approaches**: combine classical one-dimensional clustering algorithms (genes or conditions) with complementary strategies that allow the clustering to be extended to the second dimension. In some cases, they rely on dimensionality reduction techniques followed by hierarchical or partition-based clustering [111, 112].
- **Non-evaluation measure-based algorithms**: in this group, the search for biclusters is not guided by an explicit quality function. Instead, they use probabilistic models, graphical representations, or algebraic transformations to define and extract relevant structures. The main strategies are:
  - **Probabilistic models**: model the data as realizations of random variables with certain dependencies. Techniques such as Bayesian sampling, generative models, or statistical inference are used to identify

structured subsets in the data. These strategies allow the incorporation of prior knowledge, handling of uncertainty, and extensions to contextual information [113, 114, 12].

- **Graph-based approaches**: represent the expression matrix as a bipartite graph or multigraph, where the nodes are genes and conditions, and the edges indicate relevant relationships. Biclusters are identified as dense or heavy subgraphs according to some structural or statistical criterion [115, 116, 117].
- **Linear algebra and factorization**: use techniques such as singular value decomposition (SVD), non-negative matrix factorization (NMF), or iterative methods to identify local structures in the expression matrix. These approaches offer a geometric interpretation of biclustering and enable the discovery of non-trivial patterns [118, 13, 119].
- **Row and column reordering**: reorganize the expression matrix through optimal permutations that reveal coherent submatrices. Strategies include orderings based on probabilistic models or formulations of classical combinatorial problems such as the traveling salesman problem [120, 121].

Each of these strategies presents advantages and limitations depending on the type of patterns sought (constant, additive, multiplicative, evolutionary), the permitted structure (overlap, exhaustiveness), and the computational requirements. The variety of existing approaches reflects the richness of the problem and its relevance in the analysis of gene expression data.

However, biclustering also presents a number of inherent challenges. Firstly, the search for coherent submatrices is a combinatorial problem that grows exponentially with the size of the data, making its exact resolution difficult [102]. Moreover, existing methods must contend with the presence of noise in the data and high dimensionality, which complicates the identification of robust patterns [122]. There is also a great diversity of coherence criteria, such as correlation, profile similarity, or absolute constancy, which has led to a wide variety of algorithms and complicates their comparison [19]. Finally, the biological validation of the extracted biclusters requires establishing their functional relevance and interpreting them in an experimental context, which usually involves the use of external knowledge bases and functional annotation tools [123].

Although many of these algorithms are specifically designed to work with gene expression data, their foundations are primarily mathematical and statistical, without explicitly incorporating biological domain knowledge [101]. This

disconnection can limit the ability of the methods to capture functionally relevant patterns, thus complicating their biological interpretation and reducing their applicability in real-world genomic analysis contexts. Additionally, most algorithms do not converge to an optimal number of biclusters automatically, instead delegating this decision to the researcher [124, 125, 113], who must define this parameter a priori or determine it from multiple runs, introducing an additional degree of subjectivity and complexity into the analytical process.

Finally, it is worth emphasizing that biclustering constitutes a fundamental tool in the exploratory analysis of transcriptomic data, offering a more nuanced view of the functional organization of the transcriptome compared to global clustering approaches.

Chapter 3

State of the art

This chapter provides an in-depth review of the state of the art related to the two fundamental methodological pillars of this thesis: gene regulatory network inference and gene co-expression biclustering. Both problems have been extensively addressed in the literature through a variety of approaches, which differ in their theoretical foundations, objectives, and practical limitations. The aim of this review is to identify the main research trends, critically compare their most representative proposals, and delineate the methodological space in which this thesis is positioned.

First, the different techniques used to infer gene regulatory networks from expression data are analyzed, ranging from individual methods to consensus-based approaches. Next, the strategies applied in co-expression biclustering are explored, distinguishing between conventional methods and those based on evolutionary algorithms, with particular attention to the encodings used and their ability to represent complete solutions to the problem. This review not only contextualizes the contribution of this thesis in relation to previous work, but also highlights current gaps and opportunities for improvement that motivate its development.

3.1 Gene Regulatory Networks Inference

Multiple methodologies exist for inferring gene regulatory networks from expression data. As explained in Section 2.3.1, this task has been approached from very diverse perspectives, resulting in an ecosystem of techniques with different foundations and assumptions [17]. This methodological heterogeneity has led to significant variability in the results obtained, making it difficult to validate the inferred networks and creating uncertainty about which approach is most appropriate in each scenario.

As a result of these limitations, and in the absence of a clear consensus on a dominant technique, more robust proposals have emerged that integrate multiple strategies. These consensus approaches aim to leverage the complementary strengths of individual methods to improve the accuracy, stability, and reliability of the resulting networks. This section reviews both the main individual techniques and the methods that combine their predictions.

3.1.1 Individual techniques

Below are some of the main individual techniques used to infer gene regulatory networks from expression data. These methodologies tackle the problem from various theoretical and computational perspectives, applying criteria such as statistical dependency, mathematical modeling, or machine learning to identify regulatory relationships between genes:

- **ARACNE** (Algorithm for the Reconstruction of Accurate Cellular NEtworks) [96]: It employs an information-theoretic approach for the reverse engineering of transcriptional networks from microarray data. Initially, ARACNE identifies candidate interactions by estimating the mutual information (MI) between pairs of gene expression profiles, applying a statistical significance threshold to retain only the strongest associations. Subsequently, the algorithm applies the Data Processing Inequality (DPI) to remove most indirect interactions. Specifically, for each triplet of genes where all pairwise MI values exceed the threshold, ARACNE eliminates the edge with the smallest MI value, assuming it represents an indirect interaction. This method is designed to scale to the complexity of regulatory networks in mammalian cells, with a computational complexity of $O(N^3 + N^2M^2)$, where N is the number of genes and M is the number of samples.
- **BC3NET** (Bagging C3NET) [126]: It is based on the bootstrap aggregation (bagging) technique applied to the C3NET algorithm [127]. The BC3NET process involves generating an ensemble of independent bootstrap datasets

from the original dataset. For each of these bootstrap datasets, a network is inferred using the C3NET algorithm. These inferred networks are then aggregated to form a weighted network, where edge weights represent the frequency with which a connection between a pair of genes appears across the ensemble of networks. Finally, statistical hypothesis testing is applied to these edge weights to determine the significance of the connections, thus eliminating the need for manually selecting a threshold. The computational complexity of BC3NET is $O(B|n^2)$, where B is the number of bootstraps and n is the number of genes. This ensemble approach aims to reduce the variance of the estimates and address issues such as noise and outliers in expression data.

- **C3NET** (Conservative Causal Core NETwork) [127]: This algorithm is based on the estimation of mutual information (MI) values combined with a maximization step to efficiently exploit causal structural information in the data. The algorithm begins by removing non-significant connections between pairs of genes through statistical significance testing of the MI values. Then, for each gene, it identifies the connection to its neighbor with the highest mutual information value. Finally, it constructs an adjacency matrix where a connection is established if the maximum MI value for a given gene corresponds to another gene. The computational complexity of C3NET is $O(n^2)$, where n is the number of genes, making it one of the fastest algorithms. The C3NET approach focuses on inferring the "conservative causal core" of the network, that is, the strongest interactions, rather than the full network.

- **CLR** (Context Likelihood or Relatedness network) [94]: This algorithm infers transcriptional regulatory networks based on an extension of the relevance network approach. Like relevance networks, CLR uses mutual information (MI) to quantify the similarity between gene expression profiles, where a high MI suggests a potential regulatory interaction. The key innovation of CLR lies in its adaptive background correction step. After computing the MI for all possible regulator–target gene pairs, CLR estimates the statistical significance of each MI value within its network context. This is achieved by comparing the MI of a specific pair to the distribution of MI values for all other pairs involving the same regulator or the same target gene. The most likely interactions are those whose MI values lie significantly above these background distributions, allowing many spurious correlations and indirect influences to be filtered out. The algorithm computes a joint significance score based on the z-scores of the pairwise MI relative to the marginal MI distributions for each individual gene.

- **CMI2NI** (Conditional Mutual Inclusivity principle-based Network Inference) [95]: This method uses the concept of conditional mutual inclusive information (CMI2) to quantify causal associations between genes, aiming to overcome the common issues of mutual information (MI) overestimation and conditional mutual information (CMI) underestimation. CMI2 is defined as the average Kullback–Leibler (KL) divergence [128] between the joint probability distribution of three variables (two genes and a conditioning variable) and the interventional probability distributions obtained by removing the edge in each direction. For GRN inference, CMI2NI combines CMI2 with the path-consistency (PC) algorithm to eliminate indirect regulations from an initially complete graph. The algorithm starts by generating a fully connected graph and then recursively removes edges with low initial MI values and subsequent low-order CMI2 values. CMI2 is efficiently computed under the assumption of a Gaussian distribution for gene expression data using covariance matrices.

- **GENIE3** (GEne Network Inference with Ensemble of trees) [9]: This method decomposes the prediction of a regulatory network among p genes into p different regression problems. In each of these problems, the expression pattern of one gene (the target gene) is predicted from the expression patterns of all other genes (input genes), using tree-based ensemble methods such as Random Forests (**GENIE3_RF**) or Extra-Trees (**GENIE3_ET**). The importance of an input gene in predicting the expression pattern of the target gene is taken as an indication of a potential regulatory link. These potential regulatory links are then aggregated across all genes to produce a ranking of interactions, from which the full network is reconstructed. GENIE3 makes no assumptions about the nature of gene regulation, can handle combinatorial and non-linear interactions, produces directed GRNs, and is fast and scalable. Its computational complexity is on the order of $O(pTKN \log N)$, where p is the number of genes, T is the number of trees, N is the training sample size, and K is a main parameter of the tree-based methods.

- **GRNBOOST2** (Gene Regulatory Network inference using gradient BOOSTing) [129]: This is an efficient algorithm for gene regulatory network (GRN) inference that uses gradient boosting and builds upon the GENIE3 architecture. Like GENIE3, it belongs to the class of regression-based GRN inference methods. For each gene in the dataset, a tree-based regression model is trained to predict its expression profile using the expression values of a set of candidate transcription factors (TFs). Each model produces a partial GRN with regulatory associations from the most predictive TFs

toward the target gene. All regulatory associations are then aggregated and ranked by importance to generate the final GRN output. GRNBoost2 employs a regularized stochastic variant of gradient boosting machines (GBMs), equipped with a heuristic early stopping strategy based on out-of-bag improvement estimates. This early stopping is triggered when the average of the last n improvement values falls below zero. GRNBoost2 is implemented within the Arboreto framework[1], which leverages Dask [130] for parallel computation, allowing the inference process to scale to large datasets. The independence of the regression tasks for each target gene makes the algorithm highly parallelizable. GRNBoost2 stands out for its efficiency, using shallower decision trees and building significantly fewer trees than GENIE3, thanks to the bias-reducing effect of gradient boosting and the early stopping mechanism.

- **GRNVBEM** (Gene Regulatory Network inference using Variational Bayesian Expectation-Maximization algorithm) [131]: This method performs gene regulatory network (GRN) inference from time-series and pseudotime data by employing a first-order autoregressive moving average model (AR1MA1) to capture noisy gene expression dynamics. Computationally, the method relies on a variational Bayesian expectation-maximization (VBEM) framework to infer the GRN. Within this framework, the binary variables describing the network topology are treated as latent variables. VBEM optimizes a free-form distribution over the latent variables and model parameters to approximate the posterior distribution by maximizing a lower bound of the marginal log-likelihood. To enable an analytical solution, GRNVBEM adopts a conjugate model with Gaussian priors over the latent variables and a scaled Inverse-Gamma distribution for the parameters. Due to the complexity of computing the marginal likelihood, a fixed-point approximation for the variance scale is used, based on the MAP estimates of the latent variables and the weights learned in previous VBEM iterations. The inference process involves the sequential application of learning rules to update the posterior hyperparameters until a convergence criterion is met.

- **INFERELATOR** (regression and variable selection to identify transcriptional influences on genes) [10]: This method integrates genomic annotation information and gene expression data, both from steady-state and time-series conditions, to identify transcriptional influences on genes. Inferelator uses regression and variable selection techniques, specifically L1 regression (LASSO), to produce parsimonious and predictive models. As a preprocessing step prior to network inference, the algorithm may em-

[1] Available at `https://arboreto.readthedocs.io/`

ploy an integrated biclustering method called cMonkey to group genes and conditions based on coherence in expression data, co-occurrence of cis-regulatory motifs, and functional associations, with the aim of identifying putatively co-regulated gene modules. Inferelator also models interactions between transcription factors (TFs) and environmental factors by incorporating functions of the minimum of two variables into the regression design matrix. The selection of the optimal model for each gene or bicluster is performed using cross-validation (CV) to choose the L1 shrinkage parameter that minimizes the prediction error.

- **JUMP3** (jump trees) [50]: Hybrid approach for the inference of gene regulatory networks (GRNs), combining a dynamic model of gene expression with a non-parametric decision tree-based method to reconstruct the network topology. Jump3 relies on a formal on/off model of gene expression, where the transcription rate of a gene switches between two levels depending on whether its promoter is active or inactive. For each target gene, Jump3 learns a model in the form of an ensemble of decision trees, referred to as jump trees, which predict the promoter state at any given time based on the expression levels of potential regulators at that same moment. The construction of each jump tree is performed greedily in a top-down manner, partitioning the set of time points based on tests over the expression levels of candidate regulators. Unlike standard decision trees, which split data by minimizing the entropy of the output variable, jump trees split by maximizing the likelihood of the gene expression observations, using the marginal likelihood of the node's dynamic model as the splitting criterion. To prevent overfitting, Jump3 builds an ensemble of such jump trees using an adaptation of the Extra-Trees procedure, which randomizes the test at each decision node. Finally, an importance score is derived for each candidate regulator, quantifying its relevance for predicting the promoter state of the target gene, based on the increase in likelihood produced by splits in the trees where the regulator is involved.

- **KBOOST** (kernel PCA regression and gradient boosting to reconstruct gene regulatory networks) [132]: Method for fast and scalable inference of gene regulatory networks (GRNs) that employs a combination of kernel principal component regression (KPCR), boosting, and Bayesian model averaging (BMA). The algorithm takes gene expression data as input, optionally including prior TF-target interactions, and for each gene builds a predictive model based on the kernel principal components (KPCs) of the expression of subsets of transcription factors (TFs), using an RBF kernel function to capture nonlinear relationships. Through a gradient boosting process with

greedy model selection, KBoost constructs an ensemble of KPC-based models by iteratively selecting the TFs with the highest posterior distributions to predict gene expression and its residuals. Finally, the posterior probabilities of the explored models are combined using BMA to estimate the GRN, allowing the incorporation of prior knowledge as a Bayesian prior. KBoost has shown competitive performance and significantly faster runtimes compared to other GRN inference methods.

- **LEAP** (Lag-based Expression Association for Pseudotime-series) [133]: Algorithmic technique designed to construct gene networks from single-cell RNA sequencing (scRNA-Seq) data, taking into account potential time delays. Unlike methods based on simultaneous correlation, LEAP uses the estimated pseudotime of cells to order them along a temporal trajectory. It then computes the maximum correlation between the expression of gene pairs by considering different time windows with possible lags. This maximum correlation is used as a measure of co-expression strength, allowing LEAP to capture directional and potentially regulatory relationships between genes that might be overlooked by methods that only consider simultaneous associations. The algorithm also includes a function to estimate the false discovery rate (FDR) in order to assess the statistical significance of the detected associations.

- **LOC-PCA-CMI** (Local Path Consistency Algorithm based on Conditional Mutual Information) [134]: Method for inferring the structure of GRNs that follows a divide-and-conquer strategy. Initially, the method identifies overlapping local clusters of genes based on the top n highly co-expressed edges, determined through Pearson correlation analysis with false discovery rate (FDR) correction. Then, for each local cluster, the PCA-CMI algorithm [11] is applied to infer the structure of the local subnetwork by repeatedly removing uncorrelated edges, from low- to high-order dependencies. Finally, the global structure of the GRN is obtained by assembling the inferred local network structures, averaging the edge weights. This approach enables Loc-PCA-CMI to handle relatively large datasets while benefiting from the accurate structure inference provided by PCA-CMI on small gene subnetworks.

- **MEOMI** (Mixed Entropy Optimizing context-related likelihood Mutual Information) [135]: Method for GRN construction based on the computation of mutual information through the combination of James–Stein entropy estimation [136] and Bayesian estimation with a Dirichlet prior distribution [137]. A context-related likelihood algorithm (based on CLR [94]) is then applied to optimize the mutual information matrix, obtaining an

initial network by eliminating indirect relationships. This network is iteratively refined by computing conditional inclusive mutual information (CMI2), which considers the influence of multiple genes, and by applying a path consistency algorithm with dynamic thresholds to progressively remove redundant edges. This process leads to a more accurate final GRN. MEOMI aims to overcome the limitations of mutual information and conditional mutual information in order to infer direct regulatory relationships with greater accuracy.

- **MRNET** (Minimum Redundancy NETworks) [138]: Computational method for inferring gene networks from microarray data, based on the maximum relevance/minimum redundancy (MRMR) principle, an information-theoretic feature selection technique. MRNET extends this feature selection principle to networks in order to infer dependency relationships between genes. The MRNET strategy formulates the network inference problem as a series of supervised gene selection procedures, where each gene plays the role of a target output. For each target gene, the MRMR principle is applied to select a set of genes that have high mutual information with the target (maximum relevance) and are mutually minimally redundant. For each gene pair Xi, Xj, MRMR returns two scores, and the score for the pair is computed by taking the maximum of these two values. A connection between Xi and Xj is inferred if this score exceeds a given threshold. MRNET has proven to be competitive with other information-theoretic inference methods such as CLR and ARACNE in experiments using synthetically generated microarray data. The computational complexity of MRNET lies between $O(n^2)$ and $O(n^3)$, depending on the number of features selected at each step. It should be emphasized that, like other mutual information-based methods, MRNET cannot determine the directionality of interactions.

- **MRNETB** (Minimum Redundancy NETworks using Backward elimination) [139]: This is an improved version of the MRNET network inference method. The main enhancement of MRNETB lies in its variable selection strategy. While MRNET uses forward selection to identify a set of maximally independent neighbors for each variable, MRNETB employs a backward selection strategy followed by sequential replacement. This new neighbor selection strategy is implemented with the same computational cost as forward selection. MRNETB has shown significantly better performance than MRNET, regardless of the mutual information estimation method used. In comparative evaluations with other information-theoretic algorithms, such as CLR and ARACNE, MRNETB performed comparably to CLR and signifi-

cantly better than ARACNE.

- **NARROMI** (Noise And Redundancy reduction technology by combining Recursive Optimization and Mutual Information) [140]: Technique for GRN inference that aims to improve accuracy by combining recursive optimization based on ordinary differential equations (RO) with mutual information (MI) from information theory. Initially, MI is used to detect and eliminate noisy regulations with low pairwise correlations. Then, the RO algorithm is applied to progressively exclude redundant regulations originating from indirect regulators, while also being capable of determining regulatory directions without prior knowledge of the regulators. Finally, the regulatory strengths inferred by RO and the MI correlations are integrated to account for both linear and nonlinear dependencies between regulators and target genes.

- **NONLINEARODES** (NON-LINEAR Ordinary Differential EquationS) [98]: Method for GRN inference based on a nonlinear ordinary differential equation (ODE) framework to model the dynamics of gene regulation. This approach jointly leverages time-series and steady-state data to more accurately capture the transcriptional and translational processes among genes. The method decomposes the GRN inference problem into independent regression tasks for each target gene, where a nonlinear function is learned to describe the temporal evolution (or steady-state behavior) of that gene as a function of its potential regulators. To determine the relevance of candidate regulatory links, a scoring strategy based on gradient boosting trees is employed, specifically using XGBoost [141]. Finally, all putative regulatory interactions are ranked according to their importance scores to reconstruct the GRN.

- **PCA-CMI** (Path Consistency Algorithm based on Conditional Mutual Information) [11]: Algorithm that combines the Path Consistency Algorithm (PCA) and Conditional Mutual Information (CMI) to evaluate the conditional dependence between gene pairs, thereby enabling the detection of nonlinear relationships that may be overlooked by linear correlation-based methods. PCA-CMI starts with a complete graph in which all genes are interconnected and iteratively removes edges that represent (conditional) independence relationships, beginning with lower-order dependencies until reaching a graph that represents the inferred network. This process, based on the computation of CMI from the covariance matrices of gene expression data under the assumption of a Gaussian distribution, allows PCA-CMI to distinguish between direct or causal interactions and indirect associations. The method has demonstrated superior performance com-

pared to other approaches in evaluations using benchmark datasets such as those from the DREAM challenge [142].

- **PCIT** (Partial Correlation coefficient with Information Theory) [143]: Algorithm for gene network reconstruction that combines the concept of partial correlation coefficient with information theory to identify significant associations between genes. The method operates in two steps: first, it computes the partial correlation coefficients for each triplet of genes; second, it applies the Data Processing Inequality (DPI) theorem to determine a local tolerance level (ε) based on the average ratio between the partial correlation and the direct correlation. A connection between two genes is considered significant if the magnitude of their direct correlation is greater than the tolerance level multiplied by the magnitude of the partial correlation with a third gene. This strategy allows PCIT to identify moderate yet meaningful associations, being more sensitive than fixed-threshold methods when detecting interactions involving genes with low variability. It uses data-driven local tolerance thresholds instead of arbitrary global cutoffs.

- **PIDC** (Partial Information Decomposition and context) [144]: This algorithm is designed to infer GRNs from single-cell transcriptomic data using multivariate information measures. The method is based on partial information decomposition (PID) to explore statistical dependencies among gene triplets. For each gene pair, PIDC computes the proportional unique contribution (PUC) [144], which represents the proportion of mutual information explained by unique information in the context of other genes. Finally, similarly to the CLR algorithm [94], PIDC incorporates network context by estimating an empirical probability distribution of PUC scores for each gene, enabling the identification of the most significant interactions per gene and overcoming the limitations of global thresholds.

- **PLSNET** (PLS-based gene NETwork inference method) [145]: Ensemble approach that uses partial least squares (PLS) regression for feature selection. The method decomposes the GRN inference problem into individual subproblems for each target gene, where the goal is to identify relevant regulatory genes through PLS-based feature selection applied repeatedly on random subsets of potential regulators. A statistical technique is then used to refine the predictions, assigning greater weight to regulatory genes that influence multiple target genes (“hub” genes).

- **PUC** (Proportional Unique Contribution) [144]: This is the core metric of the PIDC algorithm [144], although its raw value can be considered as an independent GRN inference method. Its computation focuses on quantify-

ing the average proportion of mutual information (MI) between two genes (X and Y) that is explained by the unique information they share, considering the context of all other genes (Z) in the network. For each gene pair X and Y, the ratio between the unique information they share conditional on a third gene Z and their total mutual information is computed, and this value is summed over all other genes Z in the network. A high PUC score between two genes suggests a more direct or specific functional relationship as opposed to a redundant one involving other genes. Results on simulated data indicate that the proportion of mutual information explained by the unique contribution tends to be higher between connected gene pairs.

- **RSNET** (Redundancy Silencing and Network Enhancement Technique) [146]: GRN inference method designed to address the challenge of distinguishing direct from indirect interactions. The method initially uses mutual information (MI) to define a search space of putative regulators and rank genes based on their dependency. It then applies a constraint-based recursive optimization process, in which genes with high dependency are retained in the model while redundant connections, including weak and indirect ones, are iteratively removed.

- **TIGRESS** (Trustful Inference of Gene REgulation with Stability Selection) [97]: Method for GRN inference that formulates the problem as a sparse regression task and employs the Least Angle Regression (LARS) feature selection method combined with stability selection. TIGRESS stood out in the DREAM5 challenge [142], where it was ranked among the top methods and recognized as the best linear regression-based approach. The method introduces a novel scoring technique for stability selection, called the "area score" ($s_{area}(t, g)$), which computes the area under the selection frequency curve of a transcription factor (TF) for a target gene (TG) up to L steps of LARS, proving to be more robust and accurate than the original score. The key parameters of TIGRESS include the number of runs R, the number of LARS steps L, and the parameter α that controls the random re-weighting of the expression data.

As mentioned in Section 1.1, these inference techniques exhibit a clear disparity in their results [15], which affects their reliability and increases the uncertainty for researchers when choosing an appropriate method for their dataset. Moreover, the existence of specialization domains has been demonstrated, where certain techniques show limited effectiveness on specific subsets of networks [16], along with a strong emphasis on mathematical optimization that often overlooks the inherent biological characteristics of these networks [17]. While

mathematical rigor is essential, the hypothesis of this Thesis posits that integrating biological considerations into the consensus of these tools could significantly enhance inference quality, making it more coherent and biologically plausible.

3.1.2 Consensus approaches

In order to address the disparity in the results of the techniques discussed in the previous section and to improve the robustness of gene regulatory network inference, several consensus-based approaches have been proposed in the literature:

- **AGRN** (Accurate Gene Regulatory Network inference) [147]: It employs an ensemble learning approach that combines Random Forest Regressor (RFR), Extra Tree Regressor (ETR), and Support Vector Regressor (SVR) to infer gene regulatory networks. The consensus is based on assigning gene importance scores, using SHAP values for RFR and ETR, and coefficients with iterative subsampling for SVR. AGRN determines the influence of each method on the final outcome through a grid search. This process involves exhaustively evaluating AGRN's performance on a benchmark dataset using multiple combinations of weights assigned to RFR, ETR, and SVR. For each weight combination, both AUROC and AUPR are computed, and the combination that yields the highest performance metrics is selected.
- **EnGRaiN** [148]: Supervised ensemble learning methodology that constructs gene regulatory networks by combining predictions from multiple inference methods. The consensus criterion is established by training a machine learning model on a small training dataset containing edges with known presence or absence. This model implicitly learns to weight the predictions of the different methods, assigning greater importance to those whose predictions align more closely with the ground truth during training. The role of the trained model is to predict the probability of existence for each edge in the consensus network based on the scores provided by the individual methods, thereby achieving an optimized combination of the various predictions instead of relying on a simple average.
- **EnsInfer** [16]: It addresses the problem of GRN inference through a non-homogeneous ensemble approach. The consensus criterion in EnsInfer is based on training a Naive Bayes classifier that combines the predictions from multiple individual inference methods. Each individual technique generates confidence scores for every possible regulatory edge. These scores serve as input features for the Naive Bayes model, which learns to predict whether a regulatory edge exists or not. To achieve the best results, EnsInfer integrates the predictions from all individual techniques that pass a

statistical normality test on the training data, specifically those whose output distribution exhibits positive kurtosis. This process allows EnsInfer to leverage the diversity of the underlying methods, weighting their contributions to produce a more robust and accurate consensus prediction of the regulatory network.

- **Fujii - Weighted Consensus Algorithm** [149]: The weighted consensus method proposed by Fujii et al. is a technique for inferring gene regulatory networks that aggregates predictions from multiple individual inference methods, assigning each a weight optimized through a linear programming formulation. This optimization process is based on a training dataset used to determine the reliability of each method. The core consensus criterion consists of maximizing the number of correctly predicted edges whose confidence score is at least ε higher than that of incorrectly predicted edges within the same training dataset.

- **Peignier - Ensemble** [150]: The methodology proposed by Peignier et al. for gene regulatory network inference is based on ensemble learning, where a genetic algorithm is used to explore the space of combinations of a set of base GRN inference methods implemented in the GReNaDIne library [151]. The goal of the genetic algorithm is to evolve a population of candidate ensembles (subsets of the base methods) in order to maximize their fitness, defined as the AUROC score of the inferred GRN. Subsequently, frequent itemset mining is applied to identify the subsets of base methods most frequently selected by the genetic algorithm to form the final ensembles. The predictions of the base methods within each ensemble are integrated using a z-score standardized score averaging scheme.

The main objective of all these consensus approaches is to leverage the complementarity among different individual inference techniques in order to achieve more robust and accurate results. Nevertheless, to justify the research gap addressed by this Thesis, it is important that each of these methodologies provides answers to the following questions:

Q1 : Does the methodology integrate a sufficiently broad set of individual techniques to benefit from their diversity and complementarity?

Q2 : Is the methodology designed to be configurable in the activation of individual techniques and sufficiently adaptable to incorporate new inference methods as existing approaches evolve?

Q3 : Is the ensemble process conditioned by external references, such as known interactions or gold standards, that may limit its applicability in real-world

contexts where the underlying network is largely unknown?

Q4 : Does the methodology leverage contextual knowledge from the biological domain to enrich and guide the inference process?

Q5 : Has the methodology been evaluated on a dataset that is sufficiently broad and diverse to demonstrate that it overcomes the limiting specialization of individual inference techniques?

Table 3.1: Assessment of consensus methods based on five key questions

Question	**AGRN**	**EnGRaiN**	**EnsInfer**	**Fujii**	**Peignier**
Q1: Methods used	3	12	11	9	17
Q2: Configurable	No	Partial	Partial	No	No
Q3: Uses gold standard	Yes	Yes	No	Yes	Yes
Q4: Contextual knowledge	No	No	No	No	No
Q5: Broad evaluation	No	No	Yes	No	No

Table 3.1 provides answers to each of these key questions for the different consensus methodologies analyzed. Several relevant conclusions can be drawn from this table. First, it is worth noting that only the approach proposed by Peignier integrates a significant number and diversity of individual techniques, reaching a total of 17 base methods. Second, none of the methodologies offers a readily configurable architecture, as both EnGRaiN and EnsInfer would require retraining their models in the event of incorporating new inference techniques.

Third, most of these strategies rely on ensemble criteria that aim to minimize the error with respect to gold standards. This significantly limits their applicability in real-world scenarios, where the underlying network to be inferred is largely unknown. Moreover, this dependency introduces a considerable bias in the comparative experiments carried out in this Thesis. In fact, by optimizing directly for metrics specifically designed to evaluate accuracy against those same external references, these methods gain an unfair advantage over the methodologies proposed here. The latter do not rely on gold standard networks during the inference process, precisely because they are aimed at discovering real networks with potential clinical applicability.

Furthermore, none of the evaluated strategies takes advantage of contextual information from the biological domain, which represents a missed opportunity to improve and guide gene inference towards more robust and biologically meaningful solutions. Finally, it should be pointed out that the evaluation of these

methodologies has been mostly limited to simulated datasets, generally originating from a single source such as the DREAM challenges [142], which prevents ensuring the elimination of specialization domains of individual techniques.

Taken together, these results highlight the need to develop new consensus methodologies that overcome current limitations. Instead of focusing on minimizing error with respect to gold standards, which are not always available and are not necessarily representative, these strategies should prioritize the identification of biologically plausible regulatory networks, guided by contextual domain knowledge. In addition, they should feature a flexible architecture that allows adaptation to future advances in inference techniques, and be evaluated on diverse datasets, both real and synthetic, to ensure their applicability in real-world scenarios.

3.2 Co-expression Biclustering

The identification of groups of co-expressed genes is a common task in gene expression data analysis, as it enables the discovery of functional structures underlying complex biological processes. As explained in Section 2.3.2, this task has been approached from a wide variety of perspectives, giving rise to a broad range of methods and approaches [123]. This diversity stems primarily from the combination of different computational strategies, coherence criteria, specific patterns to be detected, and structural constraints related to overlap and coverage in the resulting biclusters.

However, to clearly define the state of the art in this study, the analysis focuses on two main groups. On the one hand, well-established techniques are considered, which are widely accepted in the specialized literature due to their robustness and proven effectiveness under various experimental conditions. On the other hand, the focus is placed on approaches closely related to the method proposed in this thesis, specifically those algorithms that incorporate evolutionary optimization processes in the search for biclusters.

3.2.1 Conventional methods

The following is a list of conventional biclustering methods that have been extensively validated and referenced in the specialized literature:

- **Bibit** (Bit-Pattern Biclustering Algorithm) [14]: Although it is a biclustering algorithm specifically designed for binary data, it has often been applied to continuous data following normalization and binarization processes. Its

operation is based on two main phases: encoding and search. In the encoding phase, the input binary matrix is transformed into an integer-encoded matrix by dividing each row into bit words and translating them into their integer representation, which reduces the column dimensionality to optimize the subsequent search phase. The search phase iterates over pairs of seed rows, applying the Boolean AND operation to generate a bit pattern. If this pattern is new and meets the minimum number of columns threshold (mnc), an initial potential bicluster is created. Then, the remaining rows are explored and added to the bicluster if their AND operation with the pattern matches the original pattern. Finally, if the resulting bicluster reaches the minimum number of rows (mnr), it is considered a valid bicluster. BiBit stands out for its speed due to bit-level operations and its robustness with respect to data density and size.

- **Bimax** (Binary Inclusion-Maximal Biclustering Algorithm) [124]: Self-described as a simple and fast biclustering method, it is based on a binary data model that assumes two levels of gene expression: no change (0) or change (1). In this model, a bicluster is defined as a submatrix in which all elements are 1, implying that a group of genes exhibits a compatible response across a subset of conditions. The main goal of Bimax is to identify "inclusion-maximal" biclusters, meaning those that are not fully contained within any other bicluster. To achieve this, Bimax employs a divide-and-conquer strategy, recursively partitioning the matrix and excluding regions containing zeros to optimize the search. The recursion ends when a submatrix consisting solely of ones is found, which represents a bicluster. Unlike other algorithms, Bimax is capable of finding all optimal biclusters within its binary model. Despite the biological simplification of its data model, Bimax has proven to produce biologically relevant results that are comparable to those of more complex methods.

- **CCA** (Cheng and Church's Algorithm) [101]: Considered the first formal biclustering approach applied to gene expression data analysis, CCA introduced the concept of a bicluster as a submatrix in which genes and conditions exhibit coherent behavior. Its main metric is the Mean Squared Residue (MSR), which measures the deviation of each value from the means of its row, its column, and the entire bicluster. The algorithm searches for biclusters with an MSR below a threshold δ, using an iterative procedure based on the removal and addition of nodes. Initially, the process starts from the full matrix, and rows or columns that increase the MSR are removed until a coherent bicluster is obtained. This is followed by an expansion phase in which rows or columns are added if their inclusion

does not raise the MSR. To discover multiple biclusters, the elements already identified are masked with random noise. Although its approach is greedy and does not guarantee global optima, CCA laid the foundation for modern biclustering and remains a key reference in exploratory analysis of transcriptomic data.

- **ISA** (Iterative Signature Algorithm) [13]: This is a biclustering technique designed to identify transcription modules (TMs), understood as subsets of co-regulated genes (Gm) and the experimental conditions (Cm) that induce such co-regulation. The algorithm starts from an initial set of genes or conditions and iteratively refines its composition until reaching a fixed point that defines a TM. At each iteration, linear transformations are applied, followed by threshold functions (f_{tC} and f_{tG}) that filter genes and conditions based on their relevance. To do this, ISA employs two normalized expression matrices derived from the original matrix E: EG and EC. The matrix EG is obtained by normalizing each row of E (the expression profile of a condition across genes) to have zero mean and unit norm, allowing consistent comparison of conditions. Similarly, EC is constructed by normalizing each column of E (the expression profile of a gene across conditions), also with zero mean and unit length, to facilitate comparison among genes. These normalized matrices are not identical, and the algorithm alternates between them to avoid biases during the iterative refinement. This process enables the identification of TMs in which genes show similar expression under specific conditions, and vice versa. By varying the thresholds, ISA can discover modular structures at different resolutions and detect overlapping modules, which is not possible with many traditional biclustering methods.

- **LAS** (Large Average Submatrices) [125]: Statistically motivated biclustering procedure designed to identify large submatrices with significant average values within a real-valued data matrix. The algorithm operates in an iterative and residual manner, using a significance score based on the Bonferroni correction [152], which takes into account both the size of the submatrix and its average value. This score is derived from a Gaussian null model. The LAS search procedure iteratively updates the row and column sets of a candidate submatrix in a greedy manner until reaching a local maximum of the scoring function. The goal is to identify submatrices that deviate significantly from the null model, and the scoring function serves as a metric for comparing and ranking submatrices of varying sizes and intensities.

- **OPSM** (Order-Preserving Submatrix) [120]: It searches for local patterns

in gene expression matrices, specifically submatrices defined by a subset of genes (G) and a subset of experiments (T) in which the expression levels of all genes in G induce the same linear ordering of the experiments in T. This type of pattern can reveal co-regulated genes across sequential stages of a biological process. Since finding OPSMs is an NP-hard problem, the proposed algorithm is based on a probabilistic model in which an OPSM is assumed to be hidden within a random matrix. The goal is to recover this hidden submatrix through the iterative evaluation of partial models that specify the smallest and largest elements in the ordering, extending them until a complete model is obtained and evaluating its statistical significance based on the number of supporting genes. The algorithm can discover multiple overlapping OPSMs and adapt to ordering conditions that are similar but not identical, offering a tool to explore local patterns that may be robust to noise in expression data.

- **Plaid** [113]: It addresses gene expression data analysis by modeling expression levels as a combination of additive layers. Initially, a background layer is established to represent general trends in the data. Additional layers are then added, each representing a bicluster, that is, a subset of genes and samples exhibiting distinctive expression patterns not explained by the background layer. The expression level of a gene in a sample is considered the sum of effects, including a mean effect, gene- and sample-specific effects, and a particular effect from the bicluster to which they belong, determined by binary membership parameters. This approach allows for the identification of potentially overlapping biclusters by sequentially adding layers to capture the underlying structure in the expression data.

- **Spectral** [118]: It aims to simultaneously cluster genes and conditions in gene expression data to identify distinctive “checkerboard” patterns, where certain genes are significantly over- or underexpressed in specific subsets of conditions. The method is based on the idea that such structures are reflected in the eigenvectors of the expression matrix, which can be obtained through Singular Value Decomposition (SVD). To uncover these hidden patterns, the algorithm incorporates normalization steps for the data matrix, either independently or jointly across genes and conditions, in order to emphasize existing biclusters by removing effects such as differences in experimental conditions or baseline gene expression levels. By analyzing the structure of the resulting eigenvectors and their resemblance to step-like vectors, the algorithm can identify subsets of genes with similar expression profiles across subsets of similar conditions, thereby discovering the biclusters.

- **xMOTIFs** (Conserved Gene Expression Motifs) [12]: It seeks subsets of genes that exhibit conserved expression levels (within the same state or value range) across a subset of samples. The algorithm employs a probabilistic approach that randomly selects samples as seeds and discriminative sets of samples to identify genes with conserved expression states between the seed and the discriminative set. The goal is to find the largest biclusters, that is, those with the highest number of conserved genes, and this process is repeated iteratively to cover all classes present in the data. Identifying these groups can be useful for distinguishing between different sample classes and for discovering potential therapeutic targets.

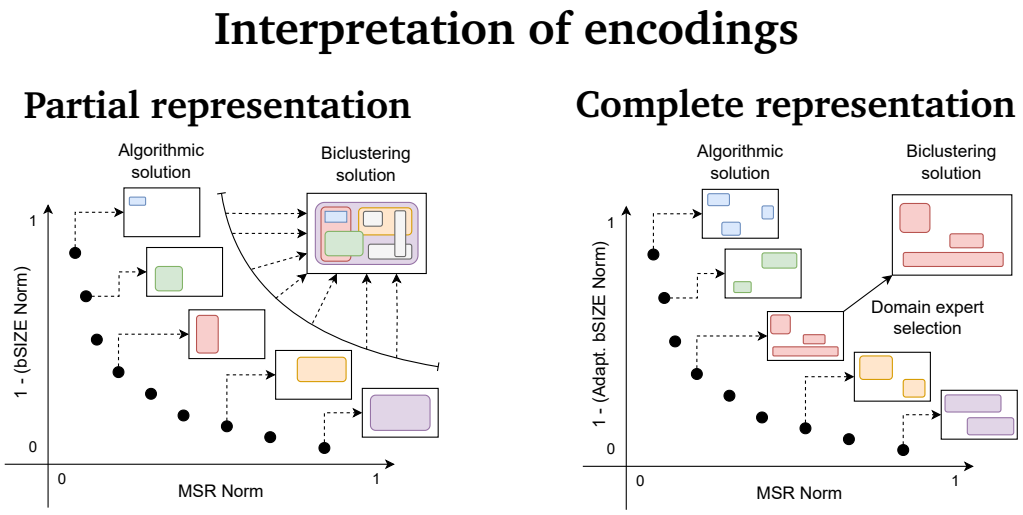
The methodological diversity among these approaches is considerable, ranging from greedy algorithms such as CCA, OPSM, and ISA, to divide-and-conquer strategies like Bimax, exhaustive enumeration techniques like BiBit, and statistically driven models such as Plaid and Spectral. However, none of these methods explicitly addresses the self-determination of the number of biclusters, nor do they incorporate contextual information specific to the domain of gene expression. Instead, all of them base their objectives on mathematical coherence functions, without leveraging the biological knowledge implicit in the data.

3.2.2 Methods based on evolutionary algorithms

Despite the methodological diversity previously mentioned for addressing the biclustering problem, metaheuristics have gained increasing popularity due to the NP-complete nature of the problem [153] and the need to simultaneously satisfy multiple objectives [19, 154]. In particular, they have proven especially applicable in the biomedical field for the analysis of gene co-expression [155, 156], where the high exploratory capacity of the solution space and the ability to optimize multiple criteria are key strengths of this approach, leading to positive results in this context.

Most metaheuristic-based proposals have employed encodings in which each individual represents a single bicluster, specifying the rows and columns it comprises [19]. From the perspective of the biomedical domain and the incorporation of domain knowledge, this partial encoding of individuals presents two main drawbacks:

- **Partial representation of the solution**: In current approaches, each individual represents only a fraction of a possible complete solution to the problem, rather than a full solution. Traditionally, the global solution has been defined as the approximate Pareto front generated by the algorithm (see Figure 3.1a for a better illustration). This formulation leads to two



(a) For the partial representation an individual of the algorithm represents a single bicluster. This means that the solution of the real biclustering problem is obtained from the union of the algorithmic solutions of the front, leading to redundancies, quality heterogeneity, loss of overlap control and exclusion of learning of the number of biclusters.

(b) In the complete representation, each individual in the population represents a possible real solution to the biclustering problem. Therefore, in this case, there is a direct equivalence between the algorithmic and the real solution, which allows a better injection of contextual knowledge and overcomes the limitations imposed by traditional encoding.

Figure 3.1: Interpretation of each coding of individuals in the approximate Pareto front obtained by biclustering evolutionary algorithms.

additional drawbacks: on the one hand, it requires subjective postprocessing to interpret the algorithm's output, and on the other hand, the final solution may include biclusters highly specialized in certain objectives that are part of the front while neglecting others. As a result, it is possible to obtain very large biclusters with poor internal coherence, or highly coherent biclusters that are excessively small [157].

- **Lack of global perspective**: The fitness evaluation of an individual is based solely on the properties of the bicluster it represents, without considering key elements such as the global distribution of all biclusters, their degree of differentiation, or the overall coverage of the data matrix. This limitation is particularly restrictive in domain-specific problems, as it prevents the design of deeper and more contextualized objectives that capture biological domain-specific aspects of the data.

This motivated the development of new encodings that enable a more com-

prehensive view of the problem, in which each individual represents a complete set of biclusters. While these approaches aim to overcome the previously mentioned limitations, their effectiveness has been only partial.

Figure 3.1 illustrates how the interpretation of the Pareto front varies depending on the encoding used. On the left (Figure 3.1a), the case of partial encoding is shown: each individual in the front contains a single bicluster, and the final solution is obtained by combining multiple individuals. On the right (Figure 3.1b), the truly complete encoding is presented, where each individual already represents a full solution to the problem.

Partial codification

To represent a single bicluster in a partial manner, the research community has mainly relied on two types of encoding. On the one hand, works such as [158, 155, 159] employ a binary representation, where the vector is divided into two parts: the first indicates, using binary values, the rows that belong to the bicluster, and the second does the same for the columns. On the other hand, studies such as [160, 156, 161] use variable-length integer encodings, where the first elements of the vector correspond to the indices of the rows, followed by the indices of the columns that form the bicluster.

Regardless of this distinction, the remainder of the implementation in these algorithmic proposals is quite similar. In fact, as pointed out in [19], the analysis of a wide range of methods has allowed this part of the state of the art to be reduced to a well-defined set of fitness functions and genetic operators.

Regarding genetic operators, the most common crossover strategies include one-point crossover, two-point crossover, and uniform crossover. However, more specialized operators have also been proposed, such as the bicluster crossover [162], which merges two parent biclusters and applies a discretization step to generate coherent offspring. As for mutation, strategies range from random modifications to more informed techniques inspired by the Cheng and Church algorithm [101], such as removing nodes with high variance [163], as well as correlation-based mutations that aim to improve the internal coherence of the biclusters [162].

Regarding objective functions, classical metrics are commonly used, such as bicluster size (bSIZE), variance (VAR) [100], row variance (rVAR) [164], and mean squared residue (MSR) [101]. In addition, more advanced metrics have been proposed to detect complex patterns: SMSR [165] for scaling patterns, ACF [166] and ACV [167] for assessing internal correlation, VE [168] for proportional patterns, and CVF [169] for measuring relative dispersion.

Some approaches combine multiple metrics into a single aggregated objective function in order to balance bicluster size, coherence, and diversity. However, the vast majority adopt a multi-objective approach, where different quality criteria are optimized in parallel to better explore trade-offs between conflicting objectives [19].

Nevertheless, it should be recalled that the partial encoding implemented in these proposals entails the previously discussed limitations, as each individual represents only a single bicluster. This restricts the design of global objectives and hinders the integration of contextual knowledge about the complete solution. For this reason, new encoding strategies have emerged that attempt to overcome these limitations by allowing each individual to directly represent a complete solution to the problem.

Complete codification

In the literature, three main proposals stand out for using alternative encodings that depart from the traditional approach by allowing multiple biclusters to be represented within a single individual:

- **BI-MOCK** [170] adopts a representation in which interactions between genes are encoded through both the value and the position of each element, giving rise to gene groupings. This design leads to a variable number of biclusters that emerges during the algorithm's convergence process. Nevertheless, all biclusters within an individual share the same set of columns, which breaks the direct correspondence between an individual and a complete solution to the problem. Consequently, it is not possible to incorporate either global or domain-specific objective functions. Only two classical objectives, applicable to individual biclusters, are used: Bicluster Size (bSIZE) and Mean Squared Residue (MSR). The fitness of each individual is computed as the average of its biclusters' scores, which means that poorly performing biclusters are not penalized as long as other biclusters achieve high scores that compensate for the overall average.
- **PBD-SPEA2** [171] also employs an encoding that supports the inclusion of multiple biclusters within a single individual. However, the number of biclusters must be predefined, making it an external parameter rather than an emergent property of the learning process. Additionally, the final solution is constructed through a stochastic post-processing phase in which biclusters from different individuals are merged. As a result, the actual solution does not correspond to any single individual in the population.
- **BiClustSMEA** [172] is a hybrid approach that combines evolutionary com-

Table 3.2: Comparison between the proposals with non-traditional encodings.

Issue	Aspect	BI-MOCK	PBD-SPEA2	BiClustSMEA
Encoding	Multiple biclusters	Yes	Yes	Yes
	Variable number of biclusters	Yes	No	Yes
	Self-learning of quantity	Yes	No	No
	Real equivalence	No	No	Yes
Objectives	Individual objectives	Yes	Yes	Yes
	Global objectives	No	No	No
	Domain-specific objectives	No	No	No
	Penalising heterogeneity between biclusters	No	No	No
Applicability	Free search space design	No	No	No
	Self-configuration	No	No	No
	Open source	No	No	No

putation with self-organizing maps. Its intricate encoding, which relies on gene and condition centroids, enables each individual to represent multiple biclusters. Although the number of biclusters is not explicitly defined by the user, it is determined during execution through a random variable. This number influences the structure of the representation but is not integrated into the learning process of the evolutionary algorithm. Additionally, as in BI-MOCK, the evaluation relies on individual objective functions whose values are aggregated using the arithmetic mean.

As summarized in Table 3.2, none of the proposals using non-traditional encodings succeed in satisfying all the desirable features for a complete and effective representation of the biclustering problem in biomedical contexts. While all three approaches allow encoding multiple biclusters per individual, only BI-MOCK and BiClustSMEA support a variable number of biclusters, and only BI-MOCK incorporates this number into the algorithm's learning process. However, BI-MOCK loses the true equivalence between an individual and a full solution by forcing all biclusters to share the same columns, which limits the model's flexibility.

Regarding optimization objectives, all the proposals rely exclusively on individual functions applied to each bicluster, aggregating their results through arithmetic means. This prevents the incorporation of global functions that evaluate the solution as a whole, as well as the implementation of domain-specific objectives such as regulatory coherence or biological differentiation between biclusters. In addition, none of the proposals penalizes heterogeneity among biclusters within the same individual, which can lead to unbalanced solutions.

Finally, in terms of applicability, all the analyzed approaches share three key

limitations: the search space cannot be freely designed, there is no built-in self-configuration mechanism, and no open-source implementations have been released, which hinders reproducibility and fair comparison between methods.

Chapter 4

Benchmark datasets

This chapter presents the dataset used for the experimental evaluation of the algorithms developed in this thesis. Since the work addresses two distinct tasks, on the one hand the inference of gene regulatory networks, and on the other the detection of co-expression patterns through biclustering, different benchmarks specifically designed for each case have been considered. For network inference, both simulated data (based on real-world networks or generated from scratch) and expression profiles obtained directly from patient samples have been used. In the case of biclustering, the benchmark includes synthetic numerical matrices generated using controlled simulators, as well as real-world gene expression datasets without a reference solution. This strategy allows for analyzing the behavior of the algorithms in controlled, semi-realistic, and real-world contexts, thus covering a wide spectrum of experimental scenarios. The following sections provide a detailed description of the datasets used for each of these tasks.

4.1 Gene Regulatory Networks Inference

In order to provide a solid and diverse experimental framework for the evaluation of network inference methods, a comprehensive dataset has been compiled, encompassing both simulated scenarios and real-world gene expression data. This benchmark includes networks from well-established challenges in the literature, data generated from real-world networks through simulation, synthetic networks built from scratch based on known topological properties, and expression profiles obtained directly from patient samples. This variety of sources allows for analyzing the behavior of the algorithms in controlled, semi-realistic, and real-world contexts, thus supporting the formulation of robust and generalizable conclusions. The following sections detail the characteristics and origin of each dataset used.

4.1.1 Simulated expression data

Synthetic gene expression datasets have been gathered from multiple sources to ensure coverage across different areas of specialization. On one side, the selection includes well-established benchmark networks frequently used in the literature to evaluate and compare the performance of various methods. Notable examples are the DREAM challenges [142] (particularly editions 3, 4, and 5) and the IRMA network in yeast [173]. On the other side, several databases compile gene regulatory networks by integrating experimental findings, literature-based evidence, and inference results from diverse algorithms. Among these are TFLink [174], RegulonDB [175], RegNetwork [176], BioGrid [177], and GRNdb [178]. However, none of these resources provide complete experimental expression data aligned with the full extent of their regulatory networks. Therefore, a gene expression data simulator was incorporated to generate coherent and realistic datasets suitable for network inference tasks.

For this purpose, the simulator SysGenSIM [179] has been employed. SysGenSIM is a software tool designed to simulate systems genetics experiments in model organisms. It allows users to input a reference network structure in the form of an edge list and generate expression data based on a nonlinear dynamic model. The simulator supports several types of perturbations, including: *knock-out* (where the transcription rate of deleted genes is set to zero), *knock-down* (where the transcription rate is scaled down to a value below one), *over-expression* (where the transcription rate is increased to a value above one), and *mixed perturbations* (which combine the previous types).

In this Thesis, the aforementioned databases were used to extract gene reg-

Table 4.1: Overview of the academic benchmark assembled for the experimental framework of this Thesis. It includes over one hundred problem instances designed to maximize diversity and support robust conclusions. Each regulatory network has been subjected to all applicable types of perturbations, with each resulting dataset constituting a distinct instance. Legend: KO (Knock-Out), KD (Knock-Down), OE (Over-Expression).

Source	Networks	Sizes	Simulator	Disturbance	Instances
DREAM3 [183]	15	10, 50 and 100	DREAM team	-	15
DREAM4 [142]	10	10 and 100	DREAM team	-	10
DREAM5 [142]	3	1643, 4511 and 5950	DREAM team	-	3
IRMA [173]	1	5	Cell culture: RT-PCR	Switch on/off	2
TFLink [174]	4	12, 75, 163 and 371	SysGenSIM	Mixed	4
RegulonDB [175]	1	2234	SysGenSIM	Mixed	1
RegNetwork [176]	2	983 and 1033	SysGenSIM	Mixed	2
BioGRID [177]	32	6 - 1505	SysGenSIM	Mixed	32
GRNdb [178]	11	320 - 1598	SysGenSIM	Mixed	11
From scratch	4	20, 50, 100 and 200	SysGenSIM (EIPO Modular)	KO, KD and OE	12
From scratch	4	20, 50, 100 and 200	SysGenSIM (Scale Free)	KO, KD and OE	12
GRNdata [184]	2	300 and 1000	SynTReN	-	2
GRNdata [184]	1	1000	Rogers	-	1
GRNdata [184]	2	1565 and 2000	GeneNetWeaver	-	2

ulatory networks, and expression profiles were generated by simulating mixed perturbations on the regulatory systems. Beyond using benchmark networks and biologically curated databases, SysGenSIM was also employed to construct synthetic gene networks from scratch. These artificial networks follow standard topological models characteristic of gene regulation, including scale-free and modular structures. Specifically, networks consisting of 20, 50, 100, and 200 genes were generated and subjected to knock-out, knock-down, and over-expression perturbations.

To further enhance the diversity of the datasets employed during experimentation, additional gene regulatory networks have been sourced from widely used simulators such as SynTReN [180], Rogers [181], and GeneNetWeaver [182]. These simulators contribute complementary network structures and expression profiles that enrich the variety of test scenarios.

Altogether, these resources form a comprehensive academic benchmark comprising over one hundred distinct problem instances. The origin of each dataset is outlined in Table 4.1 and further elaborated in the following subsections. While this benchmark has been used to support the core experimentation, it has not always been employed in its entirety. In preliminary evaluations or in specific branches of the research, reduced subsets of the benchmark have occasionally been used to facilitate testing and analysis.

DREAM

The DREAM challenges [142] consist of a collection of scientific competitions where teams of researchers collaborate to create novel methods and algorithms aimed at tackling specific issues in biology and medicine. These competitions promote innovation through a collaborative yet competitive environment, and frequently adopt an open-source philosophy to make both results and developed tools publicly available.

Over the years, these challenges have focused on a wide range of bioinformatics problems, including predicting drug responses, modeling protein structures, identifying genetic markers linked to diseases, among others. By gathering experts from across the globe, the DREAM challenges have played a key role in advancing the field of bioinformatics and enhancing our understanding of molecular biology.

- **DREAM3**: The DREAM3 "in silico" network inference challenge [183] was designed to evaluate the ability of computational approaches to infer gene regulatory networks of different sizes and connectivity levels. The datasets used in this challenge were based on subnetworks from two organisms: *Escherichia coli* (E. coli) and *Saccharomyces cerevisiae* (yeast). These datasets were synthetically generated using continuous differential equations that approximated the regulatory dynamics underlying gene expression, with a small amount of Gaussian noise added to simulate measurement errors. Each subchallenge was categorized by network size, featuring five networks (Ecoli1, Ecoli2, Yeast1, Yeast2, Yeast3) in each group. The challenge included networks with 10, 50, and 100 nodes, offering 4, 23, and 46 time series trajectories, respectively. DREAM3 has since become a widely used benchmark for evaluating gene regulatory network reconstruction methods from gene expression data. The scripts provided with the challenge introduced a standardized evaluation framework, which has helped facilitate fair comparisons among different approaches proposed in the literature.

- **DREAM4**: The DREAM4 challenges introduced several improvements over the previous DREAM3 edition, notably through the inclusion of new datasets. As in the earlier version, the in silico network inference task was divided into two categories based on network size: one with 10-node networks and another with 100-node networks, each consisting of 5 networks, similar to DREAM3. For the 10-node networks, participants received gene expression data covering 21 time points with 5 replicates, while the 100-node networks offered the same number of time points but included 10 replicates. These networks featured diverse topologies designed to resem-

ble real biological systems such as *Escherichia coli* and *Saccharomyces cerevisiae*, capturing their dynamic behavior through varying initial conditions and kinetic parameters. Expression data were generated using stochastic differential equations, with added noise proportional to gene expression levels, mimicking real microarray datasets. Each network was associated with four types of experimental data: time series, wild type, knock-out, and knock-down. This edition is considered one of the most thoroughly explored in the DREAM series, partly due to the greater availability of evaluation scripts and background materials, which encouraged broad participation and testing of diverse inference approaches.

- **DREAM5**: In this edition of the DREAM challenges, the scientific community was invited to infer genome-scale transcriptional regulatory networks using gene expression data from four sources: an *in silico* benchmark (Net 1), the human pathogen *S. aureus* (Net 2), the prokaryotic model organism *E. coli* (Net 3), and the eukaryotic model organism *S. cerevisiae* (Net 4). These networks included 1643, 2810, 4511, and 5950 genes, respectively, offering a more biologically realistic setting. However, over the course of the challenge, the network corresponding to *S. aureus* (Net 2) was excluded from the evaluation process. Since the complete regulatory interactions for these organisms were not fully known, the gold standards were inherently incomplete. Although this limitation is common in many benchmark datasets, the *S. aureus* network was particularly affected due to the limited number of experimentally supported interactions, which significantly compromised the reliability of the evaluation. As a result, and in line with many studies in the literature, only networks 1, 3, and 4 are typically used for analysis.

IRMA

The In vivo Reverse-engineering and Modeling Assessment (IRMA) network [173] was developed to evaluate the performance of different methods for reconstructing gene regulatory networks. To this end, quantitative RT-PCR was used to monitor the expression levels of the yeast *Saccharomyces cerevisiae* at multiple time points. The network consists of 5 genes (CBF1, GAL4, SWI5, GAL80, and ASH1) and includes 6 regulatory interactions. These interactions allow the network to operate in two modes, “switch on” and “switch off”, depending on whether the cells are cultured in galactose or glucose, respectively.

This synthetic network was designed to mimic the regulatory behavior of larger eukaryotic systems, while maintaining a manageable scale. It was carefully constructed to minimize interference from endogenous genes and to re-

spond specifically to galactose, which acts as an inducer of gene transcription. Despite its limited size, the IRMA network exhibits complex connectivity, including regulatory cascades, single-input motifs, and several feedback loops involving both transcriptional activators and repressors.

Simulated datasets based on real-world networks

To increase the diversity of networks analyzed in this Thesis, gene regulatory networks that have been experimentally validated in the literature were retrieved from biological databases. Expression data were then simulated based on these networks to assess how accurately various inference methods, including those proposed in this Thesis, are able to reconstruct them. The resulting datasets are as follows:

- **TFLink**: TFLink [174] offers detailed information about transcription factors, including their target gene interactions, associated nucleotide sequences, and the genomic coordinates of their binding sites. The database compiles data for humans and six model organisms: mouse (*Mus musculus*), rat (*Rattus norvegicus*), zebrafish (*Danio rerio*), fruit fly (*Drosophila melanogaster*), nematode (*Caenorhabditis elegans*), and yeast (*Saccharomyces cerevisiae*). Each entry in TFLink is accompanied by source annotations, which may include references to databases, experimental techniques, or scientific publications. The construction of TFLink involved evaluating several databases, ultimately selecting ten for integration: DoRothEA [185], GTRD [186], HTRIdb [187], JASPAR [188], ORegAnno [189], REDfly [190], ReMap [191], TRED [192], TRRUST [193], and Yeastract [194]. In this Thesis, the networks labeled as "small-scale" by TFLink for the organisms *Caenorhabditis elegans*, *Drosophila melanogaster*, *Rattus norvegicus*, and *Saccharomyces cerevisiae* were selected. After applying a filtering criterion based on the number of supporting detection methods, the resulting networks contain 75, 163, 12, and 371 genes, respectively.
- **RegulonDB**: RegulonDB [175] is a comprehensive and regularly updated database dedicated to the study of gene regulation in *Escherichia coli K-12*. It provides extensive information on regulatory elements, transcription factors, gene interactions, and experimental conditions. The database integrates data manually curated from scientific literature, high-throughput experiments, and computational predictions. For this Thesis, the file labeled "TF-gene interactions" from the dataset containing experimentally supported evidence was downloaded. A filtering step was then applied to exclude all interactions marked with "weak" confidence or ambiguous regulatory signs. As a result, a single network comprising 2,234 genes was

obtained.

- **RegNetwork**: RegNetwork [176] is an extensive database that compiles transcriptional and post-transcriptional regulatory interactions for both humans and mice. It encompasses five categories of interactions: TF-TF, TF-gene, TF-miRNA, miRNA-TF, and miRNA-gene. This resource integrates manually curated data from multiple databases, as well as inferred regulatory relationships based on transcription factor binding sites (TFBSs). Furthermore, conserved TFBS information is used to predict potential regulatory links between regulators and their targets, increasing the comprehensiveness of the dataset. In this Thesis, the human and mouse networks were downloaded without applying any filtering or modifications. Consequently, the aim is to infer regulatory networks consisting of 983 genes for the human dataset and 1,033 genes for the mouse dataset.

- **BioGRID**: BioGRID [177], formally known as the Biological General Repository for Interaction Datasets, is an open-access resource dedicated to the collection and dissemination of genetic and protein interaction data from both model organisms and humans. It contains an extensive set of more than 1,740,000 curated interactions, derived from high-throughput experiments and individual studies, all compiled from over 70,000 publications in the scientific literature. The database offers in-depth coverage for certain key organisms, including budding yeast (*S. cerevisiae*), fission yeast (*S. pombe*), and thale cress (*A. thaliana*), with ongoing efforts aimed at expanding to various *metazoan* species. These curation efforts are primarily focused on uncovering biologically relevant pathways and conserved networks with implications for human health. In this Thesis, 32 gene networks from BioGRID have been incorporated, corresponding to the following organisms: *Human papillomavirus 5*, *Human papillomavirus 6b*, *Bacillus subtilis 168*, *Bos taurus*, *Macaca mulatta*, *Middle-East Respiratory Syndrome-related Coronavirus*, *Canis familiaris*, *Chlamydomonas reinhardtii*, *Chlorocebus sabaeus*, *Neurospora crassa OR74A*, *Cricetulus griseus*, *Danio rerio*, *Oryctolagus cuniculus*, *Oryza sativa Japonica*, *Emericella nidulans FGSC A4*, *Plasmodium falciparum 3D7*, *Gallus gallus*, *Glycine max*, *Simian Immunodeficiency Virus*, *Human Herpesvirus 1*, *Simian Virus 40*, *Human Herpesvirus 4*, *Human Herpesvirus 5*, *Streptococcus pneumoniae ATCCBAA255*, *Strongylocentrotus purpuratus*, *Sus scrofa*, *Human Herpesvirus 8*, *Vaccinia Virus*, *Human Immunodeficiency Virus 2*, *Xenopus laevis*, *Human papillomavirus 16*, and *Zea mays*. These networks range in size from 6 to 1,505 genes, contributing substantial diversity to the experimental dataset used in this Thesis.

- **GRNdb**: GRNdb [178] is an accessible and user-oriented database designed to facilitate the exploration and visualization of predicted regulatory networks. These networks are composed of interactions between transcription factors (TFs) and their downstream target genes, collectively referred to as regulons. The regulatory relationships are inferred from large-scale RNA-seq datasets and previously known TF-target interactions across multiple biological conditions in humans and mice. It should be emphasized that all the regulatory information in GRNdb is obtained through omics-based computational analyses, without direct experimental validation. The platform enables efficient searching, browsing, and retrieval of TF-target pairs and associated motifs, supporting both single-cell and bulk expression data. Moreover, it offers functionalities for examining gene expression profiles and exploring associations between gene expression and patient survival across numerous TCGA cancer types. In this Thesis, the following 11 networks have been considered: *Fetal-Brain*, *Fetal-Thymus*, *Adult-Pancreas*, *Adult-Muscle*, *Adult-Adipose*, *Adult-Ascending-Colon*, *Adult-Lung*, *Adult-Liver*, *Fetal-Calvaria*, *Adult-Epityphlon*, and *Adult-Rectum*. These networks contain up to 1,598 genes.

Simulated data from scratch

In addition to the expression data simulated from experimentally derived gene regulatory networks, additional simulations were performed from scratch to replicate degree distributions commonly reported in the literature.

- **EIPO Modular distribution**: The Exponential In-degree and Power law Out-degree (EIPO) Modular distribution [195] is a synthetic network model designed to emulate key structural characteristics of real-world gene regulatory networks. It features an exponential distribution for in-degrees and a power-law distribution for out-degrees. To introduce modularity, the model generates multiple densely connected gene clusters, or "modules", which are then interconnected by rewiring edges according to a predefined probability. This structure captures both the degree distributions and modular organization commonly observed in biological systems, resulting in a more biologically plausible model of gene interactions. Based on this distribution, networks comprising 20, 50, 100, and 200 genes were generated, with simulated gene expression data produced under knock-out, knock-down, and over-expression perturbations for each network size. This yielded a total of 12 simulated instances.
- **Scale-Free distribution**: The scale-free distribution describes a network topology in which the node degree follows a power-law distribution. In this

context, the likelihood of a node having a given number of connections decreases according to a power law, as opposed to a normal or Gaussian distribution. This type of distribution has been widely reported in biological networks, which has led to its inclusion in the SysGenSIM simulator. The same network sizes and perturbation types as in the EIPO Modular distribution were used—namely, networks of 20, 50, 100, and 200 genes subjected to knock-out, knock-down, and over-expression perturbations—resulting in a total of 12 simulated datasets.

- **Other simulators**: To further enhance the diversity of the dataset used in this Thesis, additional networks were obtained from simulators other than SysGenSIM. The R package GRNdata [184] was employed for this purpose, and networks generated by several well-established simulators were included. Specifically, two networks with 300 and 1,000 nodes were collected from SynTReN [180], one network with 1,000 genes was obtained from Rogers [181], and two networks with 1,565 and 2,000 nodes were retrieved from GeneNetWeaver [182].

4.1.2 Real-world expression data

To evaluate the algorithm's performance under real-world conditions, two gene expression datasets representing relevant biomedical scenarios have been used. The first corresponds to samples from melanoma patients, where analyzing the immune environment is key to optimizing targeted therapies. The second contains transcriptomic profiles from patients with myalgic encephalomyelitis and fibromyalgia, two chronic diseases without validated biomarkers, in which the inference of regulatory networks may provide new hypotheses about their molecular mechanisms. In both cases, the data were preprocessed following standard reference protocols and used in studies aimed at identifying biomarkers and relevant regulatory relationships.

Melanoma

This gene expression dataset corresponds to real-world samples obtained from melanoma patients [196]. Gene expression levels were quantified using NanoString technology [197], with the Immune Profiling Panel specifically designed for immuno-oncology studies. This panel includes a total of 770 genes related to immune response, covering markers for 24 immune cell types, immune checkpoint inhibitors, cancer-testis antigens, as well as genes involved in both innate and adaptive immune responses.

The primary objective behind the generation of this dataset is to characterize

the immune environment associated with melanoma, in order to optimize therapeutic strategies based on immune system modulation. Additionally, the dataset aims to support the identification of genetic biomarkers that may improve patient stratification, clinical response prediction, and risk assessment for toxicity in immunotherapy treatments.

Expression data processing was carried out according to the recommendations of the NanoString platform, using officially supported libraries [198, 199]. This procedure included quality control steps, background noise correction using negative controls, and normalization of raw counts through scaling factors computed from housekeeping genes, with the goal of minimizing the impact of potential technical biases.

For the analyses conducted as part of this thesis, the already processed expression data available in [200] were used. In that study, which was also focused on gene network inference, a subset of 35 genes was selected based on the highest variability in expression levels across samples, that is, those with the lowest relative stability.

Myalgic Encephalomyelitis and Fibromyalgia

This dataset corresponds to gene expression profiles obtained from samples of 43 women, classified into four groups according to their clinical diagnosis:

- 8 patients diagnosed with **Myalgic Encephalomyelitis/Chronic Fatigue Syndrome** (ME/CFS), according to the 2003 Canadian consensus criteria [201] and the 2011 International Consensus Criteria [202].
- 10 patients diagnosed with **Fibromyalgia** (FM), assessed following the American College of Rheumatology (ACR) criteria from 1990 [203] and 2010 [204].
- 16 patients with a **combined diagnosis of ME/CFS and FM**, meeting the established criteria for both conditions.
- 9 **healthy controls**, matched by age and body mass index (BMI) with the patient groups, with no history of chronic pain or fatigue, and free from medication.

All diagnoses were made by the same clinical expert in FM and ME/CFS at the Hospital de Manises (Valencia, Spain), ensuring consistency in the clinical evaluation.

Gene expression levels were measured from peripheral blood mononuclear cells (PBMCs), extracted via venipuncture after a 12-hour overnight fast. Sam-

ples were collected in K2EDTA tubes (Becton Dickinson, Franklin Lakes, NJ, USA), processed within two hours using a density gradient, and subsequently cryopreserved. Total RNA was isolated using the RNeasy Mini Kit (Qiagen, MD, USA), and its quality was verified with the Agilent TapeStation 4200 (Agilent), ensuring an RNA Integrity Number (RIN) greater than 7 in all samples.

Transcriptomic analyses were performed using customized Affymetrix HERV-V3 microarrays [205], designed to detect the expression of 1,559 genes involved in relevant cellular pathways, including immunity, inflammation, cancer, central nervous system function, cell differentiation, telomere maintenance, chromatin organization, and gag-like genes.

Samples were anonymized and randomly distributed across groups to reduce potential batch effects. The resulting CEL files were processed using the R package `oligo` [206], applying the Robust Multi-array Average (RMA) algorithm for normalization, background correction, and summarization of intensities. Differential expression analysis was conducted with the `limma` package [207], considering as differentially expressed those probes with a Benjamini–Hochberg (BH) adjusted p-value (FDR) below 0.1 and an absolute log2 fold change greater than 1.

The frequent co-occurrence and still unclear etiology of these two chronic conditions (ME/CFS and FM), together with the lack of validated biomarkers, motivated the inclusion of this dataset to identify potential distinctive or shared molecular interactions.

4.2 Co-expression Biclustering

This section describes the datasets used to evaluate the performance of the developed biclustering algorithms. In all cases, the data are represented as numerical matrices, where the goal is to identify simultaneous subsets of rows and columns (biclusters) that exhibit coherent patterns.

Both artificially generated matrices using simulators and real-world gene expression data have been considered. In the simulated cases, the biclusters embedded during data generation are used as a reference to assess the quality of the solutions found. In contrast, for real-world datasets, no ground truth is available, so external validation based on domain knowledge is applied.

Table 4.2: Characteristics of the data generated by G-bic.

Variable	Size of planted biclusters	Noise on data matrix	Nº Biclusters	Overlapping in columns
NB (Number of Biclusters)	50 x 50	0.0%	{3, 5, 8, 10}	Unrestricted
SB (Size of Biclusters)	{(25,25), (50,50), (75,75), (100,100)}	0.0%	3	Unrestricted
NL (Noise Level)	50 x 50	{5, 10, 15, 20} %	10	Unrestricted
OL (Overlap Level)	50 x 50	0.0%	10	{10, 15, 20, 25} %

4.2.1 Simulated data

Simulated data allow for the creation of controlled environments in which to objectively assess the ability of algorithms to identify biclusters. For this purpose, specific tools are used that enable the configuration of multiple aspects of the data matrix, such as the number of rows and columns, the number and size of the biclusters, the level of noise, and the degree of overlap among them.

In this thesis, two different simulators have been considered. The first is a generic tool that generates numerical matrices without assuming any specific domain. The second is specifically designed for the biomedical context and simulates structures typically found in gene co-expression studies.

Generic simulated datasets (G-bic)

This dataset consists of artificial matrices generated using the G-bic tool [208], designed to evaluate biclustering algorithms in generic contexts. These matrices were specifically generated to contain constant biclusters, which represent a suitable baseline case for an objective assessment of algorithm performance.

Although the G-bic simulator allows for the configuration of multiple aspects of the generated matrices, this thesis focuses on systematically exploring four variables that are particularly relevant for evaluating algorithm behavior: number of biclusters (NB), bicluster size (SB), noise level (NL), and column overlap (OL). Each of these variables was tested at four different levels, generating three different matrices per level using different random seeds. In total, 48 numerical matrices of size 1000×500 were obtained, each accompanied by its corresponding reference set of biclusters (gold standard). The specific characteristics of each configuration are detailed in Table 4.2.

It is worth noting that, although G-bic allows the configuration of column overlap, it does not offer full control over row overlap. Therefore, an additional post-processing step was applied, in which rows shared by multiple biclusters were duplicated. Subsequently, the cells outside the biclusters were replaced with values from unclustered rows, which were then removed to preserve the original matrix size and avoid distortions.

Table 4.3: Characteristics of the matrices generated by the FABIA simulator.

Instance	Description	Matrix Size	Nº Biclusters	Bicluster Size	Overlap	Noise (Std. Dev.)
Inst. 1	Small Biclusters - Low Noise	200 x 100	10	20 x 10	0% x 32%	1.0
Inst. 2	Large Biclusters - Moderate Noise	500 x 200	8	65 x 25	0% x 35%	2.0
Inst. 3	Mixed Biclusters - High Noise	300 x 150	12	(20 - 30) x (7 - 12)	0% x 30%	4.0
Inst. 4	Medium Biclusters - Moderate Noise	400 x 150	10	40 x 15	0% x 37%	1.5

Biologically-inspired simulated datasets (FABIA)

This dataset consists of artificial matrices specifically designed for gene expression analysis. To generate them, the simulator included in the `FABIA` R package [209] was used, which is widely employed in biclustering studies within this domain.

Unlike the previous simulator, FABIA allows for the generation of biclusters that reflect more realistic co-expression patterns, including variability in expression intensity and the presence of additive noise. In this thesis, four different configurations were defined, selected with the aim of representing a variety of typical scenarios in the biomedical field, ranging from small and well-defined biclusters to mixed structures with higher levels of noise and column overlap. The specific characteristics of each generated instance are provided in Table 4.3.

4.2.2 Real-world expression data

In addition to the simulated data, real-world gene expression matrices have been used to validate the biological applicability of the solutions obtained. Specifically, the dataset compiled by [210] has been employed, which aggregates gene expression time series in *Saccharomyces cerevisiae* (yeast) from various previous studies [211, 212, 213].

These data were generated through cDNA microarray experiments and have undergone a thorough preprocessing process, including the removal of genes with excessive missing values and normalization using the Multiple-Slide Normalization procedure [214]. After this processing, a total of 17 datasets were obtained, each consisting of approximately 1000 genes selected based on their variability over time.

Unlike the simulated datasets, these data do not have a technical reference (gold standard) to directly compare the quality of the obtained biclusters. Instead, validation is performed through biological functional enrichment analysis of the grouped genes.

The characteristics of the 17 datasets used are summarized in Table 4.4, ex-

Table 4.4: Summary of the real-world gene expression data sets used in the experiments, adapted from [210].

Name	**Source**	**Time points**	**Original genes**	**Filtered genes**
alpha factor	[211]	18	6178	1099
cdc 15		24	6178	1086
cdc 28		17	6178	1044
elutriation		14	6178	935
1mM menadione	[212]	9	6152	1050
1M sorbitol		7	6152	1030
1.5mM diamide		8	6152	1038
2.5mM DTT		8	6152	991
constant 32nM H_2O_2		10	6152	976
diauxic shift		7	6152	1016
complete DTT		7	6152	962
heat shock 1		8	6152	988
heat shock 2		7	6152	999
nitrogen depletion		10	6152	1011
YPD 1		12	6152	1011
YPD 2		10	6152	1022
yeast sporulation	[213]	7	6118	1171

tracted and adapted from [210].

Part II
Methodology, analysis and results

Chapter 5

GENECI: Baseline consensus inference through mono-objective evolution

This chapter presents the design of an evolutionary machine learning algorithm called `GENECI`, which acts as an organizer for the construction of ensembles by combining the outputs of several prominent inference techniques described in section 3.1.1 (ARACNE [96], C3NET [127], BC3NET [126], CLR [94], GENIE3 [9], KBOOST [132], MRNET [138], MRNETB [139], PCIT [143] and TIGRESS [97]) and optimizing the consensus network derived from them, according to their confidence levels and topological characteristics.

5.1 Algorithmic Proposal

GENECI (GEne NEtwork Consensus Inference) takes up the idea of weight assignment seen in [149], but tries to improve the optimization process, choosing an approach similar to the one presented in [215] although adapted for GRNs. This paper is helpful to know in practice the concept of evolutionary machine learning [216] despite having a purpose outside network inference. This thesis aims to solve these problems through computational learning by using its results to optimize the quality of a solution derived from them, thanks to the application of an evolutionary algorithm.

It has been decided to implement a genetic algorithm within the evolutionary branch. Firstly, the flexibility of its operators allows for searching for implementations that adapt correctly to the problem. Secondly, as it is an algorithm widely

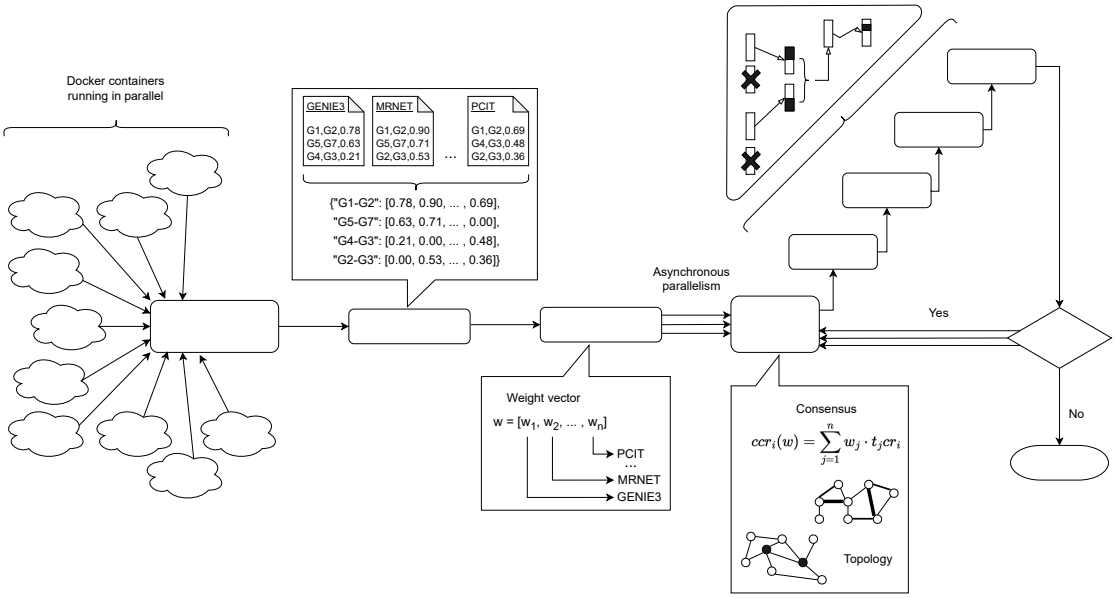


Figure 5.1: Architecture and workflow covered in GENECI. First, the execution of multiple individual inference techniques in parallel is enabled by encapsulating their implementations in Docker containers. After that, their results are normalized and collected by the evolutionary algorithm in order to optimize weight vectors that assign a value to each technique. The weight vectors are iteratively subjected to evaluation (depending on the quality and topology of the consensus networks they represent), selection, crossover, mutation and finally an additional repair step to keep the sum of values at unity.

used in the literature, it has numerous execution environments that facilitate the design of our algorithm and the establishment of a solid and efficient construction base.

The genetic algorithm has been implemented using the jMetal framework [217] and takes as input a set of files with confidence lists related to interactions between genes. These lists can come from complementary runs on the Python package built or from external runs produced on other techniques. Although both options exist, it is recommended to run these techniques on the GENECI environment as it guarantees uniformity in the output format (tables made up of Source, Target and Trust columns) as well as the standardization of their confidence levels between 0 and 1, which is crucial for consensus building.

Notably, all the functionalities incorporated in GENECI (including the execution of the evolutionary algorithm itself) have been encapsulated in Docker images. The main reason for choosing this design is that GENECI can enjoy the conveniences and advantages of the Python programming language (such as the

construction of the package in PyPI or the hierarchical design of commands using the Typer library) while tolerating the implementation of each functionality in the most suitable programming language for this purpose, i.e. the most efficient or the one with the most specific and suitable libraries.

In the GENECI optimization process, the presence of the gold standard is avoided, and the consensus networks are evaluated according to their confidence levels and topological characteristics. Specifically, the designed objective function examines measures such as the confidence levels of the different links, the adequacy of the weights assigned in the ensemble to produce these values, the number of hubs [1] present in the consensus network and their similarity to a scale-free distribution[2].

Figure 5.1 shows a schematic diagram where each stage addressed in GENECI is contemplated. As usual, the algorithm includes the main stages of evaluation, selection, crossover and mutation, to which an additional repair stage is added because of the chosen representation. In addition, it can be seen how the stopping criterion imposed in this case is determined by a maximum number of evaluations specified as the input parameter. In more detail, its implementation pseudocode is presented in Algorithm 1.

To speed up the execution of the algorithm, an asynchronous parallelization has been implemented to evaluate, cross and mutate several individuals simultaneously. This approach considerably reduces the execution time in those machines with the appropriate performance despite increasing the amount of RAM consumed during the algorithm's progress.

The following sections describe the aspects and stages of the designed evolutionary algorithm, concluding with a compilation of all the necessary input parameters.

5.1.1 Solution Representation

The individuals of the population are represented by vectors of weights, where each position refers to a certain reconstruction of the network (indirectly a machine learning technique) and its value means the power or weight that this list has to vote for the consensus. Therefore, given a list of techniques $t = \{t_1, t_2, ..., t_n\}$ where each of them contains a list of relations $r_t = \{tr_1, tr_2, ..., tr_m\}$

[1] nodes with a statistically significant number of links concerning the rest

[2] that which is characterized by having a small number of high-degree nodes, concerning the rest that usually has a low number of links. This distribution is the opposite of the random distribution, where all nodes have a fairly similar number of connections.

Algorithm 1 Main code of the generalized EA

Require: files: List of input csv files with trusted lists.
Ensure: consensusList: Final consensus list.

```
inferredNetworks = ReadAll(files)
population = GeneratePop(len(inferredNetworks))
while numEvaluations < max do
    fitness = Evaluation(population)
    numEvaluations += len(population)
    selectedPopulation = Selection(fitness)
    crossPopulation = Crossover(selectedPopulation)
    mutPopulation = Mutation(crossPopulation)
    repPopulation = Repair(mutPopulation)
    population = repPopulation
end while
fitness = Evaluation(population)
bestIndividual = GetBest(fitness)
consensusList = MakeConsensus(bestIndividual)
return consensusList
```

with their respective confidence values $cr_t = \{tcr_1, tcr_2, ..., tcr_m\}$, an individual with a vector of weights $w = \{w_1, w_2, ..., w_n\}$ implies that the confidence value of the relation r_i in the consensus list (ccr_i) is given by Equation (5.1):

$$ccr_i(w) = \sum_{j=1}^{n} w_j \cdot t_j cr_i \tag{5.1}$$

where $t_j cr_i$ is the confidence value of the relationship r_i in the list produced by the technique t_j. If the relationship r_i is not contemplated in that list, it will be understood that the technique has not detected that interaction, and therefore, its confidence level for t_j is 0. A more detailed example of this calculation can be seen in the column *Consensus confidence* of Table 5.1, where the construction of a 6-gene consensus network is simulated for a set of three techniques and a given individual.

The representation of individuals (solutions) as weight vectors implies the need to constantly keep the sum of their values at one. During the generation of offspring, crossover and mutation operators can alter this property. To solve this problem, it has been decided to add an additional operator in the optimization process to recover this feature of the solutions.

Table 5.1: Example of input to the evaluation process. This table presents a gene network of 4 interactions (column 1) that has been inferred using three different individual techniques (columns 2, 3 and 4), the results of which are intended to be consensual. In this case, an individual is evaluated, proposing the following vector of weights: (0.5, 0.3, 0.2). The first significant value to be calculated is the consensus confidence (column 6), which consists of a simple weighted sum where the weight of each technique is multiplied by the level of individual confidence reported by the technique for the interaction in question. Second, a vector (column 7) is constructed, storing in each position the mean between the weight of the technique and the distance normalized to the median of the confidence levels of all techniques (column 5). This approach allows the calculation of the second significant value, the distance. This value consists of the difference between the maximum and the minimum of the vector constructed above, and the fitness function will try to minimize it.

Interaction	Tec 1	Tec 2	Tec 3	Median	Ind [0.5, 0.3, 0.2]		
					Consensus confidence	Vector	Distance
G1 - G2	0.78	0.9	0.69	0.78	0.78 * 0.5 + 0.9 * 0.3 + 0.69 * 0.2 = 0.798	[(0+0.5)/2, (1+0.3)/2, (0.75+0.2)/2]	0.4
G5 - G6	0.63	0.71	-	0.63	0.63 * 0.5 + 0.71 * 0.3 + 0 * 0.2 = 0.528	[(0+0.5)/2, (0.13+0.3)/2, (1+0.2)/2]	0.39
G4 - G3	0.21	-	0.48	0.21	0.21 * 0.5 + 0 * 0.3 + 0.48 * 0.2 = 0.201	[(0+0.5)/2, (0.78+0.3)/2, (1+0.2)/2]	0.35
G2 - G3	-	0.53	0.36	0.36	0 * 0.5 + 0.53 * 0.3 + 0.36 * 0.2 = 0.231	[(1+0.5)/2, (0.47+0.3)/2, (0+0.2)/2]	0.65

Two different repairers were designed. The first one, named *Standardization-Repairer*, performs a simple standardization where the values are rescaled, so their sum is 1. The second repairer, named as *GreedyRepairer*, performs a greedy repair where after choosing a position of the vector at random, it keeps its values until the sum exceeds unity (in which case it sets the following values to 0) or reaches the last position and adds the number needed to sum to 1. Finally, it was shown that the greedy repairer obtains worse results as a consequence of distorting the weights of the individuals excessively, bringing the optimization process closer to a random search. The repairer in charge of rescaling the values favours the algorithm's performance by maintaining the proportions of the values.

5.1.2 Evaluation

At each iteration, all the individuals in the population are evaluated to know the quality of their proposals and to make the selection step possible. The process begins with the translation of the vector of weights to the evaluated concept, the consensus network derived from the voting system seen above. However, for evaluation purposes, more than the confidence level alone is needed as a reference of reliability since a relationship between genes may obtain a fairly high value but has originated from an inadequate distribution of weights. For example, distributions that give too much weight to a particular technique, to a very small subset of them, or to techniques whose confidence values are not

supported by any others.

For this reason, the evaluation process is responsible for calculating a vector per interaction and the confidence value of each relationship after consensus. The average distance between the median confidence of all the techniques (scaled between 0 and 1) and the weight assigned to the evaluated individual are stored for each technique. After that, the distance between that vector's maximum and minimum values is calculated and stored, together with the previously calculated confidence.

This distance will try to be minimized in the first term of the fitness function, so the aim is to establish a compensation system between the distance to the median and the weight assigned by the individual. In this way, as all the values of the calculated vector have a similar value (and therefore, the distance between the maximum and the minimum is minimized), the techniques whose proposal is different from the rest (greater distance to the median) will be penalized by being given a lower weight. On the contrary, the techniques with a proposal quite close to the rest (smaller distance to the median) will be compensated by assigning a high weight.

Therefore, as shown in Table 5.1, assigning a higher weight to the first technique is beneficial for the first three interactions. This is because the first technique is the closest to the median for these cases, which provides some reliability to its proposal and consequently rewards the individual for having assigned it a higher weight. However, for the last interaction (G2 - G3), it can be seen that the calculated distance has grown because the technique whose value is more reliable (closer to the median) for this case is the third one, to which the individual has assigned the lowest weight.

Objective Function

An aggregate objective function with two weighted terms has been designed: Quality and Topology.

The **Quality** term aims to encourage the emergence of solutions with high confidence levels that also come from consistent weight distributions that assign greater importance to those techniques whose values have high support concerning the rest. Its purpose is to establish a certain contrast between good and bad links so that the links finally reported are of high reliability. In this term, a quality value is assigned to each interaction considered in the problem. For this assignment, two significant values are considered, the consensus confidence level and another previously introduced value called "distance". The first considerable value (consensus confidence) is calculated through a weighted sum. The

weight assigned by the individual being evaluated and the individual confidence level reported by that technique for the interaction in question is multiplied for each technique.

On the other hand, the "distance" value, as explained with the example shown in Table 5.1, is the difference between the maximum and the minimum of a vector that stores for each technique the mean between its weight and the distance between its individual confidence value and the median of the set of techniques. Finally, the quality value associated with interaction will be the mean between its consensus confidence level and the unit subtracted by the distance. In other words, an interaction will have a good quality when it has a high confidence level and a small distance value. A small distance value means that the maximum and minimum of the calculated vector are close values and that, therefore, the techniques with a confidence value far from the median have been assigned a lower weight than the rest. Likewise, it means that the techniques with a confidence level close to the median (smaller distance) have been compensated with a higher weight. Finally, those interactions whose quality exceeds the mean are chosen, and an attempt is made to maximize their quality while approximating their quantity to 10% of the total number of interactions. This threshold is set up because minimizing the number of good links would result in a fuzzy network. The aim is to establish a clear contrast that allows us to report truly reliable interactions.

Its implementation can be found in Algorithm 2. For each link in the consensus list, the mean between its confidence value and the unit subtracted by the distance mentioned above is calculated and stored in the $distConf$ vector in the pseudocode (lines 1 to 3). The mean of the previous vector is then calculated (line 4), and those whose value manages to exceed it are set as good links. On the one hand, the sum of the values of all these good links is carried out. On the other hand, their quantity is stored (lines 5 to 10). Since the algorithm is oriented to minimization, the result of this term is lower when the quality of the consensus network is higher.

First, to optimize the number of these good links, q_1 is calculated (lines 11 and 12 in Algorithm 2). Specifically, this variable will reduce its value when the number of good links is closer to 10% of the total number of possible links in the network. Second, for these links to have the highest possible quality, the sum of their averages between confidence and unity minus distance should be maximized. For this purpose, in q_2 the mean of the $distConf$ of these good links is calculated and adapted to the minimization objective by subtracting its value from unity (line 13 in Algorithm 2).

Algorithm 2 First term of the fitness function: Quality

Require: c: consensus list with confidence and distance values.
Ensure: $quality$: value of the first term of the fitness function.
1: $distConf$ = []
2: **for** i **in** $len(c)$ **do**
3: $distConf[i] = (conf_i + (1 - dist_i))\ /2$
4: **end for**
5: $mean = \frac{1}{len(c)} \cdot \sum_{i=1}^{len(c)} distConf[i]$
6: $distConfSum = 0$
7: $cnt = 0$
8: **for** i **in** $len(c)$ **do**
9: **if** $distConf[i] > \ mean$ **then**
10: $distConfSum \mathrel{+}= distConf[i]$
11: $cnt \mathrel{+}= 1$
12: **end if**
13: **end for**
14: $numPosLinks = N^2_{genes}$
15: $q_1 = |cnt - 0.1 \cdot numPosLinks|\ /\ (0.9 \cdot numPosLinks)$
16: $q_2 = 1 - (distConfSum/cnt)$
17: $quality = 0.25 \cdot q_1 + 0.75 \cdot q_2$
18: **return** $quality$

The fact that in q_2 only the quality of good links is reported allows GENECI to establish the balance mentioned above between distance to the median and weight by focusing exclusively on the most relevant interactions in the network. This allows to eliminate possible noise and contradictions that lower intensity relationships could cause and also not constantly penalize more selective and strict inference techniques such as C3NET.

Finally, the result of the term *Quality* is the value of q_1 multiplied by 0.25 plus that of q_2 multiplied by 0.75 (line 14 in Algorithm 2). This means that this function gives more importance to good links' quality than quantity.

The second term, called **Topology**, is more oriented toward improving the structure of the consensus network. To this end, it intends to positively evaluate those proposals that present networks with a scale-free distribution (as real biological networks usually are). Mathematically, it tries to increase the degree (number of links) of those genes with a high potential to be considered hubs. At the same time, it is intended that the number of genes that meet this condition should be relatively low since this is usually observed in real genetic networks.

Algorithm 3 Second term of the fitness function: Topology

Require: b: binary network originated after cut-off.
Ensure: $topology$: value of the second term of the fitness function.

$degree$ = []
for i **in** N_{genes} **do**
    for j **in** N_{genes} **do**
        $degree[i] = n[i][j]$
    end for
end for
$mean = \frac{1}{N_{genes}} \cdot \sum_{i=1}^{N_{genes}} degree[i]$
$hubsDegreeSum$ = 0
$hubs$ = 0
for i **in** N_{genes} **do**
    if $degree[i] > \ mean$ **then**
        $hubsDegreeSum$ += $degree[i]$
        $hubs$ += 1
    end if
end for
$t_1 = |hubs$ - 0.1 $\cdot\ N_{genes}|$ / (0.9 $\cdot\ N_{genes}$)
if $hubs >$ 0 **then**
    $t_2 = 1 - (hubsDegreeSum/hubs)\ /\ (N_{genes} - 1)$
else
    t_2 = 1
end if
$topology = (t_1 + t_2)\ /2$
return $topology$

The goal is to promote the approximation of the network to a scale-free configuration and to move away from a random structure.

However, the input of this second term is not the same as the first one. In order to correctly study the network topology, it should be decided which links are finally labelled as definitive and which are not reliable enough to do so. This decision is made by applying a certain cut-off criterion. Three different criteria were designed for this issue:

- The first one called *MaxNumLinksBestConfCriteria* takes as input the number of links k to be obtained in the definitive network. After knowing that input, it simply sorts from highest to lowest all interactions based on their confidence value and returns the top k.

- In the second one called *MinConfidenceCriteria*, the minimum confidence required to report a link is entered as the input value. Therefore, all interactions are filtered and only those that have exceeded the threshold are returned.
- The third one identified as *MinConfDistCriteria* has a similar operation to the previous criterion, except that in this case the threshold refers to the average between the confidence and the unit subtracted by the distance of the links.

After putting the three criteria to the test, it was found that the one that considers both metrics is the most effective.

As shown in Algorithm 3, the code starts by calculating the average of the degrees of all the nodes in the network (lines 1 to 5). After that, it selects as hubs those genes with a degree higher than the average to quantify them and store the sum of their degrees (lines 6 to 11).

Similarly to the previous term, with t_1 an optimization of the number of hubs is carried out trying to approximate its value to 10% of the total number of genes (line 12 in Algorithm 3). This quantity is the one that has been considered appropriate to bring the node degree distribution closer to a scale-free distribution, where a small number of nodes concentrate most of the network connections. On the other hand, with t_2, the sum of the degrees of all the hubs is maximized. For this purpose, the average of these degrees is calculated. Then, it is normalized by dividing it by the maximum achievable degree and subtracted from the unit to adapt the variable to the minimization objective (lines 13 to 16 in Algorithm 3).

Finally, the value of the returned term is the average of these two metrics (line 17 in Algorithm 3). Therefore, the fitness value assigned to an individual is given by Equation (5.2):

$$Fitness(Ind) = w_Q \cdot Quality(Ind) + w_T \cdot Topology(Ind) \tag{5.2}$$

where w_Q and w_T are the weights assigned to the *Quality* and *Topology* terms respectively, which are given as input parameters.

5.1.3 Selection

Selection is carried out using the classical binary tournament in which individuals are randomly grouped in pairs and pitted against each other, so only those

with better scores are selected. Pairwise matching is performed as often as necessary to cover the indicated population size. In other words, the same individual can be tested on more than one occasion and even selected for the next phase. However, in the next generation, it will certainly be crossed with other individuals, and its offspring will subsequently be subjected to different mutations.

5.1.4 Crossover

The crossover operation simulates a reproduction process between individuals, where their respective genetic materials are crossed to procreate offspring. This operation occurs with a fairly high probability modifiable in the jMetal framework. How this genetic material is crossed is what leads to multiple possible operators. Depending on the characteristics of the problem, the type of crossover chosen will have better or worse results. However, the best way to check the choice of a good operator is by testing and comparison.

The jMetal framework offers a wide range of crossover operators. These include *SBXCrossover* (Simulated Binary Crossover) [218], *BLXAlphaCrossover* (Blend Alpha Crossover) [219], *DifferentialEvolutionCrossover* [220], *NPointCrossover*, *NullCrossover* and *WholeArithmeticCrossover*. Finally, the *SBXCrossover* operator was chosen after several scores tests. Firstly, this operator was the one that reported the best results concerning the others. Secondly, its choice was the most coherent if considered a linear expression between the two weight vectors. It tends to maintain the feasibility of the solution and reduce the distortion cost produced by the repairer.

5.1.5 Mutation

After crossing the individuals of the previous generation, the offspring are subjected to a mutation process to incorporate new genetic material into the population. Otherwise, the resolution of the problem would be completely limited by the genetic content of the initial population, which reduces the search procedure and conditions the solution to the initial decisions of the algorithm. This operation occurs with a fairly low probability, again modifiable from the jMetal framework. As with the crossover stage, jMetal integrates a wide variety of operators to cover this phase of the evolutionary algorithm. Specifically, the mutation operators available are the following: *PolynomialMutation*, *CDGMutation*, *LinkedPolynomialMutation*, *GroupedPolynomialMutation*, *GroupedAndLinkedPolynomialMutation*, *SimpleRandomMutation*, *UniformMutation*, *NonUniformMutation* and *NullMutation*. Finally, after verifying the good results generated in combination with *SBXCrossover*, the *PolynomialMutation* was chosen. In

this case, the modification of the mutated values is compensated by the rest of the vector weights by repairing individuals.

5.1.6 Output

After completing the number of evaluations set in the input parameter, `GENECI` selects the best vector of weights found during the execution and produces an output consisting of 5 files:

- List of optimized interactions with their respective consensus confidence values.
- Binary network resulting from applying the selected cut-off criterion to the previous list.
- Weights assigned to the different techniques in the final solution.
- A plain text file with the evolution of the fitness values.
- Optionally, a pdf file with a graphical representation of this evolution.

5.2 Experimentation

GENECI has a fairly large number of parameters, which are shown in Table 5.2 along with their respective descriptions. Before elaborating on the real experimentation of this chapter, it was necessary to carry out a parameterization exercise to guarantee the optimal performance of the algorithm.

5.2.1 Parameter Settings

Similarly, before parameter refinement, it was necessary to ensure a certain consistency between the fitness values of the individuals and their accuracy in the prediction of gene networks. That is to say, a good fitness value should translate into a good quality index in the subsequent network prediction. To guide the evolutionary algorithm to some extent, several networks (mainly from the DREAM challenges) were tested with different values of the parameters associated with the weights of the fitness function terms. Finally, the combination with the best accuracy results (regardless of their fitness values) was chosen. This combination was 0.75 for the first term and 0.25 for the second.

Once this was done, the rest of the parameters were tested to optimize the fitness values. Since testing all combinations of values was completely unfeasible,

Table 5.2: GENECI input parameters.

Parameter	Description
–confidence-list	CSV file paths with trusted lists.
–gene-names	Path to the TXT file with the name of the genes separated by comma and without space. If not specified, only genes specified in the confidence lists will be considered.
–crossover	Crossover operator.
–crossover-probability	Crossover probability.
–mutation	Mutation operator.
–mutation-probability	Mutation probability.
–repairer	Repairer to keep the sum of weights equal to 1.
–population-size	Population size.
–num-evaluations	Number of evaluations.
–cut-off-criteria	Cut-off criteria for network binarization.
–cut-off-value	Numeric value associated with the selected criterion.
–Q-weight	Weight associated with term *Quality*.
–T-weight	Weight associated with term *Topology*.
–threads	Number of threads to be used during parallelization. By default, the maximum number of threads available in the system is used.
–graphics	Graphical representation of the evolution of the fitness value.
–output-dir	Path to the output folder.

an incremental procedure was carried out to try to progressively fix the values of the parameters, starting with the analysis of the most fundamental ones and ending with those of lesser importance.

Parameters related to crossover and mutation probabilities were set before. It is already known in the literature that an adequately constructed evolutionary algorithm should respond well to a high crossover probability and a low mutation probability [221, 222]. However, testing their values ensures the consistency of the implementation. Although the crossover probability seemed to be clearly fixed at 0.9, the mutation probability varied depending on the number of consensual lists. After reviewing the literature, it could be seen how the recommended mutation probability for these cases is the maximum between 0.01 and $1/n$ [215, 223], where n is the length of the vector, which in this case is the number of lists provided in the input. Therefore, since having a number less than

0.01 would mean trying to agree on more than 100 lists (which is infeasible), $1/n$ was set as the optimal value associated with the mutation probability.

The next parameters to be optimized were those related to the repairer and cut-off criterion. As mentioned in their respective sections, *StandardizationRepairer* and *MinConfDist* were finally established as the optimal values for these parameters. In this case, since both parameters are quite specific to our problem, instead of doing point executions, we proceeded to perform a systematic test on all data sets. For reasons of length, these results can be found in the supplementary material in the repository. The winners were easily predictable since the alternatives lacked adequate meaning. First, concerning the repairers, it was to be expected that the greedy ones would offer worse results. There was some randomness in its operation, and it did not fully maintain the essence of the vector it was intended to repair. On the other hand, in the cut-off criteria, the justification discussed in previous sections regarding the reliability based on the confidence values together with the distance between weight means and distance concerning the median confidence allowed to expect that the criterion contemplating both metrics would indeed be the most effective.

Finally, the parameters of population size and the number of evaluations remain to be analyzed. It is evident that the higher the value given to them, the higher the quality of the result obtained, but the more execution time they consume. To find a certain balance, several combinations were tested to find the one that would ensure the algorithm's convergence at a suitable stage under a reasonable number of iterations. It should be mentioned that, regardless of the size of the input network, the number of techniques applied in the experimentation is the same, so the vector of weights to be optimized is always of the same size. This means that, although the evaluation is slower for large networks, the possible combinations of weights assigned to the lists cover the same search space as for the rest of the networks. Finally, a population size of 100 individuals and a total of 50,000 evaluations were established for the experimentation addressed in this chapter.

5.2.2 Experimental Procedure

After setting all the `GENECI` parameters, an experimental procedure was constructed to demonstrate the validity of the work carried out and the benefits of the proposed strategy. The first part of this study takes data from academic benchmarks to quantify the accuracy of `GENECI`. First, data from some of the DREAM challenges [142] (specifically editions 3, 4 and 5) are used, as they have been extensively studied in the literature [224, 225, 226, 227, 228, 229] and

provide specific evaluation scripts that allow us to compare accuracy values with other research articles. And secondly, the IRMA 5-gene network [173] is considered, whose gold standard allows us to evaluate the quality of the results as a binary classification problem. Finally, GENECI is confronted with a real-world biological network of melanoma patients [196] whose interactions are validated by specific literature searches. For more details on these data, see section 4.1.

For each dataset, the process starts by inferring their corresponding gene regulatory networks using all the individual techniques integrated into the proposal. For the benchmark data, the predictive capacity of the results provided by the individual techniques is evaluated to conduct a subsequent comparison exercise concerning the quality of the GENECI consensus networks. Specifically, metrics AUROC (Area Under the ROC curve) and AUPR (Area Under the Precision-Recall curve) are calculated. The area under the receiver operating characteristic curve (AUROC) is a single scalar value that quantifies the overall performance of a binary classifier. Its value is bounded by the interval [0.5-1.0], where the minimum value represents the performance of a random classifier, and the maximum value is associated with a perfect classifier. Secondly, the area under the precision-recall curve (AUCPR) is a model performance metric that has been recognized as useful for classification performance assessment for unbalanced binary responses in bioinformatics [230]. This is the case for predicting interactions between genes that form the GRNs, as the number of truly interconnected genes is small compared to all the possible connections. Its value increases the better the classifier is evaluated.

For the DREAM challenges, a subcommand integrated into the package is used to call the evaluation scripts presented in the respective challenges. Another generic subcommand is used for the IRMA data that treats the case as a binary classification problem. Subsequently, a total of 25 independent runs are elaborated by the evolutionary algorithm (except for the last DREAM5 network due to its size), and the quality of the prediction made for each is validated. Finally, a comparison is made between the median AUROC and AUPR values of these 25 runs and the individual values of the different techniques.

The GENECI result is studied for real-world data by validating the presence of the main interactions reported in the literature from a biomedical point of view.

5.3 Results and Discussion

This section presents the results provided by GENECI for the experimental procedure described previously. A section is dedicated to each data set, illustrating the

Table 5.3: Accuracy values for DREAM3 and size 10 networks. AUPR and AUROC values are provided for each technique and problem, highlighting in bold the results obtained by GENECI and the best obtained by any of the individual techniques.

Técnica	D3_10_Ecoli1		D3_10_Ecoli2		D3_10_Yeast1		D3_10_Yeast2		D3_10_Yeast3	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
ARACNE	0.561	0.1529	0.524	0.1852	0.6269	0.1546	**0.6031**	0.417	0.5043	0.2463
BC3NET	0.6116	0.2146	0.4804	0.2158	0.5606	0.2359	0.5914	0.3459	0.4846	0.228
C3NET	0.5754	0.1599	0.5218	0.1832	0.5719	0.1368	0.6003	0.414	0.5167	0.2538
CLR	0.5719	0.1477	0.4542	0.1599	0.5788	0.1724	0.5782	0.3783	0.4893	0.2619
GENIE3_ET	**0.6157**	0.1762	0.6524	0.2172	0.525	0.1212	0.5858	0.3417	**0.5809**	0.2855
GENIE3_GBM	0.5673	0.139	**0.7458**	0.2848	0.4863	0.1341	0.5643	**0.4183**	0.5381	0.2695
GENIE3_RF	0.5938	0.1598	0.6818	0.2369	0.5	0.1064	0.5545	0.345	0.5689	0.3072
KBOOST	0.5949	0.1739	0.648	0.2392	0.3438	0.0928	0.5415	0.3246	0.3168	0.1786
MRNETB	0.5472	0.141	0.4631	0.1685	0.5487	0.1285	0.5754	0.4043	0.4485	0.223
MRNET	0.5155	0.1469	0.5116	0.1785	0.5206	0.1264	0.5788	0.4122	0.4402	0.2318
PCIT	0.5455	**0.2987**	0.4862	0.1626	0.5631	0.1694	0.412	0.2336	0.5675	**0.3501**
TIGRESS	0.481	0.1174	0.6569	**0.4301**	**0.6763**	**0.242**	0.4892	0.2491	0.4305	0.2013
Median GENECI	**0.5627**	**0.1707**	**0.6089**	**0.2468**	**0.5175**	**0.1311**	**0.5982**	**0.3645**	**0.5127**	**0.2711**
Best GENECI	**0.5685**	**0.1791**	**0.6196**	**0.2523**	**0.5275**	**0.1369**	**0.6025**	**0.3956**	**0.5287**	**0.3245**

precision values obtained in each case and graphical representations that allow visualizing network topologies, distribution of fitness values, weights assigned by GENECI and a series of comparisons between the results of different inference techniques.

5.3.1 Benchmarks

DREAM3 - Size 10

First, the results for the 10-node networks of the DREAM3 challenge are shown. Table 5.3 shows the AUROC and AUPR values for each of the individual techniques and the median of these same metrics for the 25 independent runs of the evolutionary algorithm. In addition, for comparative purposes, the best value obtained by the inference techniques and the one achieved by GENECI are highlighted in bold for each column. In this case, the values of the accuracy metrics reported for the consensus networks are in a competitive range but without standing out from the rest. The exception occurs for the AUROC obtained for the second yeast network, where the distance between the best result of the techniques and that of GENECI is relatively small. However, this case is considered insignificant, considering its isolated character and the low precision quality reported by the individual techniques.

The explanation of these results is that the second term of the fitness func-

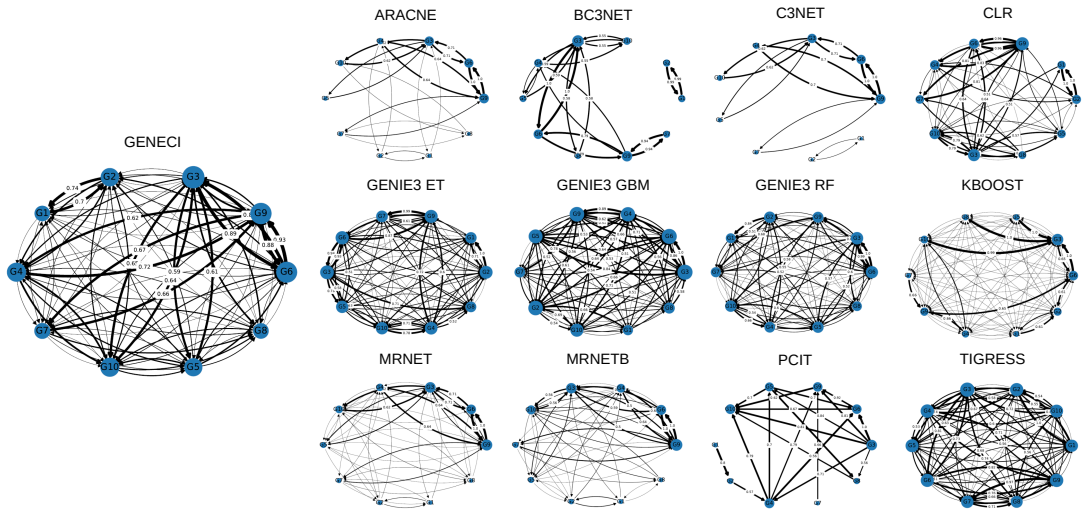


Figure 5.2: For the first 10-gene yeast network of the DREAM3 challenge, the gene networks inferred by the individual techniques and the consensus gene network computed in the run whose AUROC corresponds to the median exposed in Table 5.3 are illustrated. Graphs attempt to represent gene regulatory networks by setting up genes in the form of nodes and interactions through links. In addition, it can be seen that the directionality and confidence of these interactions are represented in these networks.

tion is practically frozen for cases of such a small size. When faced with 10-node networks, the evolutionary algorithm does not seem to have enough margin to carry out the optimization part aimed at improving the consensus network topology. However, this is not considered a problem if one remembers that the goal of `GENECI` is to optimize the consensus of real-world gene networks, where the number of transcription factors is much larger than that contained in this subchallenge.

Figure 5.2 shows the graphs associated with each of the networks inferred by the individual techniques and the consensus network whose AUROC corresponds to the median of the runs. These graphs show the directionality of the interactions (direction of the arrows), the degree of the genes (size of the nodes) and the intensity of the relationships (thickness of the links and numerical specification for the highest values). In addition to appreciating the small size of these networks, it can be seen how the techniques that have obtained the best results in Table 5.3 (TIGRESS and those derived from GENIE3) present random networks that are far from the scale-free configuration that usually appears in real-world gene networks of larger size. This means that the `GENECI` target is far from the topological characteristics of the gold standard in these cases, which

Table 5.4: Accuracy values for DREAM3 and size 50 networks. In this table, a gene network is contemplated for each pair of columns, where in each row the AUPR and AUROC values are provided for each inference technique.

Técnica	D3_50_Ecoli1		D3_50_Ecoli2		D3_50_Yeast1		D3_50_Yeast2		D3_50_Yeast3	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
ARACNE	0.4893	0.0241	0.539	0.0424	0.537	0.0505	0.5252	0.0761	0.5283	0.0887
BC3NET	0.5015	0.0263	0.5271	0.0372	0.5455	0.0429	0.5144	0.0715	0.5245	0.084
C3NET	0.5083	0.0265	0.5209	0.0391	0.5363	0.0508	0.5198	0.0764	0.5244	0.0868
CLR	0.5831	0.0334	0.627	0.0606	0.5401	0.0504	0.5292	0.0814	0.5491	0.1014
GENIE3_ET	0.5338	0.0294	0.6373	0.0783	0.5463	0.0514	0.5709	0.0872	0.5752	0.0965
GENIE3_GBM	0.4951	0.0299	0.6269	**0.087**	**0.5704**	**0.0778**	0.571	**0.0906**	0.5775	**0.1058**
GENIE3_RF	0.5563	0.0338	0.6318	0.0811	0.5665	0.07	**0.5892**	0.0895	0.5739	0.1004
KBOOST	0.5459	0.0277	0.528	0.0379	0.4628	0.0347	0.4659	0.0668	0.5168	0.0738
MRNETB	**0.6046**	0.0363	**0.6401**	0.057	0.5467	0.0557	0.5357	0.0755	0.5507	0.0992
MRNET	0.5803	0.0329	0.6237	0.0557	0.5409	0.0507	0.5383	0.0786	0.5474	0.0983
PCIT	0.5765	**0.0403**	0.5953	0.0677	0.5636	0.0588	0.499	0.0644	0.539	0.0843
TIGRESS	0.5794	0.0289	0.3846	0.0246	0.5591	0.0335	0.5876	0.084	**0.5807**	0.0983
Median GENECI	**0.5998**	**0.0348**	**0.6461**	**0.0713**	**0.5517**	**0.0657**	**0.5688**	**0.0839**	**0.574**	**0.1001**
Best GENECI	**0.6027**	**0.0349**	**0.6475**	**0.0719**	**0.5553**	**0.0662**	**0.5694**	**0.0846**	**0.5821**	**0.1014**

again explains the limited results shown.

DREAM3 - Size 50

This section shows the results obtained for the 50-node networks of the DREAM3 challenge. In Table 5.4, it can be seen that the increase in the size of the networks to be inferred has led to better results than in the previous section. This is because a size of 50 nodes already gives a certain margin to the evolutionary algorithm to optimize the topological characteristics of the consensus networks. In this case, medians of the AUROC and AUPR values of `GENECI` are quite close to the best results of the individual techniques. In addition, it is worth mentioning that the maxima of these metrics are selected individually so that, in many cases, the method that provides the best result concerning AUROC is not the same as the best about AUPR. Therefore, the fact that `GENECI` is able to approach the maximums of both metrics (or even surpass them), is in many cases an improvement over any of the individual techniques.

Figure 5.3 shows boxplots representing the weights assigned to the different techniques by the final solutions of the 25 independent runs. An additional diagram concerning the fitness values of these solutions is depicted at the bottom right, where the variation of the achieved values is shown for each network.

Regarding the assignment of weights, it can be seen that in most of the networks, the proposed solutions throughout the 25 runs are quite similar. However,

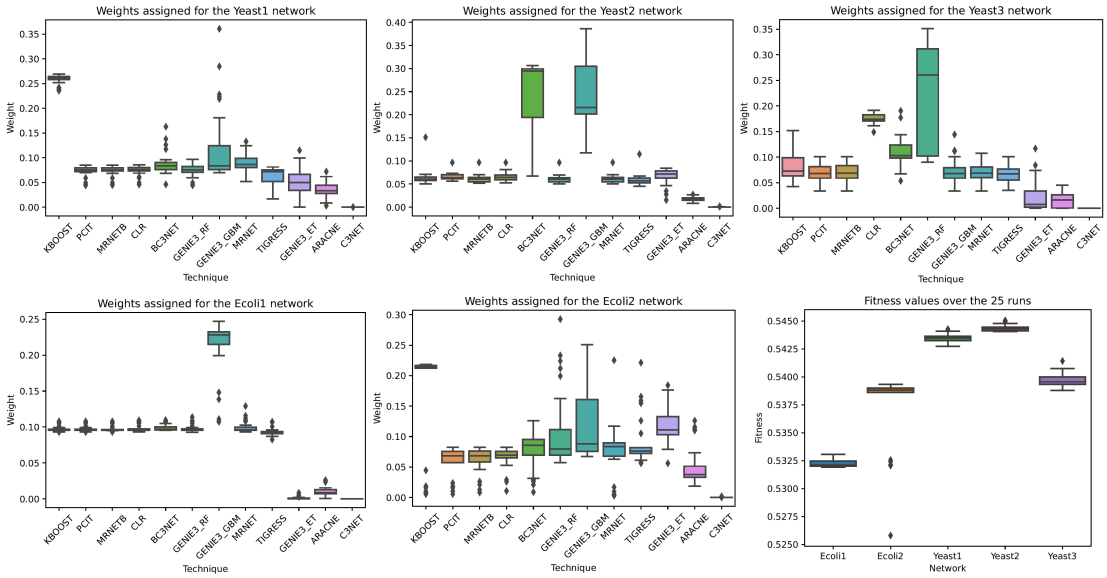


Figure 5.3: Boxplots of the fitness values and weights over the 25 independent runs. The first 5 graphs represent the distribution of the weights assigned by GENECI across all the runs for each of the techniques. Finally, the sixth figure shows the distribution of the fitness values obtained in the different runs performed.

there is the exception of the Ecoli2 network, which in addition to showing outliers in the fitness boxplot also presents a slightly more random distribution of weights. This may be because this network presents more local minima that hinder the algorithm's progress. Most runs seem to have stalled at one of them, as there are sporadic runs with better results. Consequently, both the distributions of good solutions and those related to premature convergences coexist in the boxplots of the weights, which explains the variability shown in the graph.

From a different perspective, specific techniques in certain networks do not seem to converge to a given weight. Two different situations can be seen in the illustrated graphs. First, in the case of Yeast2 graph, BC3NET and GENIE3_GBM obtain quite different weights depending on the execution. The fact that the fitness values remain constant is a sign that their variability is not related to the algorithm's convergence. Moreover, since all other techniques remain constant, it can be deduced that the weight increase in one is usually reflected in a decrease in the weight of the other, although being this balance completely indifferent with regard to the fitness value of the solution. The only case in which this is possible is when the techniques infer quite similar networks, and therefore the granting of greater weight to one or the other is practically indiscernible for the consensual network.

Table 5.5: Accuracy values for DREAM3 and size 100 networks. The AUPR and AUROC values are presented in two clearly distinguishable bands. The first band shows the precision values for the individual inference techniques, while the second band shows the values obtained by GENECI after the consensus of the techniques, distinguishing between the median of the runs and the best result obtained from them.

Técnica	D3_100_Ecoli1		D3_100_Ecoli2		D3_100_Yeast1		D3_100_Yeast2		D3_100_Yeast3	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
ARACNE	0.5512	0.0238	0.5323	0.0187	0.553	0.0332	0.5216	0.053	0.5126	0.064
BC3NET	0.5442	0.0175	0.5286	0.0175	0.523	0.0191	0.5148	0.0435	0.5063	0.0569
C3NET	0.5413	0.0233	0.5189	0.017	0.5232	0.0263	0.5165	0.0507	0.5108	0.0638
CLR	0.659	0.0311	0.5909	0.0268	0.5553	0.0472	0.521	0.0557	**0.5284**	0.0693
GENIE3_ET	0.6596	0.0338	0.5822	0.0374	0.6235	0.0496	0.5343	0.0528	0.5176	0.0672
GENIE3_GBM	0.6176	0.0363	0.5729	0.0455	**0.6515**	**0.0629**	**0.5566**	**0.0621**	0.5274	0.0713
GENIE3_RF	**0.6673**	**0.042**	0.6001	**0.0506**	0.6465	0.0557	0.5548	0.0602	0.5269	**0.0719**
KBOOST	0.4975	0.0153	0.506	0.0132	0.4934	0.0193	0.4692	0.0389	0.4721	0.0516
MRNETB	0.6422	0.0332	**0.6045**	0.0235	0.5478	0.0357	0.5116	0.0482	0.5246	0.0658
MRNET	0.6352	0.032	0.6002	0.0229	0.5505	0.032	0.513	0.05	0.5284	0.0671
PCIT	0.5972	0.0236	0.5879	0.0242	0.5133	0.0269	0.4909	0.0432	0.5084	0.0564
TIGRESS	0.6257	0.0178	0.557	0.0258	0.5051	0.0163	0.5251	0.0412	0.4863	0.0504
Median GENECI	**0.6813**	**0.0368**	**0.6093**	**0.0347**	**0.5869**	**0.0433**	**0.5305**	**0.0577**	**0.5291**	**0.0693**
Best GENECI	**0.6918**	**0.0373**	**0.6115**	**0.0351**	**0.5905**	**0.0436**	**0.5336**	**0.0583**	**0.5299**	**0.0694**

Second, in the case of GENIE3_RF for the Yeast3 network, there is a quite similar situation to the previous one, except that on this occasion, the increase or decrease in the weight of this technique is uniformly assumed by the rest. This may be a consequence of the fact that the confidence values of this list are quite close to the median of the remaining ones, and therefore, voting during consensus is somewhat redundant.

DREAM3 - Size 100

Finally, the experimentation on the DREAM3 challenge networks is concluded by presenting the results associated with the 100-node networks. Similar to the previous section, Table 5.5 shows how the results provided by GENECI are practically at the same level as those obtained by the best individual techniques.

In fact, it is observed that the increase in size continues to bring benefits to the results. While in the 50-node table, only in one case GENECI came to exceed the maximum AUROC of the individual techniques, in the 100-node table, this occurs for 3 of the 5 networks. This means that the optimization performed on the topological characteristics of the network becomes more meaningful the larger the size of the network to be inferred.

Figure 5.4 shows the Ecoli1 network agreed upon by GENECI in the run cor-

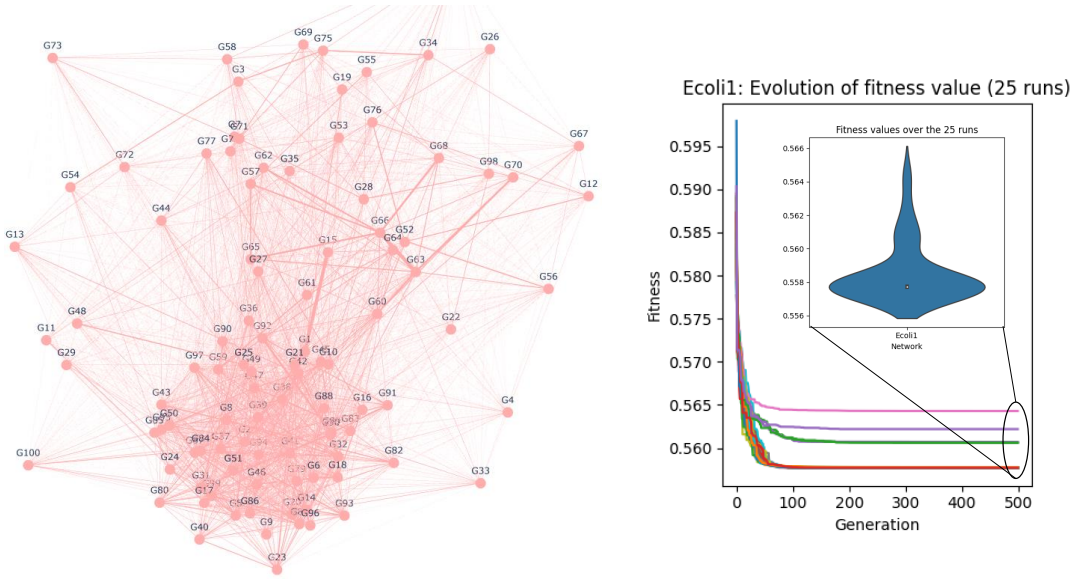


Figure 5.4: For the first 100-gene Ecoli network of the DREAM3 challenge, the consensus gene network calculated in the run whose AUROC corresponds to the median illustrated in Table 5.5 is plotted on the left. On the right, the evolution of the fitness values obtained during the 25 runs and a violin plot representing the distribution of their corresponding final values.

responding to the median of the AUROC values. As can be seen, the network is clearly scale-free, as the degrees of the different nodes are distributed non-uniformly. The interactive 3D representation generated using the Python package built in this proposal is very useful for network analysis, allowing rotations, overlapping techniques, zooming, the query of confidence values, etc. The fitness curves for the 25 runs performed on this network are shown on the right. In them, one can visualize how a few runs seem to have stagnated at a local minimum, which is also appreciable in the violin plot located at the top right. After several tests, it has been shown that `GENECI` tends to converge well before 50,000 evaluations, so reducing this value would represent some gain concerning the execution time without harming the quality of the results.

Finally, it is worth mentioning that the AUROC and AUPR values for some of the individual techniques had already been calculated previously in the literature. Specifically, some accuracy values for the DREAM3 challenge networks are reported in [226, 228], where the resemblance of the results to those obtained in this work provides some reliability in the study addressed. It is striking to note the low quality of accuracy that is achieved today in the task of inferring GRNs,

Table 5.6: Accuracy values for DREAM4 and size 10 networks. For each gene regulatory network included in this dataset (columns), the AUPR and AUROC values are shown after comparing the networks inferred by the different techniques (rows) with the respective gold standards.

Técnica	D4_10_1		D4_10_2		D4_10_3		D4_10_4		D4_10_5	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
ARACNE	0.6236	0.331	0.4489	0.171	0.5618	0.2771	0.6693	0.3033	0.6688	0.3168
BC3NET	0.7236	0.4237	0.4954	0.1841	0.5671	0.1936	0.6349	0.2785	0.6613	0.2166
C3NET	0.6636	0.354	0.4865	0.1823	0.5262	0.2625	0.6344	0.2866	0.6966	0.3497
CLR	0.6507	0.3483	0.4861	0.1881	0.5947	0.2693	0.6893	0.2694	0.6912	0.3648
GENIE3_ET	**0.8631**	**0.4533**	0.614	0.2401	0.6533	0.2437	0.7073	0.2686	**0.8259**	0.4167
GENIE3_GBM	0.664	0.2638	0.5971	0.2296	**0.696**	0.3017	0.6883	0.2658	0.6613	0.3193
GENIE3_RF	0.8284	0.4441	**0.6326**	0.2546	0.6898	**0.3486**	0.6883	0.3097	0.8024	**0.4379**
KBOOST	0.5858	0.2324	0.603	0.2368	0.576	0.2533	**0.7383**	**0.3412**	0.6667	0.2393
MRNETB	0.6867	0.3614	0.4907	0.1838	0.6302	0.2936	0.6718	0.317	0.6704	0.34
MRNET	0.6493	0.3439	0.5046	0.1876	0.5422	0.2744	0.7118	0.3173	0.672	0.3416
PCIT	0.5884	0.3262	0.5819	**0.2948**	0.5649	0.2084	0.5395	0.2562	0.5577	0.2579
TIGRESS	0.5973	0.3257	0.614	0.2191	0.4871	0.1674	0.4915	0.15	0.4017	0.1268
Median GENECI	**0.7689**	**0.4371**	**0.5887**	**0.2709**	**0.6933**	**0.274**	**0.7493**	**0.3445**	**0.7906**	**0.4248**
Best GENECI	**0.7733**	**0.4393**	**0.5938**	**0.2724**	**0.6987**	**0.2777**	**0.7572**	**0.3561**	**0.8077**	**0.4638**

and it is for this reason that it remains a significant area of research.

DREAM4 - Size 10

Table 5.6 shows the AUROC and AUPR values for each of the individual techniques and the median of these metrics for the 25 independent runs of the proposal. Indeed, as in the previous challenge, the nodes' scarcity and the networks' small size lead to moderate results for GENECI. This is visible in networks 1, 2 and 5, as well as in the AUPR of network 3. However, the exception of network 4 stands out, where GENECI achieved competitive results. In this case, unlike the exception seen in DREAM3 for networks of the same size, the AUROC and AUPR values provided by the individual techniques are neither homogeneous nor of low quality, so this time it is considered a merit on the part of the algorithm.

To analyze this sporadic behaviour, Figure 5.5 shows for Net-4 10-gene the graphs related to the different individual techniques and the GENECI consensus gene network corresponding to the median. In the graphs, despite the small size of the network, a certain scale-free distribution is shown, where the most interconnected node also has the most intense relationships. This means that when GENECI rewards individuals that increase the degree of the only existing hub (1/10 = 10% of genes), it is actually bringing the consensus network closer to the gold standard. This behaviour in the rest of the networks is unsatisfactory

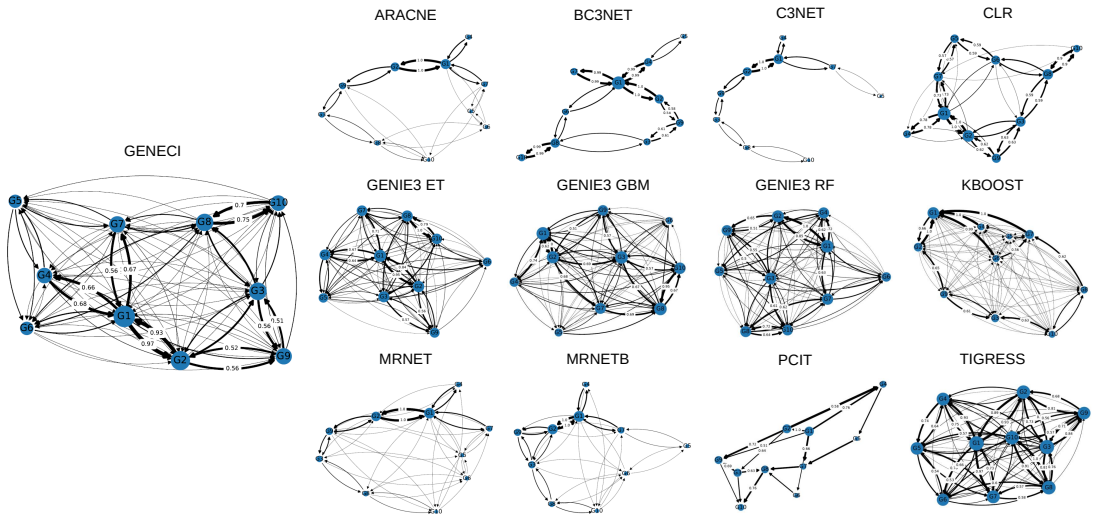


Figure 5.5: For Net-4 10-gene of the DREAM4 challenge, the gene networks inferred by the individual techniques and the consensus gene network computed in the run whose AUROC corresponds to the median exposed in Table 5.6 are illustrated. In these graphs we can see how each gene corresponds to a node and each edge to a specific gene interaction. The directionalities are expressed by arrows and the confidence values by the thickness of the links, even specifying their value when this is highly significant.

because there is no single hub in the network since, as explained in the section on DREAM3, networks of this size tend to have a random configuration.

DREAM4 - Size 100

Again, incorporating a larger number of nodes favours the optimization of the topological characteristics of the consensus network. Table 5.7 shows the results for the 100-node networks (DREAM4). It should be remembered that, as in the rest of the sections, obtaining good precision values by GENECI implies incorporating a reliable method that guarantees outstanding results for networks of various densities and characteristics (discarding the excessively small ones of 10 nodes). This implies that when it is desired to infer a gene regulatory network whose structure is unknown, the application of GENECI provides a reliable and effective resolution method. This is not feasible with the individual techniques since, as shown in the tables, they tend to provide good results only for specific subsets of problems, reporting less satisfactory results for the rest.

Unlike the other cases, on this occasion, GENECI seems to overcome the individual inference techniques through the AUPR values. This occurs in 3 of the

Table 5.7: Accuracy values for DREAM4 and size 100 networks. This table shows the results of the evaluation scripts run on each of the individual (first band) and consensus (second band) results for each problem network (columns). The best value of the individual techniques and the values of the consensus networks per GENECI (best and median of all runs) are shown in bold.

Técnica	D4_100_1		D4_100_2		D4_100_3		D4_100_4		D4_100_5	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
ARACNE	0.5568	0.0317	0.5412	0.0453	0.559	0.0673	0.5525	0.0419	0.5826	0.0594
BC3NET	0.5629	0.0334	0.5312	0.0315	0.5905	0.0617	0.5537	0.06	0.5939	0.0565
C3NET	0.5345	0.029	0.518	0.038	0.5522	0.0664	0.54	0.0383	0.5622	0.0575
CLR	0.6963	0.048	0.6291	0.0578	0.7079	0.1036	0.6654	0.0621	0.6768	0.0764
GENIE3_ET	**0.7733**	**0.0756**	0.6741	0.0526	0.7289	0.1143	0.7046	0.0683	0.7555	0.0842
GENIE3_GBM	0.7608	0.0567	**0.6929**	0.0624	0.719	0.0983	0.7046	0.0634	**0.7707**	0.0815
GENIE3_RF	0.756	0.062	0.6873	0.0633	**0.7411**	**0.1182**	**0.7195**	0.0698	0.7694	0.082
KBOOST	0.6135	0.0461	0.5234	0.0431	0.5764	0.054	0.5247	0.0367	0.5171	0.0414
MRNETB	0.6848	0.047	0.6334	**0.0639**	0.7169	0.1076	0.6667	0.0604	0.6798	0.0739
MRNET	0.6771	0.0446	0.6322	0.0583	0.7124	0.1022	0.6622	0.0568	0.6786	0.081
PCIT	0.6172	0.0614	0.5649	0.0486	0.6149	0.1017	0.5988	**0.0772**	0.6339	**0.0894**
TIGRESS	0.6581	0.0261	0.5595	0.0617	0.6476	0.0319	0.6281	0.0402	0.6509	0.0312
Median GENECI	**0.7613**	**0.0674**	**0.6631**	**0.0685**	**0.7316**	**0.1241**	**0.7078**	**0.0742**	**0.7368**	**0.1143**
Best GENECI	**0.7623**	**0.0676**	**0.665**	**0.0691**	**0.7396**	**0.1265**	**0.7097**	**0.0748**	**0.7377**	**0.1144**

5 exposed networks, highlighting notably the consensus addressed on network number 5 of this subchallenge.

Figure 5.6 shows the consensus gene network corresponding to the median calculated by GENECI. Again, the distribution of node degrees is not uniform, with a large group being mostly interconnected in contrast to the rest of the network. On the right are the fitness curves, which in this case, reflect the correct convergence of all the executions and the end in a fixed and concrete fitness value.

Finally, it is worth mentioning that, as with DREAM3, previous works have tested different techniques for inferring networks from this challenge. Results and quality metrics related to DREAM4 networks can be seen in [224, 225, 226, 227, 228, 229], where again the kinship between these values and those calculated in this chapter bring some confidence to this study.

DREAM5

Due to its large size, 15 runs have been carried out for the last network instead of 25. Therefore, in this case, the values reported for Net 4 in Table 5.8 refer to the AUROC and AUPR values concerning those 15 runs (denoted by *). Due to the slowness of the single TIGRESS technique, it has been finally discarded from

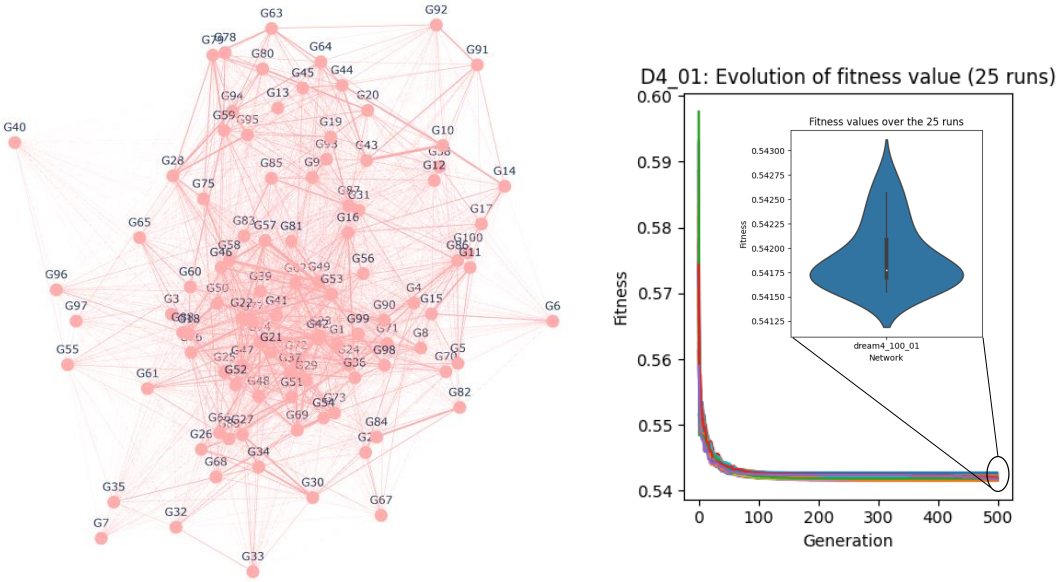


Figure 5.6: For the first 100-gene network of the DREAM4 challenge, the consensus gene network calculated in the run whose AUROC corresponds to the median illustrated in Table 5.7 is plotted on the left. On the right, the evolution of the fitness values obtained during the 25 runs and a violin plot representing the distribution of their corresponding final values.

the optimization process for this challenge. In the table, the results are again favorable. There is an exception for the AUROC obtained in the third network of the challenge. However, it is worth mentioning that the distance observed concerning the best individual technique is due to its accuracy being a clear outlier for the rest of the values. These results finally demonstrate that `GENECI` responds satisfactorily to the input of gene expression data from large networks with biological support.

For individual inference techniques, similar results are reported in [231, 224], where the absence of the second gene network during the evaluation process is indeed noted.

IRMA

Table 5.9 shows the AUROC and AUPR results for each of the individual techniques and the median of the 25 independent runs of `GENECI`. It should be noted that the evaluation process carried out on these networks differs from the previous ones. In this case, IRMA does not offer specific evaluation scripts, so the

Table 5.8: Accuracy values for DREAM5 networks. AUPR and AUROC values are provided for each technique and problem, highlighting in bold the results obtained by GENECI and the best obtained by any of the individual techniques.

Técnica	D5_1		D5_3		D5_4	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
ARACNE	0.5538	0.1011	0.5131	0.0296	0.5005	0.0176
BC3NET	0.5673	0.0966	0.5149	0.0257	0.5006	0.0176
C3NET	0.5393	0.0865	0.5071	0.0243	0.5005	0.0176
CLR	0.7402	0.2234	0.5892	0.0602	0.5211	0.0212
GENIE3_ET	**0.8149**	0.2502	**0.6632**	**0.0879**	0.543	0.0222
GENIE3_GBM	0.7952	**0.3042**	0.6108	0.0682	0.5318	0.0208
GENIE3_RF	0.8135	0.2802	0.6535	0.0844	**0.5494**	**0.0223**
KBOOST	0.4679	0.0634	0.5572	0.0445	0.5037	0.0182
MRNETB	0.7421	0.2	0.5948	0.0697	0.52	0.0195
MRNET	0.7404	0.2054	0.5943	0.0557	0.521	0.0196
PCIT	0.6761	0.1712	0.5751	0.0621	0.5173	0.0194
Median GENECI	**0.8007**	**0.2788**	**0.6211**	**0.0751**	**0.5316***	**0.0214***
Best GENECI	**0.8024**	**0.2801**	**0.6264**	**0.0762**	**0.5317***	**0.0214***

Table 5.9: Accuracy values for IRMA networks. In this table, a gene network is contemplated for each pair of columns, where in each row the AUPR and AUROC values are provided for each inference technique.

Técnica	IRMA_switch-off		IRMA_switch-on	
	AUROC	AUPR	AUROC	AUPR
ARACNE	0.6667	0.6815	0.6667	0.6815
BC3NET	0.5833	0.5679	0.5833	0.5679
C3NET	0.6667	0.6815	0.6667	0.6815
CLR	0.6111	0.5339	0.7222	0.609
GENIE3_ET	0.6667	0.6815	**0.8611**	**0.7865**
GENIE3_GBM	0.5	0.4	0.75	0.7759
GENIE3_RF	0.6944	0.6261	0.75	0.7759
KBOOST	**0.7778**	**0.7099**	0.6111	0.5339
MRNETB	0.6667	0.6815	0.6667	0.6815
MRNET	0.6667	0.6815	0.6667	0.6815
PCIT	0.5	0.4	0.5	0.4
TIGRESS	**0.7778**	0.6	0.7778	0.6
Median GENECI	**0.8611**	**0.7865**	**0.8889**	**0.75**
Best GENECI	**0.8611**	**0.7865**	**0.8889**	**0.75**

confidence lists reported by the different techniques have been binarized to perform this task. Subsequently, the generic evaluation subcommand has been used,

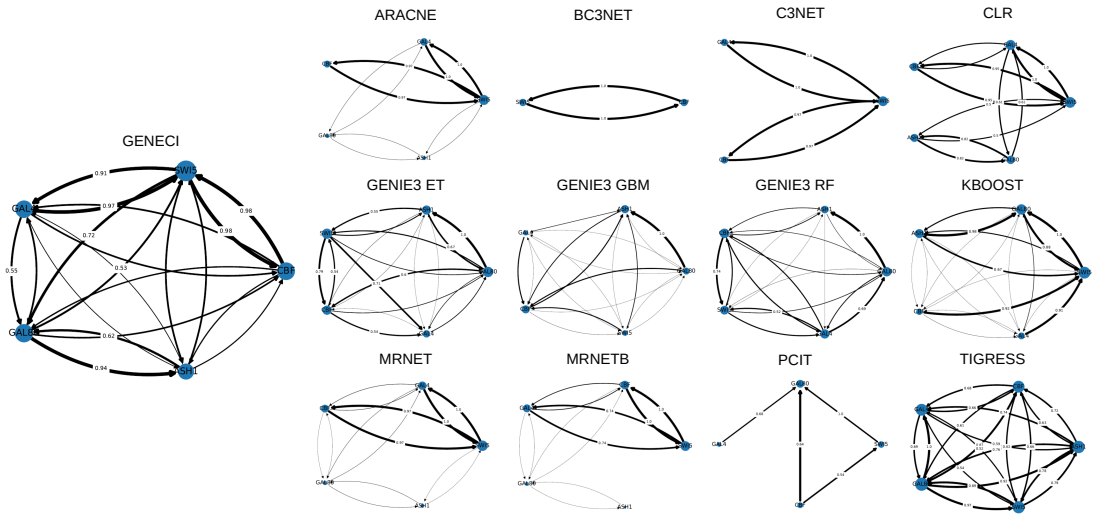


Figure 5.7: For the "switch on" IRMA network, the gene networks inferred by the individual techniques and the consensus gene network calculated in the run whose AUROC corresponds to the median exposed in Table 5.9 are illustrated. The graphs shown in this figure try to represent the different inferred networks, arranging the genes in the same layout in order to facilitate visual comparison between them.

facing the binary classification problem.

Finally, it can be seen that after applying the same evaluation criteria to all the networks, `GENECI` is again shown to perform well (see Table 5.9). It outperforms the AUROC and AUPR maxima of the individual techniques for most cases, with a remarkably significant difference in the case of the "switch off" network. The graphs reported by the different inference techniques and the consensus network constructed by `GENECI` are shown in Figure 5.7. It can be seen that the network is really small, with a clear disagreement in the set of proposals when establishing the existing interactions.

The IRMA network has been frequently used in the literature to test various techniques for network reconstruction [231, 226, 227, 228, 229]. However, the lack of a rigorous official criterion for measuring the accuracy of the different tools has given the scientific community a certain margin to choose the procedure that best suits its proposal in each case. For this reason, the results reported in the literature on the levels of quality differ considerably between articles, making it difficult to compare them in the first instance.

Statistical Significance

In this section, a statistical analysis is addressed that allows a rigorous comparison of the precision obtained by each of the individual inference techniques, as well as the consensus of these techniques developed by GENECI, from a global point of view that considers all the results presented so far (except for the MELANOMA dataset).

According to Friedman's statistical ranking and Holm's non-parametric tests [232] performed for both AUROC and AUPR values (see Table 5.10 and 5.11 respectively), the best GENECI result is the one that obtains the first position in both cases (thus acting as a control denoted with *).

After this, it can be seen that the median of GENECI and the techniques derived from GENIE3 are also in good positions, and no statistical difference in their performance can be assured. The rest of the techniques show statistically lower performances since, in their case, the null hypothesis of Holm's test is rejected.

GENIE3 obtained such good results during our experimentation due to the main use of the DREAM challenges as a data source. This algorithm shows a clear specialization of the time series exposed in these challenges, as it won several times. This is part of what is discussed in the manuscript about the unidentified specialization of the different inference techniques, and that causes that in the absence of a gold standard, it is not possible to know a priori which is the best tool to infer the problem network. This is what GENECI tries to solve, i.e., it does not try to outperform all the individual techniques in their domains of specialization (since, by the No Free Lunch theorem itself, this would be impossible) but aims to obtain quality results (competitive concerning the best technique) for a wide range of problems, gaining generalization capacity and relieving the researcher of the need to choose and rely on an individual technique.

On the other hand, it is worth mentioning that GENECI, as well as boosting the weight of the most promising techniques for each data set, has shown that it can quickly silence those that are not. The presence of noise from the less competitive techniques has not been detected, and GENECI has not hesitated to assign low weights to them in those cases where it has been considered pertinent.

All this assures that the GENECI proposal obtains a highly competitive performance and ensures robustness for its use in the inference of real-world datasets for which no previous solutions are known.

It is worth noting that the lower performance of GENECI on very small networks, observed in some of the DREAM challenge benchmarks, does not repre-

Table 5.10: Friedman mean rank with Holm's adjusted p values (0.05) for AUROC.

AUROC		
Algorithm	$Friedman'sRank$	$Holm'sAdj - p$
***Best GENECI**	**3.100e+00**	-
GENIE3_RF	3.833e+00	0.742e+00
GENIE3_ET	4.067e+00	0.742e+00
Median GENECI	4.367e+00	0.723e+00
GENIE3_GBM	5.617e+00	0.079e+00
MRNETB	7.417e+00	3.215e-04
CLR	7.467e+00	3.170e-04
MRNET	8.083e+00	2.769e-05
TIGRESS	9.117e+00	2.034e-07
PCIT	1.002e+01	1.518e-09
ARACNE	1.002e+01	1.518e-09
BC3NET	1.033e+01	2.344e-10
C3NET	1.052e+01	7.896e-11
KBOOST	1.105e+01	2.386e-12

sent a limitation in real-world applications. These instances are artificial problems that attempt to reproduce regulatory dynamics at a reduced scale, while real-world biological networks are typically much larger and more complex. Since the second aggregate term of the fitness function was designed to reflect topological features of realistic networks, its behavior was expected to be less effective in such reduced artificial settings. This effect is not problematic in practice, as the algorithm is primarily intended for large-scale gene regulatory networks, where it has demonstrated competitive and robust performance.

5.3.2 Real-World: MELANOMA

Finally, GENECI has also experimented on non-simulated gene expression data. Specifically, real-world clinical data from melanoma patients are used (see 4.1.2).

Since, in this case, there is no gold standard to validate the accuracy of the consensus network, a review of the current literature will proceed to manually check the existence of the most relevant inferred relationships in studies related to melanoma and immunology. Figure 5.8 shows the results of the different individual techniques and the GENECI consensus gene network at the top left. The three relationships that obtained the highest level of confidence after consensus are (1) IL1R2 - ARG1, (2) IL18R1 - IL1RL1 and (3) HLA-DQA1 - HLA-DQB1.

Table 5.11: Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.

AUPR		
Algorithm	$Friedman'sRank$	$Holm'sAdj - p$
***Best GENECI**	**2.867e+00**	**-**
GENIE3_RF	3.783e+00	0.412e+00
Median `GENECI`	4.233e+00	0.412e+00
GENIE3_GBM	5.200e+00	0.092e+00
GENIE3_ET	5.433e+00	0.070e+00
CLR	7.333e+00	1.772e-04
MRNETB	7.517e+00	1.002e-04
MRNET	8.117e+00	8.194e-06
PCIT	8.350e+00	3.074e-06
ARACNE	9.250e+00	3.082e-08
C3NET	9.750e+00	1.857e-09
BC3NET	1.077e+01	2.853e-12
KBOOST	1.117e+01	1.846e-13
TIGRESS	1.123e+01	1.233e-13

Regarding the first interaction, several studies relate both genes within the context of cancer [233, 234]. Specifically, in [235] it is concluded that the amebiasis pathway could be involved in melanoma metastasis through these genes. Regarding the second connection, despite appearing in the literature as highly associated with allergic pathologies such as asthma or dermatitis [236, 237, 238], in [239] both genes are related to repressors involved in epithelial cancers. Finally, for the third interaction, in addition to numerous articles that relate the HLA antigen family to this type of cancer [240, 241], in some of them, this relationship becomes the protagonist, and the central axis of the study [242].

5.3.3 Time Complexity

Regarding the algorithm's asymptotic time complexity (Big-O), an implementation-informed approximation is provided below:

- *Preprocessing:* running the k base inference methods on the dataset yields a total cost $C_{\text{base}} = \sum_{i=1}^{k} C_i$.

- *Evolutionary optimization (over E evaluations):* in each evaluation, (A) consensus construction from the k method outputs takes $O(m \cdot k)$, and (B) objectives compute as follows: *Quality* in $O(m \cdot k + m)$; *Degree distribution* in

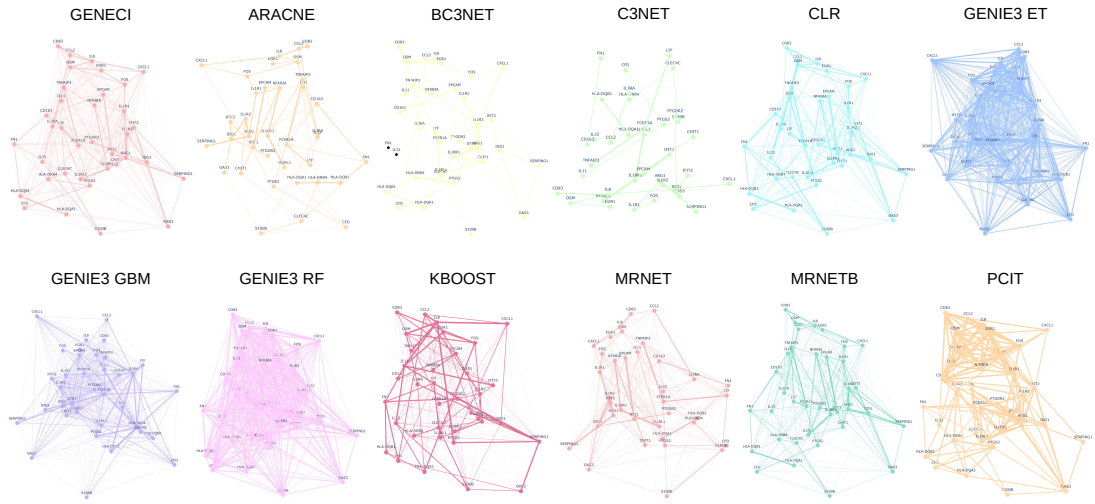


Figure 5.8: For Melanoma expression data, gene networks inferred by multiple individual techniques as well as the consensus one produced by `GENECI` are represented. Graphs attempt to represent gene regulatory networks by setting up genes through nodes and interactions through links. In this case, as they are captures of interactive representations, the directionality of the interactions is not visible to the naked eye. However, the equal arrangement of nodes allows the topology of the different networks to be easily compared.

$O(m + n \log n)$. Summing the per-evaluation costs:

$$O(m \cdot k) + O(m \cdot k + m) + O(m + n \log n) \;=\; O(m \cdot k + m + n \log n).$$

- *Overall:*

$$\boxed{C_{\text{base}} \;+\; C_{\text{GENECI}}} \;\approx\; C_{\text{base}} + O\Big(E\,(m \cdot k + n \log n)\Big).$$

- *Symbols:* n = number of genes (nodes); m = number of candidate directed interactions (up to $n(n-1)$); k = number of base methods; P = population size; T = number of generations; E = total fitness evaluations (typically $E \approx T \cdot P$); C_{base} = total cost of running all base methods once.

Chapter 6

Memetic Inference: Incorporating prior knowledge into the evolutionary process

In the biomedical field, the adoption of memetic algorithms has gained significant traction, demonstrating their versatility and effectiveness in several applications [243, 244, 245, 246, 247, 248]. These algorithms, which combine intensive local search with global evolutionary strategies, have been successfully applied to solve complex problems in this domain. For instance, in [245] the optimization of PPI (Protein-Protein Interactions) network alignment considers both topological structure and sequence similarities, surpassing existing methods in accuracy. Additionally, in protein structure prediction, memetic algorithms have been designed using knowledge from databases to guide the search towards similar native structures, showing promising results comparable to reference prediction methods [244, 246]. In the field of cancer diagnosis, the application of memetic algorithms has demonstrated to enhance the selection of relevant genes by combining local and global search techniques to identify discriminant genes with precision [243].

Finally, the memetic approach has also reached the focus of this Tesis, the reconstruction of GRNs. In [247], an innovative approach is proposed to learn parameters of Recurrent Neural Networks (RNN) and develop an LASSO (Least Absolute Shrinkage and Selection Operator) based framework for the effective reconstruction of GRNs. This method demonstrates superior ability to handle the complexity and sparsity of relationships in real GRNs, outperforming other RNN learning algorithms in large-scale network reconstruction. More recently, in [248], a memetic algorithm is proposed for inferring sparse GRNs using Max-

imum Entropy Probability Models (MEPMs). This approach addresses the problem from a multi-objective optimization perspective, considering maximum entropy and MEPM constraints as separate objectives.

Given the statistical rigor demonstrated by the GENECI proposal in its results (see chapter 5) and considering the validity that the memetic approach has shown in biomedical domain problems, it is more than justified to introduce this approach to address the specific problem of reaching a consensus among several inference techniques for the reconstruction of GRNs.

6.1 Proposed Approach

In this chapter, a memetic algorithm is proposed to optimize the consensus of different techniques for the inference of gene regulation networks. This is based on the previous proposal where an evolutionary process drives this optimization based on the quality and topological characteristics of the networks (see chapter 5). This tool has been complemented with a local search phase to guide the optimization process, thanks to prior knowledge of certain gene interactions in the network. This additional phase is located and exemplified in Figure 6.1. For a more technical analysis, the pseudocode is set out in Algorithm 4.

The set of candidates subjected to local search is iteratively explored in a loop spanning the length of the individual (line 3 in Algorithm 4). This set comprises the individual provided by the previous phase without any modification (case $i = -1$ in Algorithm 4) and each of the variations resulting from granting an additional vote to each technique (case $i \neq -1$ in Algorithm 4). In other words, the first variation will correspond to adding an additional vote of confidence to the first technique, quantified as the value of one vote in the case that the system is not weighted (case $i = 0$ and line 6 in Algorithm 4). The exact formula for calculating the new value of the technique in the vector is explained and exemplified in Figure 6.1.

After generating the candidates, they are repaired and the consensus network derived from each of them is constructed (lines 7 and 8 in Algorithm 4). Finally, the distance of their confidence levels from the known interactions in the network is measured (line 9 in Algorithm 4). The known interactions will usually be assigned a confidence level equal to 1 in the comparison file. However, if the medical researcher wishes to assign a certain probability to their knowledge, any other value between 0 and 1 is accepted. This means that knowledge of a non-existent interaction could also be reflected, but this case is less common.

If the distance is less than the recorded minimum, the current one becomes

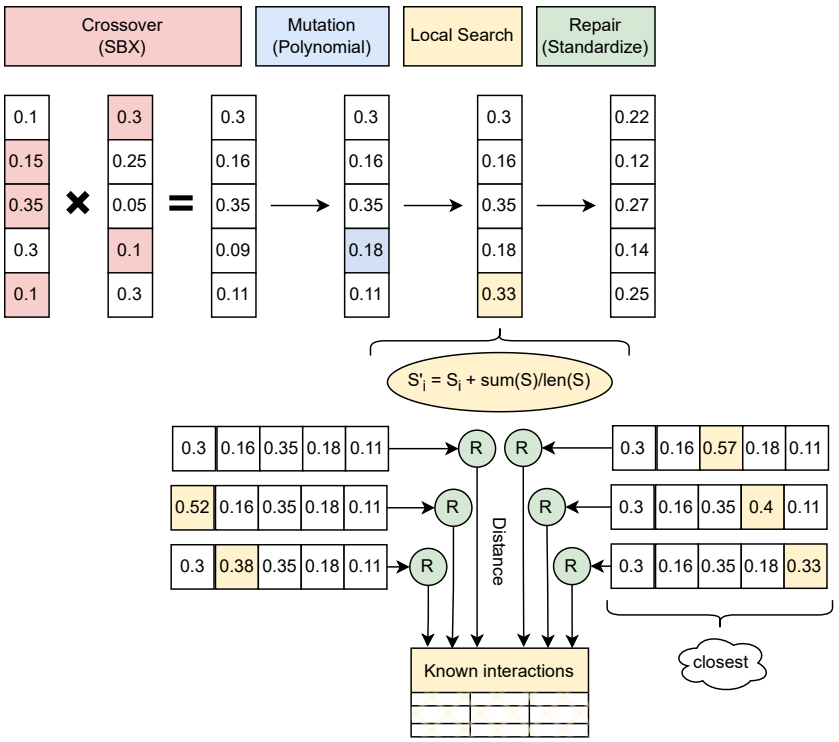


Figure 6.1: Succession of phases within the evolutionary process. Individuals are crossed through simulated binary crossover and subsequently subjected to polynomial mutation. Following this, the local search begins where several variations of the individual (encoding a given solution) are compared to select the one whose consensus network is closest to the known interactions. Finally, the individuals are repaired to resume their representation in the form of a weight vector.

the new minimum and the best solution is replaced by the current one (lines 10-12 in Algorithm 4). At the end of the loop, the solution with the smallest distance to the reference is returned (line 13 in Algorithm 4).

The distance is calculated as a simple summation of the absolute value differences between the value of the known interactions (usually 1) and the confidence levels assigned by the consensus network for these interactions. However, the possibility that the set of known interactions is a poorly distributed sample that always favors the same technique during the consensus, has been considered. To mitigate this possibility, an additional parameter has been added that defines the interactions that participate in the calculation of the distance on each iteration.

This parameter is exemplified in Figure 6.2 by covering its three possible values, namely: the option *all* is contemplated, in which all the known interactions

Algorithm 4 Main code of the local search phase

Require: Individual sol, Known interactions involved in distance calculation ref.
Ensure: Improved individual $resSol$.

```
1: resSol ← copyOf(sol)
2: minDistance ← inf
3: for i in (−1, len(sol)) do
4:     tmpSol ← copyOf(sol)
5:     if i ≠ −1 then
6:         tmpSol[i] += sum(sol)/len(sol)
7:     end if
8:     RepairSolution(tmpSol)
9:     net ← GetNetwork(tmpSol)
10:    distance ← Distance(net, ref)
11:    if distance < minDistance then
12:        minDistance ← distance
13:        resSol ← tmpSol
14:    end if
15: end for
16: return resSol
```

participate in all local searches; the option *some* in which a randomly chosen subset of them participates on each occasion; and finally the option *one* in which only one of the known interactions chosen randomly is used on each local search.

This local search phase aims at breaking the limitations imposed by `GENECI` in its aggregate term *Quality*, where techniques whose confidence levels are quite consistent with the remaining ones are somewhat rewarded. Although the consistency of confidence values can increase the reliability of a technique, this strategy sometimes lets certain peculiar interactions that are only inferred by a small subset of techniques slip away. The local search allows for the utilization of prior information to the inference of the network to identify these cases and redirect the evolution of the individuals. It is evident that both strategies are interdependent and must coexist in the evolutionary process, as exceeding the use of previously known information could provoke overfitting.

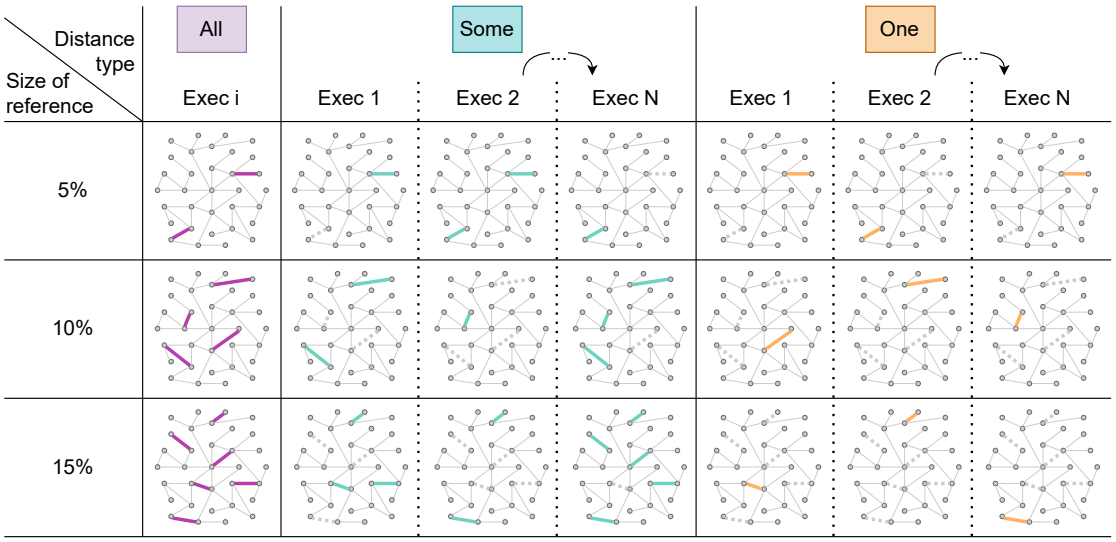


Figure 6.2: Examples of interactions involved in the distance calculation in different executions based on the proportion of the gold standard extracted as a set of "known by the expert" interactions (rows) and the type of distance (columns). The case of extracting 5%, 10%, and 15% of the gold standard for the distance types *all*, *some*, and *one* respectively, is shown. All executions take the same reference in the case of *all*, while for *some* and *one*, there is a certain random component that causes differences on each local search.

6.2 Experimentation

The experimentation addressed in this study employs the academic benchmarks provided by the DREAM challenges [142] (specifically their 3rd and 4th editions) and the yeast network of IRMA [173] (see section 4.1). All these networks were also part of the experimentation of `GENECI` and constitute a total of 27 inference cases. The known interactions of these networks that will guide the evolutionary process have been defined from their gold standards (known solutions information). Specifically, 5% of these references have been extracted for each execution.

The accuracy of the results will be calculated using the AUROC and AUPR metrics, which were set by the DREAM challenges themselves for their competition and make it possible to compare these results with other studies in the literature. Other metrics such as F1-Score and MCC are not considered, as the use of the chosen benchmark standards is deemed sufficient to cover this study.

6.2.1 Parameter settings

Given that this proposal partially follows the evolutionary process of GENECI, which is in fact common in standard EA settings, it has been decided to keep as much as possible the parameter setting that was configured in the experimentation of its corresponding chapter, hence allowing a fair comparison. Therefore, the default settings of simulated binary crossover (with a probability of 0.9), polynomial mutation (with a probability of $1/n$, where n is the number of techniques to be consolidated), and repair based on vector standardization have been established. However, for the additional phase proposed in this chapter, it remains to determine the probability with which the local search is carried out (which is independent of the crossover and mutation probability) and the way the information from the known interactions is used for the calculation of the distance.

To find the most suitable values for these two parameters, all possible combinations between their values have been considered. For the probability of the local search, the candidate values 0.1, 0.25, 0.4, and 0.55, have been defined. And for the type of distance, the already discussed options of *all*, *some*, and *one*.

Each combination of parameters has been tested with 15 independent executions for each network considered in this study. Afterwards, the performance of each solution was calculated using the AUROC and AUPR metrics with regard to the gold standards. For each network and combination of parameters, the median of their precision values was extracted, which finally allowed the calculation of a Friedman statistical ranking with Holm's non-parametric tests.

The results are shown in Table 6.1 for the AUPR metric and in Table 6.2 for the AUROC metric. It can be seen how the winning combination for both cases is the one that always takes into account all the known interactions in the distance calculation and with a higher probability of local search. That is, the combination that employs to a greater extent the external information provided. However, rigorous statistical significance cannot be attributed to this victory since only in one case does it meet the established threshold of $p < 0.05$.

A point to consider regarding the lack of statistical significance is that academic problems have a relatively small network size, sometimes around 10 nodes. This causes the difference between taking all or only a subset of interactions for distance calculation to rely on a couple of interactions, which does not allow for a significant statistical conclusion. However, there is an observable trend towards providing more accurate solutions when the available information is maximized simultaneously through probability and the method of distance calculation.

Table 6.1: Friedman mean rank with Holm's adjusted p values (0.05) for AUPR. Several distance (D) and local search probability (P) configurations are compared based on the AUPR metric. For this purpose, 15 independent runs of each configuration were performed and the median of them (Median) was rescued. After running Friedman's statistical ranking (second column), the winner (highlighted in bold with $*$) is taken as a reference to measure statistical significance against the rest using Holm's nonparametric tests (third column).

AUPR		
Algorithm	$Friedman'sRank$	$Holm'sAdj - p$
***Median D-all P-0.55**	**4.88889**	-
Median D-one P-0.25	5.90741	0.725979
Median D-one P-0.1	5.96296	0.725979
Median D-all P-0.25	6.03704	0.725979
Median D-some P-0.4	6.24074	0.673303
Median D-all P-0.4	6.53704	0.465230
Median D-some P-0.1	6.62963	0.456477
Median D-one P-0.55	6.75926	0.396552
Median D-some P-0.25	6.90741	0.341178
Median D-one P-0.4	6.92593	0.341178
Median D-some P-0.55	7.00000	0.314504
Median D-all P-0.1	8.20370	0.008033

Table 6.2: Friedman mean rank with Holm's adjusted p values (0.05) for AUROC. The procedure and nomenclature are identical to those in Table 6.1.

AUROC		
Algorithm	$Friedman'sRank$	$Holm'sAdj - p$
***Median D-all P-0.55**	**5.53704**	-
Median D-some P-0.1	6.24074	1.99405
Median D-one P-0.25	6.42593	1.99405
Median D-one P-0.4	6.42593	1.99405
Median D-all P-0.25	6.46296	1.99405
Median D-one P-0.55	6.46296	1.99405
Median D-one P-0.1	6.59259	1.99405
Median D-all P-0.1	6.61111	1.99405
Median D-all P-0.4	6.74074	1.99405
Median D-some P-0.4	6.77778	1.99405
Median D-some P-0.25	6.79630	1.99405
Median D-some P-0.55	6.92593	1.72664

Regarding the other combinations, another factor that cannot be measured and may have affected the results should be taken into account, granting better precision to combinations with less use of information and worsening the results of others that made greater use of it. In each execution, to form the set of known interactions, a random 5% of the network's gold standard was extracted. Although the number of reference interactions was the same in all executions, their informational value is not necessarily equivalent. That is, the knowledge about the existence of certain interactions may be more valuable than that of others. This is an unpredictable and inevitable fact, since eliminating randomness and establishing fixed reference relationships could bias the results even more.

In the context of the academic networks employed in this study, it is logical to consider extending the winning combination and adding a higher probability of local search to further improve precision levels. However, it should be noted that in such academic problems, the temporal expression levels are simulated from a predefined set of interactions, which ultimately represents the gold standard of the problem. This means that whenever known interactions are added from this gold standard, information from the optimal solution is being shared. This is not the case with real-world networks, and even less so with networks that are intended to be inferred (e.g. in vivo experiments that are not performed yet). In other words, in the cases for which this proposal is intended, the information provided could form part of a good solution known to the domain expert, i.e. a set of interactions that effectively provides a logical explanation of what happens to the gene expression levels during the experiment. However, this may not be the only possible explanation, and there may be other similar alternatives that fit the scenario better. If such information is consistently favored with high probability, it could disturb the direction in which the population evolves during the algorithm execution. Nevertheless, keeping these interactions in mind regularly can bring the population closer to a high-potential zone without condemning the evolution to a possible local minimum.

Given that the optimal solution for these real-world networks intended to be inferred is unknown, the deviation that can be caused by overusing local search could be critical. Therefore, in this case, the most intelligent stance is caution rather than blindly parameterizing in full this proposal based on simulated problems without this broader perspective.

Furthermore, even in academic data where the information injected into the local search is part of the optimal solution to the problem, there is a certain risk that a poorly distributed sample of known interactions may end up diverting the evolution of individuals. The deterioration that these cases can cause to the

accuracy of the results increases with the probability of local search. Therefore, once again, setting certain limits is a good practice to maintain a balance that ensures the proposal's security.

Therefore, despite the lack of rigorous statistical significance, the combination of distance *all* and probability 0.55 is chosen as the winner, as it has obtained the first position in the ranking for both precision metrics.

6.3 Results and discussion

After configuring the parameters of the memetic algorithm, this section quantifies the improvement achieved by this proposal after adding the additional phase of local search. To this end, the precision results presented in the GENECI chapter (see chapter 5) are compared with those obtained by the best parameter combination seen in the previous section. Specifically, for each network and precision metric, the median of GENECI's executions is compared with the median of the executions of the current proposal. This comparison has been decided to be represented visually for editions 3 and 4 of the DREAM challenges (see Figs. 6.3 and 6.4 respectively) and presented quantitatively in Table 6.3 for the IRMA yeast network.

Figure 6.3 shows that the median accuracies of the solutions from the approach in this chapter surpass, in most cases, the accuracies provided by the original version of GENECI. Upon closer examination, it is noticed that there is a certain relationship between the size of the networks and the stability of this improvement. That is, for larger networks, the enhancement provided by the additional phase of this approach is more robust and decisive. However, in the case of small networks, more varied differences are observed between the two algorithms, with ties or even a slight lead of the original version appearing in certain cases. This, in a way, validates the choice of the application domain selected for this proposal which, despite being tested on simulated networks, is intended for inferring real-world networks with significantly larger sizes.

Regarding the instability observed for small-sized networks, it is worth mentioning that these cases have a higher probability of obtaining a poorly distributed sample, as the samples have very few interactions and therefore a good representation is not achieved in any case. Therefore, the instability observed in these cases confirms what was previously mentioned in the parameterization, as even with the introduction of correct interactions, a bad sample can divert the proper evolution of the population. However, thanks to the caution and balance achieved in the parameterization, the impact of these exceptional and

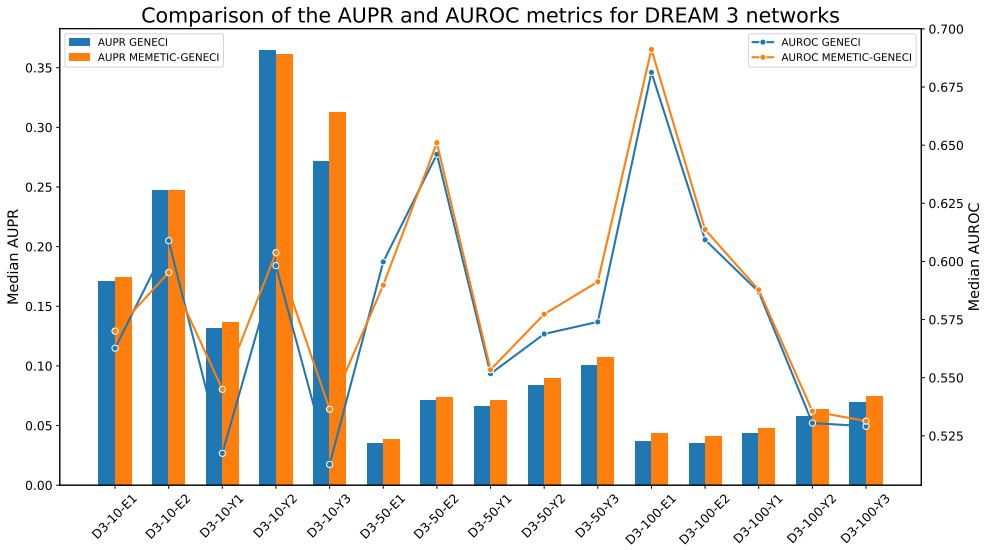


Figure 6.3: Comparison of the AUROC and AUPR performance metrics for the `GENECI` (in blue) and `Memetic Inference` (in orange) algorithmic proposals on each of the networks belonging to the third edition of the dream challenges (horizontal axis). For identification, the challenge prefix (D3) is followed by the size of the network (10, 50 or 100) and finally the initial of the organism on which it is based (Y: Yeast, E: E. coli). The bars indicate the medians of the AUPR values and the lines with markers represent the medians of the AUROC values for each network. The AUPR and AUROC values are displayed on separate vertical axes due to their different measurement scales, reserving the left axis for AUPR and the right axis for AUROC.

indetectable cases a priori is quite moderate on the accuracy of the solutions. It is possible to guide and influence the evolution of individuals without completely damaging their convergence.

In Figure 6.4, the precision levels of both proposals for networks from DREAM 4 are compared. In this plot, the connection between the size of the networks and the stability of the improvement provided by the local search phase is once again confirmed. Additionally, in this subset of networks, the correlation between both metrics is observed in greater detail. That is, both metrics seem to simultaneously show the same degree of improvement in most cases. This adds a certain reliability to the proposal of this chapter.

Finally, in Table 6.3, the precision levels for the yeast network of IRMA are presented. In this case, given that it is such a small network with such a high initial precision level, the margin for improvement is minimal. Additionally, the information available in the set of known interactions is extremely limited, around

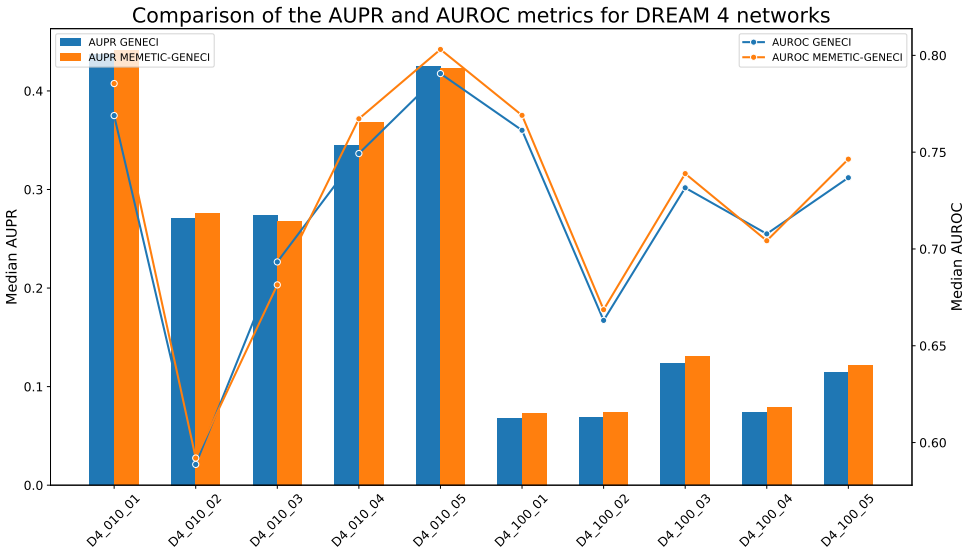


Figure 6.4: Comparison of the AUROC and AUPR performance metrics for the `GENECI` (in blue) and `Memetic Inference` (in orange) algorithmic proposals on each of the networks belonging to the fourth edition of the dream challenges (horizontal axis). The nomenclature and interpretation of the graph are identical to those in Figure 6.3.

Table 6.3: Accuracy values for IRMA networks. In this table, a gene network is contemplated for each pair of columns, where in each row the AUPR and AUROC values are provided for each algorithm.

Técnica	**IRMA_switch-off**		**IRMA_switch-on**	
	AUROC	**AUPR**	**AUROC**	**AUPR**
Median `GENECI`	0.8611	0.7865	0.8889	0.75
Median `Memetic Inference`	0.8611	0.7865	0.8939	0.7549

1 interaction (the minimum allowed). Nevertheless, a subtle improvement has been achieved in the "switch-on" version, maintaining exactly identical values for the "switch-off" instance. The fact that identical values are obtained is due to the small size of the network, causing precision values to be quite staggered.

After analyzing all the sets of networks, it can be checked how the memetic proposal surpasses `GENECI` in the majority of cases. To provide greater rigor to this comparison, the Wilcoxon test has been calculated, which has provided a p-value of 2.468690e-03 for AUROC and 1.592934e-05 for AUPR. That is, the improvement in the precision of the results is statistically significant.

The ability to achieve statistically significant improvements with such a restricted sample of known interactions (5% of the gold standard) highlights the algorithm's efficacy in integrating and maximizing the informational value of a limited data set. This is especially crucial in the field of computational biology, where the complete and accurate availability of data can be a constant challenge.

It is worthy to note that thanks to the precautions taken during parameterization, this proposal has demonstrated robustness and reliability. During the experimentation, the subset of interactions designated to form the reference in the local search phase was chosen randomly. This random choice has led to the emergence of poorly distributed samples that could disturb the optimization of the population. However, it has been shown that the impact on the deterioration of accuracy has been minimal in these exceptional cases.

Furthermore, it is also important to comment that this proposal has managed to improve results in a set of extensively worked and studied benchmarking networks, whose margin for improvement was initially very limited. The algorithm's ability to find and exploit areas for improvement in these networks indicates its potential to inject the knowledge provided by the expert and maximize its use to discover novel insights in the data.

Chapter 7

MO-GENECI: Multi-objective consensus guided by biological context

In this chapter, MO-GENECI (Multi-Objective GEne NEtwork Consensus Inference) is introduced, presenting a novel approach with a multi-objective focus to guide the optimization of consensus inference from an extensive pool of inference techniques within the context of gene regulatory networks. It builds upon the groundwork laid out in GENECI (see chapter 5) and enhances its implementation through:

- Implementation of a multi-objective approach involving new fitness functions. The individual aggregate terms of GENECI have been thoroughly analyzed, a deeper exploration has been conducted within the context of real biological networks, and a more rigorous procedure has been designed to enhance decision-making during the implementation of the functions (comparison of versions). First, a new *Quality* function has been implemented and improved through exhaustive comparisons of modifications. Second, the *Topology* function has been renamed as *Degree distribution* and refined. Thus, the *Topology* now dispenses with the binarization of the consensus network and has incorporated a goodness-of-fit test to a Pareto distribution directly applied to the confidence levels of interactions. Additionally, a third objective named *Motifs* has been implemented from scratch, based on detecting frequently occurring patterns in real gene regulation networks.
- New crossover and mutation operators specific to the addressed problem

have been implemented, ensuring the feasibility of solutions after execution to eliminate the search space distortion caused by the use of repair operators.

- Replacement of the genetic algorithm using a more suitable and robust multi-objective model based on NSGA-II [79].
- The number of available techniques for the initial inference of gene regulation networks has been expanded. In the initial version, ten techniques were considered, whereas now, a total of 26 techniques are used. Furthermore, to minimize the execution time of this phase as much as possible, not only has the parallel execution of containers been maintained, but a function has been implemented to automatically and intelligently distribute the available cores, allocating more resources to computationally expensive parallelizable techniques.
- A parameter setting and analysis procedure has been designed using a larger dataset, thanks to the addition of new known gene networks and the integration of the SysGenSIM simulator [179]. This collection forms an experimental benchmark comprising 106 gene regulation networks sourced from diverse origins, ensuring comprehensive coverage across all specialization domains in this study.

7.1 Algorithmic Proposal

MO-GENECI represents a groundbreaking approach that combines multiple state-of-the-art inference techniques, harnessing their collective power to optimize the confidence levels associated with each interaction. This optimization is accomplished through a cutting-edge multi-objective evolutionary algorithm, carefully designed with customized operators and fitness functions directly aligned with the specific biological context under investigation.

The main workflow is outlined in detail in Figure 7.1. This graphical representation illustrates the procedure to be followed, starting from the initial expression data analysis to the exportation of consensus networks. The Python package developed (available on PyPi [1]) includes complementary functions that cover all these steps, including individual GRN inference, which can be performed using 26 different inference techniques within the same working environment. However, the key and truly innovative phase is the "optimizing consensus" stage, where the specifically designed multi-objective evolutionary algo-

[1] `https://pypi.org/project/geneci/`

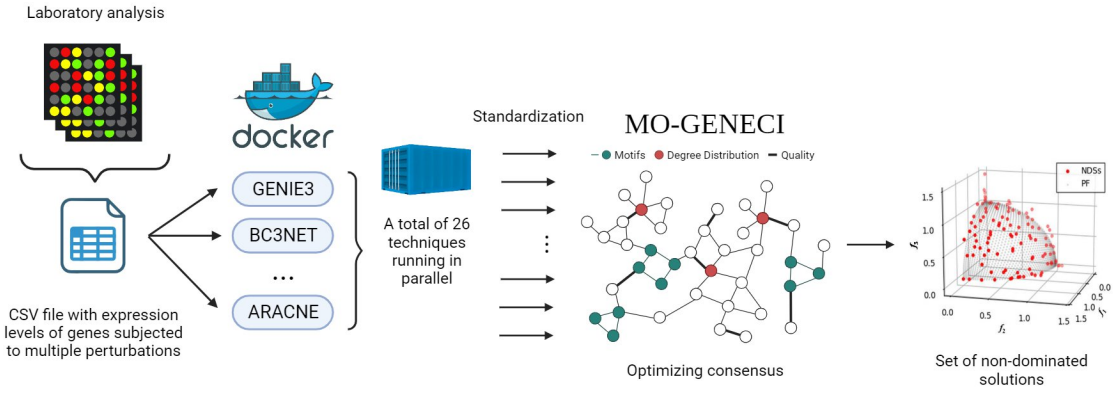


Figure 7.1: Workflow Implemented in `MO-GENECI`. After providing the input expression data, the inference techniques are executed in parallel thanks to their encapsulation in Docker containers. Once the results of all techniques are obtained, the lists containing confidence values for each interaction are handed over to the multi-objective evolutionary algorithm. This algorithm generates an initial random population (weight vectors) and undergoes the iterative process of evaluation (*Quality*, *Degree distribution*, *Motifs*), selection, crossover, and mutation until the maximum number of iterations is reached. Upon completion of the multi-objective evolutionary algorithm, the result consists of the set of non-dominated solutions from the last generation [249].

rithm comes into play.

In this sense, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [79] provides the structural support for `MO-GENECI`. This skeleton, which defines the main stages of the algorithm, is customized through the choice of a specific representation of a weighted voting system, the incorporation of carefully designed crossover and mutation operators, and the design of several fitness functions tailored to address this bioinformatic problem from the biological context to which it belongs. Figure 7.2 shows a flow chart with the main phases of the designed algorithm. In the following sections, the logic implemented in each of these phases will be explained in greater detail.

NSGA-II is an evolutionary algorithm designed to solve multi-objective optimization problems. It is based on the principles of genetic algorithms and is characterized by three main features that are quite beneficial for the problem addressed:

- **Elitism**: The best solutions from the current population are carried over to the next generation. This ensures that the quality of solutions does not degrade over time.
- **Diversity Preservation**: NSGA-II uses a mechanism known as "crowding

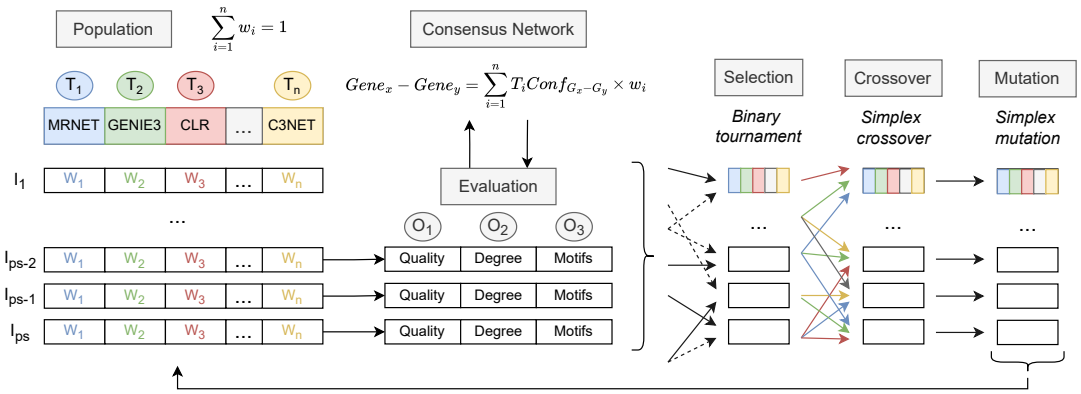


Figure 7.2: Flowchart of the multi-objective evolutionary algorithm developed in this proposal. The algorithm starts with the generation of an initial population, whose representation is in the form of a vector of weights (simplex). The individuals are evaluated for each of the fitness functions, previously applying the conversion of the individual to its corresponding consensus network. This is followed by a selection process guided by a binary tournament. Finally, the individuals are subjected to the crossover and mutation operators selected for the representation of this problem, giving rise to the next generation of individuals.

distance" to maintain diversity in the population. The crowding distance measures how close an individual solution is to its neighbors. Solutions with a larger crowding distance are preferred as they are less crowded.

- **Non-Dominated Sorting**: This is a method of ranking solutions based on their dominance. Solutions are sorted into different "fronts" based on the number of solutions that dominate them. The first front contains non-dominated solutions, the second front contains solutions dominated by the first front, and so on.

MO-GENECI follows the general outline of an NSGA-II, as depicted in Algorithm 5. In this pseudocode, the primary phases of the evolutionary algorithm are outlined. It is worth noting that some of these phases, such as the representation, fitness functions, and crossover and mutation operators, have been designed from scratch for the specific case under consideration in this study. Therefore, they will be further elaborated upon in subsequent sections to provide a comprehensive understanding of the approach.

Initially, a random population is created (line 1 in Algorithm 5), which is then evaluated using several fitness functions designed in this study (line 2 in Algorithm 5). Following this, an iterative process begins and continues until the maximum number of generations is reached (lines 3, 4, and 12 in Algo-

Algorithm 5 MO-GENECI Algorithm.

Require: Num of generations T, Population size P, Num of objectives M, Fitness functions $F : f_1, f_2, \ldots, f_M$, Crossover operator x, Mutation operator m
Ensure: Pareto-optimal front PF

1: $P \leftarrow$ generate_random_population(P)
2: $E \leftarrow$ evaluate_population(P, F)
3: $t \leftarrow 1$
4: **while** $t < T$ **do**
5: $ranks \leftarrow$ rank_population(E)
6: $crowd\_dist \leftarrow$ crowding_distance$(E, ranks)$
7: $selected \leftarrow$ select_population$(P, ranks, crowd_dist)$
8: $offspring \leftarrow$ crossover$(selected, x)$ ▷ Section 7.1.2
9: $offspring \leftarrow$ mutate$(offspring, m)$ ▷ Section 7.1.3
10: $P \leftarrow$ replace_population$(P, offspring)$
11: $E \leftarrow$ evaluate_population(P, F) ▷ Section 7.1.4
12: $t \leftarrow t + 1$
13: **end while**
14: $PF \leftarrow$ get_pareto_front(E)
15: **return** PF

rithm 5). During this process, solutions in the population are classified based on non-dominance and assigned a rank and crowding distance (lines 5 and 6 in Algorithm 5). The crowding distance is used to maintain diversity, especially when splitting a front. The `MO-GENECI` selection process consists of two steps: first, individuals are selected based on their rank, and if ranks are equal, crowding distance is used. This approach is known as crowding distance tournament selection (line 7 in Algorithm 5).

After selection, specific algorithmic operators for adapted crossover and mutation are applied to generate offspring (lines 8 and 9 in Algorithm 5). Subsequently, parent and offspring populations are merged (line 10 in Algorithm 5), forming the next generation. This process repeats in a loop or, in the case of the final generation, the result becomes the Pareto front obtained from the algorithm (line 14 in Algorithm 5).

As the algorithmic skeleton of `MO-GENECI`, NSGA-II has been found to be effective in finding a better spread of solutions and better convergence near the true Pareto-optimal front compared to other multi-objective evolutionary algorithms [79]. Its advantages include its ability to solve multi-objective optimization problems, its elitism (which increases the convergence speed), and its parameter-less sharing approach.

However, despite all these advantages, other options were considered and ultimately discarded. Firstly, MOEA/D [81] does not perform well with non-normalized fitness functions, which does not fit in this case considering the *Motifs* fitness function explained below. Secondly, OMOPSO [250] has its own mutation operator and does not maintain solution feasibility. Thirdly, GDE3 [251] is designed to use differential evolution crossover, which also pushes individuals out of the search space. Finally, while compatible, SMPSO [252] does not take advantage of the specific problem-based design described below due to its lack of a crossover phase.

7.1.1 Solution Representation

For the representation of individuals, weight vectors are employed, where each position specifies the weight assigned to a particular technique in the voting system. Therefore, the sum of all positions in the vector must always be equal to 1. This greatly influenced the implementation of the crossover and mutation operators in the evolutionary algorithm, which are deeply detailed in subsections 7.1.2 and 7.1.3, respectively.

7.1.2 Crossover

As mentioned earlier, it has been decided to implement a crossover operator that generates new solutions within feasible regions in the search space. Therefore, the chosen crossover operator is the *Simplex Crossover* [253], whose Java implementation has been adapted from the MOEA framework [254].

The *Simplex Crossover* is a multi-parent recombination operator proposed for real-coded genetic algorithms. This operator generates offspring by uniformly sampling values from the simplex formed by m ($2 \leq m \leq len\_vector + 1$) parental vectors. Experimental results with commonly used benchmark functions in evolutionary algorithm studies [253] have shown that *Simplex Crossover* performs well on functions with multimodality and/or epistasis with a moderate number of parents: 3 parents for low-dimensional functions or 4 parents for high-dimensional functions. Both values have been considered in the parameter setting procedure to ensure the proper choice of this variable.

7.1.3 Mutation

Similar to the crossover phase, in the mutation phase, it was necessary to use an operator that allows the maintenance of solution feasibility after its execution. However, in this case, instead of adopting an operator from the literature, a new

Algorithm 6 Main code of Simplex Mutation operator.

Require: Individual from population ind, Mutation probability $mutProb$, Mutation strength $mutStr$
Ensure: Mutated individual ind

```
1: if randomDouble(0, 1) < mutProb then
2:     team1Size = randomInt(1, size(ind) - 1)
3:     team1 = randomSubset(ind, team1Size)
4:     team2Size = randomInt(1, size(ind) - team1Size)
5:     team2 = randomSubset(ind - team1, team2Size)
6:     amount = sum(team1) * mutStr
7:     mutateVariables(ind, norm(team1), -amount)
8:     mutateVariables(ind, norm(team2), +amount)
9: end if
10: return ind
```

one has been designed specifically for this problem. This operator is named the *Simplex Mutation*, and it is based on applying a negative perturbation to a subset of the solution in such a way that its subsequent positive adjustment, rather than being spread across the entire vector, is directed towards another specific subset.

The *Simplex Mutation* operator requires the specification of two parameters: mutation probability and mutation strength. The former is commonly used in such operators and determines the probability with which an individual in the population undergoes the mutation process. The latter refers to the magnitude of the perturbation applied to the vector, also ranging between 0 and 1.

The pseudocode for its implementation is shown in Algorithm 6. The mutation procedure begins by extracting two subsets from the solution vector (lines 2 to 5 in Algorithm 6). The first subset will undergo the negative perturbation, while the second one will dampen this impact by positively reinforcing its values. Both the size of the groups and the positions in the vector that compose them are random. The only constraint is that a position cannot belong to both groups simultaneously.

Then, the values of the positions belonging to the first group are summed up. This sum, multiplied by the second input parameter (mutation strength), determines the value to be subtracted from the first subset (line 6 in Algorithm 6). This value is not uniformly distributed across the subset; instead, each member has a percentage of this value subtracted based on the normalization of the subset. Therefore, each member of the group has the calculated value subtracted, multiplied by its normalized figure within the subset (line 7 in Algorithm 6).

After applying this negative perturbation, the amount subtracted from the first group is added to the second group following the same distribution procedure (line 8 in Algorithm 6).

An example is shown in Figure 7.3 to clarify this implementation. It offers a possible mutation of the individual (0.1, 0.05, 0.2, 0.1, 0.15, 0.1, 0.05, 0.15, 0.1), which, given its length, would belong to a run that tries to agree on 9 concrete inference techniques. The first step is forming groups whose sizes and members are entirely random. Team 1 (in red) will suffer a loss in their values, while Team 2 (in green) will increase their numbers to compensate for the previous loss. In this example, the mutation strength value is set to 0.2. The total amount to subtract in team 1 is the sum of its members multiplied by this factor, in this case: $0.2 * 0.4 = 0.08$. This amount is divided to subtract each member's share based on the team's normalization. Since the second member is half of the team's total weight, half of the total amount will be subtracted: $0.08 * 0.5 = 0.04$. After performing this operation on each member, the reverse process is applied to the members of team 2. That is, each member's portion of this amount is added to each member based on their normalization.

7.1.4 Evaluation

The evaluation process begins by transforming each individual into its corresponding consensus network proposal. In other words, for each interaction in each individual, the sum of the products between the confidence level reported by a technique and the weight assigned to it by the individual for the voting system is calculated. Afterwards, the individual and its corresponding consensus network are provided to each objective function for evaluation. `MO-GENECI` follows a 3-objective strategy, combining: *Quality*, *Degree distribution* and *Motifs*.

Objective 1: Quality

The *Quality* function aims to encourage the emergence of solutions whose consensus networks have a subgroup of interactions distinguished from the rest with high confidence levels that, in turn, originate from consistent weight distributions. In other words, it assigns greater importance to individual techniques whose reported values exhibit higher concordance than the remaining ones. The pseudocode for its implementation can be seen in Algorithm 7.

This function assigns a quality value to each interaction considered in the problem based on the solution being evaluated (lines 1 to 3 in Algorithm 7). Two critical values are considered for each interaction to make this assignment. The first one is the consensus confidence level, which is obtained by summing the

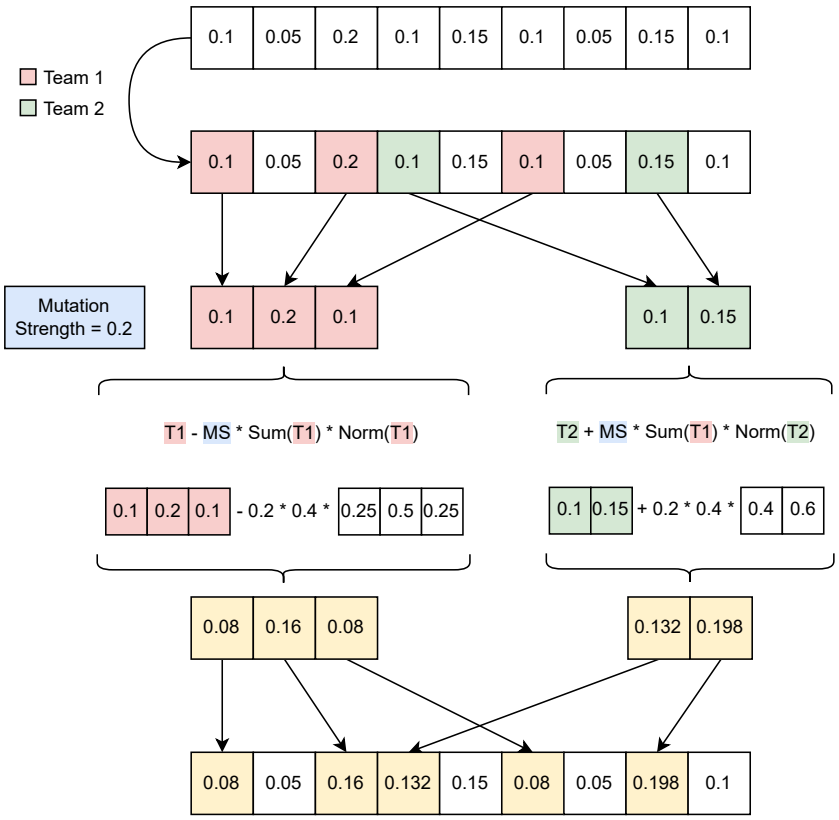


Figure 7.3: Example of mutation of an individual belonging to an execution trying to agree on 9 inference techniques. In red are highlighted the members of team 1 (intended to reduce their values), in green the members of team 2 (intended to increase their values), in blue the value of the mutation strength factor (set in this case to 0.2) and finally the values finally modified in yellow.

products of the confidence level given by each technique and its corresponding weight in the vector. The second value is called “distance” and is calculated as the difference between a specific vector’s maximum and minimum values for the interaction. For each technique, this vector stores the average between its weight and, the difference between the confidence value assigned to the interaction and the median of the confidence levels of all techniques for that same case.

Finally, the quality value assigned to an interaction is obtained by calculating the average between its consensus confidence level and the result of subtracting one from the distance value. In other words, an interaction will be considered of good quality when its confidence level is high, and its distance value is small. A small distance value indicates that the maximum and minimum of the calculated vector are close to each other, which implies that lower weights have been

Algorithm 7 First fitness function: Quality.

Require: Consensus list with confidence and distance values c
Ensure: Value of the fitness function $result$
1: $quality$ = []
2: **for** i **in** $len(c)$ **do**
3: $quality[i] = (conf_i + (1 - dist_i))\ /2$
4: **end for**
5: $qualitySum = 0$
6: $cnt = 0$
7: **for** i **in** $len(c)$ **do**
8: **if** $quality[i] > \ mean(quality)$ **then**
9: $qualitySum \mathrel{+}= quality[i]$
10: $cnt \mathrel{+}= 1$
11: **end if**
12: **end for**
13: $result = 1 - (qualitySum/cnt)$
14: **return** $result$

assigned to techniques with confidence values far from the median compared to other techniques. It also indicates that techniques with confidence levels close to the median (i.e., with a smaller distance) have been compensated with high weights.

Finally, for the evaluated individual, the *Quality* function selects those interactions whose quality surpasses the mean (lines 4 to 9 in Algorithm 7) and returns the unit subtracted by the mean of this subset (lines 10 and 11 in Algorithm 7). Other options were tested to arrive at this definitive version of the function, including:

1. Changing the way the vector from which the distance originates is calculated by replacing the median with the mean.

2. In the calculation of the final quality, interactions above the cutoff criterion or all interactions are selected instead of those above the mean.

3. An additional step that influenced the result by evaluating the number of interactions that entered the previous subset. This step was present in the initial proposal under the term “contrast”.

A temporary configuration was executed to make a decision regarding the definitive implementation of this objective in `MO-GENECI`. This involved replacing NSGA-II with a genetic algorithm while retaining the designed representation

and operators. The only change was in the evaluation process, now considered a single fitness function. In each execution, this fitness function would be the specific version of *Quality* from which its individual accuracy was desired. For each version, the algorithm was run on all networks considered in this study with a size of less than 1000 genes (see section 4.1). Subsequently, the accuracy of the results for each implementation was evaluated by comparing, for each network, the gold standard with the consensus network derived from the obtained solution.

This experimentation yielded an AUROC and AUPR value for each network and version. On the one hand, AUROC (Area Under the Receiver Operating Characteristic Curve) is a metric used to assess the discriminative ability of a classification model. It represents the model's ability to distinguish between positive and negative classes. On the other hand, AUPR (Area Under the Precision-Recall Curve) is another metric especially useful when classes are imbalanced, which is common in the inference of gene regulatory networks. It focuses on the true positive rate (recall) and the model's precision. Both metrics have been widely used to report on the efficacy of gene regulatory network inference techniques [9, 129, 132, 97, 50, 135, 98]

Finally, a statistical ranking of Friedman was computed using non-parametric Holm tests for both metrics [255]. Friedman's statistical ranking compares the performance of various proposals across multiple datasets. It calculates a test statistic based on the differences between the average ranks of the models across datasets. After calculating Friedman's statistical ranking, Holm tests are used to determine whether the winning version has significant differences in its performance compared to the rest. Holm tests control the Type I error by adjusting the p-values for each comparison, helping to prevent incorrect conclusions.

The result can be seen in Table 7.1 and Table 7.2. After observing the results, it was decided to choose the *Median Average* version as the definitive implementation of the *Quality* fitness function.

Objective 2: Degree distribution

The objective of *Degree distribution* is designed to favor solutions that lead to consensus networks with a distribution following a power-law. This characteristic of biological networks has been asserted in the literature [44, 45], and even the most straightforward implementation supporting this idea yielded excellent results in large networks (see chapter 5).

In a network with a degree distribution following a power-law, the frequency of nodes with a degree k is inversely related to k raised to a certain power. This

Table 7.1: Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.

AUPR		
Quality Version	$Friedman's Rank$	$Holm's Adj - p$
***Median Average**	**3.2093**	-
Mean Average	3.38953	0.62946
Mean Cut-off	4.23256	0.01231
Median Cut-off	4.44186	0.0029
Median	5.02907	5.534e-06
Mean Average Contrast	5.02907	5.534e-06
Mean	5.15698	1.109e-06
Median Average Contrast	5.51163	4.982e-09

Table 7.2: Friedman mean rank with Holm's adjusted p values (0.05) for AUROC.

AUROC		
Quality Version	$Friedman's Rank$	$Holm's Adj - p$
***Median Average**	**2.85465**	-
Mean Average	3.36628	0.170793
Mean Cut-off	4.61047	5.19242e-06
Median Cut-off	4.70349	2.23266e-06
Median	4.8314	4.84291e-07
Mean	5.02326	3.20887e-08
Mean Average Contrast	5.07558	1.65298e-08
Median Average Contrast	5.53488	5.0559e-12

leads to most nodes having few links while a few highly connected nodes, called "hubs", are present in the network. This idea can be visually appreciated with greater clarity in Figure 7.4.

Given that this concept is quite broad, an attempt was made to find a more precise distribution that fits within this framework and allows for a more refined procedure, such as a goodness-of-fit test. This is the case with the Pareto distribution, a continuous probability distribution that fits within the scale-free distribution and follows a power-law.

In the context of gene regulatory networks, this distribution aligns with the idea that most genes have relatively low connections. In contrast, a few genes have a very high number of interactions [44]. This organization of the gene regulatory network has significant implications for network dynamics and robustness. Highly connected hubs play a crucial role in signal propagation and information integration within the network [257, 258].

The implementation of this fitness function is reflected in Algorithm 8 and analyzes the consensus network calculated from the individual. For each node,

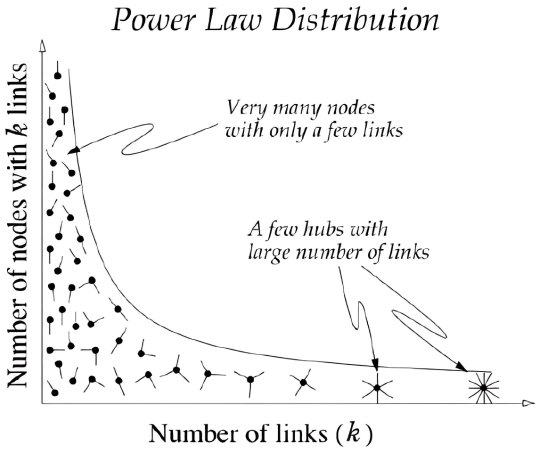


Figure 7.4: Visualization of a degree distribution following a power-law [256]. In this type of distribution, most nodes have only a few connections, while a small number of highly connected hubs concentrate a large fraction of the links. This characteristic 'long tail' pattern is frequently observed in biological networks, where scale-free organization enables robustness and modularity.

Algorithm 8 Second fitness function: Degree Distribution.

Require: Consensus list with confidence values c
Ensure: Value of the fitness function $result$
1: $degreeMap = \{String : Float\}$
2: **for** i **in** $len(c)$ **do**
3: $weight = c[i]$["weight"]
4: $degreeMap[c[i]$["source"]] += $weight$
5: $degreeMap[c[i]$["target"]] += $weight$
6: **end for**
7: $degreeArray = mapToSortedArray\,(degreeMap)$
8: $result = goodnessFitParetoTest\,(degreeArray)$
9: **return** $result$

it calculates its degree as the decimal sum of confidence levels for interactions, where the gene appears as a source or target (lines 1 to 5 in Algorithm 8). After storing the degree of all genes in a vector (line 6 in Algorithm 8), it is provided to the goodness-of-fit test for a Pareto distribution (line 7 in Algorithm 8, implemented based on [259]). It then returns a value related to the probability that the network follows this distribution.

Like with *Quality*, several versions of this function were considered. The first version (weighted), which has been described and is the definitive one, and a

Table 7.3: Friedman mean rank with Wilcoxon p values (0.05) for AUPR and AUROC.

	AUPR		AUROC	
Top. Version	*Friedman Rank*	*Wilcox p*	*Friedman Rank*	*Wilcox p*
***Weighted**	**1.4186**	-	**1.3256**	-
Binarized	1.5814	0.0935	1.6744	0.0015

second version (binarized) that, instead of using decimal confidence levels, applied a cutoff criterion to binarize the network. In this case, since there are only two versions to compare, a Wilcoxon test has been applied. The results obtained are presented in Table 7.3, and despite not being able to claim a statistically significant difference between the two versions for the AUPR metric, the first one from the ranking was ultimately chosen.

Objective 3: Motifs

In the field of systems biology, it is known that biological networks commonly exhibit a series of patterns known as motifs. These patterns are typically specific configurations of interactions between molecules, such as proteins, genes, or metabolites, that repeat in different parts of the network. Motifs are considered basic structural and functional units of biological networks, and their study helps understand principles of organization, modularity, and dynamics in complex systems. The detection and analysis of motifs provide valuable information about molecular interactions and signal propagation in a network, contributing to the understanding of the function and regulation of biological systems [46, 47, 260, 48].

The importance and well-established existence of these motifs in biological networks motivated the design of this third fitness function. The idea is to encourage the emergence of solutions whose consensus networks have a high density of motifs that have already been confirmed in previous studies as common in biological networks and gene regulatory networks.

The implementation of this fitness function is represented in the pseudocode presented in Algorithm 9. First of all, it is necessary to perform a prior binarization exercise (line 1 in Algorithm 9) to detect these motifs in the consensus networks generated by the algorithm's solutions. In other words, to convert the consensually agreed confidence levels of interactions into a definitive statement of their existence or absence in the network. To do this, an appropriate cutoff criterion must be designed to establish as definitive those interactions that are more reliable.

Algorithm 9 Third fitness function: Motifs.

Require: Consensus list with confidence values c, List of motifs ids to detect $motifs$
Ensure: Value of the fitness function $result$

```
1: binaryMatrix = getBinaryMatrix(c)
2: key = deepHashCode(binaryMatrix)
3: if key in keys(cache) then
4:     result = cache[key]
5: else
6:     g = getDirectedJGraph(binaryMatrix)
7:     result = 0.0
8:     for m in motifs do
9:         result -= count(m, g)
10:    end for
11: end if
12: return result
```

The implemented cutoff criterion selects the top $x\%$ interactions in the network, i.e., those with the highest consensual confidence levels. Assigning this $x\%$ threshold does not need to be done through a rigorous parametric exercise, because the number of motifs found will be closely tied to how permissive the cutoff criterion is. In this regard, the more relationships are ultimately reported in the network, the greater the number of motifs detected. This is why the decision was made to choose the top 40% of interactions as the cutoff. It's a sufficiently high quantity to form a functional gene network, but not too large to prevent a substantial distinction between the consensus networks generated by different solutions.

However, it is evident that by integrating the binarization step into this fitness function, there will be cases where multiple solutions lead to the same binary network. To optimize the computational cost of the algorithm, a cache was created to avoid repeating the motif detection exercise in binary networks that were previously evaluated during the algorithm's execution (lines 2 to 5 in Algorithm 9).

Finally, the binary matrix is converted to a directed graph from the JGrapht library [261] to facilitate the detection of different motifs in the consensus binary network (line 6 in Algorithm 9). Since JMetal is oriented towards goal minimization, the count of each motif will be subtracted to a cumulative variable that will represent the final score of this function (lines 7 to 9 in Algorithm 9).

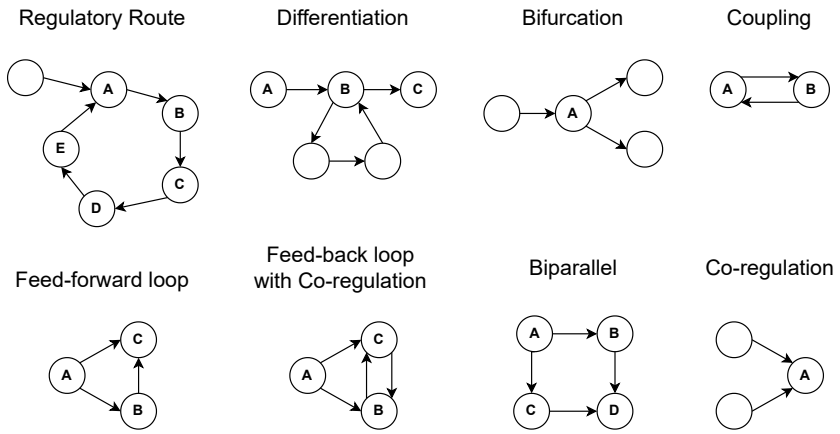


Figure 7.5: Motifs considered in the study of the third objective function. These motifs represent fundamental regulatory patterns commonly found in gene networks: (i) Regulatory Route, a sequential path of interactions with feedback to the initial node, reflecting ordered information flow; (ii) Differentiation, a branching pathway without cycles, essential for controlling cell specialization; (iii) Bifurcation, a node splitting into multiple successors, allowing diverse responses to signals; (iv) Coupling, reciprocal regulation between two nodes, contributing to stability and homeostasis; (v) Feed-forward loop, where a regulator influences a target both directly and indirectly via an intermediate factor, conferring robustness against fluctuations; (vi) Feedback loop with co-regulation, which adds reciprocal interactions between regulators and targets, enhancing coordination and precision; (vii) Biparallel, representing indirect multi-path regulation that, taken together, reflects strong functional relationships; and (viii) Co-regulation, where multiple regulators converge on the same gene, enabling fine-tuned and context-specific expression control. These motifs are used as structural building blocks to evaluate the proposed motif-based objective function.

In this case, unlike the two previous functions, instead of considering various versions, different motifs were analyzed to see which of them could better approximate the algorithm's solutions to real gene networks. Although 8 specific motifs were rigorously compared in the end, it should be noted that more manual testing was done earlier, which included other patterns. This testing led to a final list of candidates, represented in Figure 7.5, and described below:

- **Regulatory Route**: It is a specific sequence of interactions between nodes in a directed network or graph. This pattern arises as a result of the regulation of biological processes or any other phenomenon where information flows sequentially and orderly through a series of components. In this pattern, an initial node has a single successor, and each descendant node in the path has a single predecessor. Additionally, it is required that the last node in the path has a back edge to the initial node. The method counts

both, the number of regulatory paths present in the graph and their size.

- **Differentiation**: It is a pattern in a directed graph where nodes represent a cell differentiation process. This motif is fundamental in developing and maintaining tissues and organs in multicellular organisms, as it allows for the precise regulation of cell differentiation and prevents overproduction or underproduction of differentiated cells. A node is considered part of a differentiation pathway if it can be reached from an initial node without forming cycles. The method counts the number of nodes participating in the graph's differentiation pathways.

- **Bifurcation**: It represents a branching point in the graph, where a node has at least two successors and a single predecessor, i.e., the node branches into multiple pathways. Branch points in gene regulatory networks allow cells to respond to a wide range of signals and stimuli. When a cell receives a signal, it can activate or deactivate specific genes, leading to various outcomes.

- **Coupling**: It represents a coupling relationship between two nodes in the graph, with an edge connecting them in both directions of regulation. The method counts the number of coupled node pairs present in the graph. In gene regulatory networks, coupling can contribute to stability and homeostasis. This is supported in [262] where this motif is considered as a self-regulation of length 2. Coupled genes can maintain balanced expression and avoid excessive fluctuations with reciprocal interaction.

- **Feed-forward loop**: It involves the interaction of three main elements: a regulatory transcription factor (TF1), an intermediate transcription factor (TF2), and a target gene (G). In this motif, TF1 directly regulates the expression of the target gene G while activating the expression of the intermediate transcription factor TF2, which also modulates the expression of G, creating a cascade of regulation. This motif contributes stability and robustness to gene regulatory networks. The presence of TF2 acts as an additional regulator that dampens fluctuations in the TF1 signal, preventing excessively rapid or inappropriate responses to changes in the environment.

- **Feedback Loop with CoRegulation**: This motif is similar to the previous one, but it incorporates an additional interaction between the target gene G and the intermediate transcription factor TF2 (which, in this case, could be considered as another gene) in the opposite direction to the existing one. Feedback allows for the adjustment and maintenance of gene expression levels within certain limits, while co-regulation ensures that genes in

Table 7.4: Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.

AUPR		
Motifs Version	$Friedman'sRank$	$Holm'sAdj - p$
***Regulatory route**	**3.00581**	-
Differentiation	3.02326	0.96276
Bifurcation	3.38953	0.60861
Coupling	3.79651	0.10284
Feed-forward loop	5.38953	7.020e-10
Feed-back loop co-regulation	5.48256	1.674e-10
Biparallel	5.55233	5.571e-11
Co-regulation	6.36047	1.885e-18

the circuit are activated or repressed simultaneously and coordinately, enhancing the precision and efficiency of the biological response.

- **Biparallel**: In this motif, situations are sought where a transcription factor indirectly regulates the expression of a gene through multiple pathways, i.e., indirect interactions that individually lack significance, but together signify a close relationship between both elements.

- **Co-regulation**: It occurs when a gene is simultaneously regulated by more than one transcription factor. This motif is common in gene regulatory networks because a gene often has multiple regulatory genes or regulatory pathways that converge on it. Gene expression is a highly coordinated process that requires the appropriate activation or repression of genes at different times and in different tissues. The presence of multiple transcription factors acting in concert on a gene provides the possibility of finer and more adaptable regulation.

In the first instance, a fitness function was designed to count each motif (considering additional aspects such as size in specific cases mentioned earlier). To assess the accuracy of each fitness function individually, the same temporal configuration used for comparing versions seen in the previous objectives was employed again.

The results are shown in Table 7.4 and Table 7.5. In this case, instead of selecting a specific option and limiting the optimization of this objective to a single motif, it was considered more appropriate to build a function that considers the joint detection of the best motifs. In this case, the top 4 from the ranking have been chosen, as no statistically significant difference can be claimed between them for both metrics.

Table 7.5: Friedman mean rank with Holm's adjusted p values (0.05) for AUROC.

AUROC		
Motifs Version	$Friedman's Rank$	$Holm's Adj - p$
***Bifurcation**	**3.32558**	-
Regulatory route	3.48837	1.02661
Differentiation	3.56977	1.02661
Coupling	4.3314	0.02127
Feed-forward loop	4.91279	8.587e-05
Biparallel	5.27326	9.238e-07
Co-regulation	5.5407	1.818e-08
Feed-back loop co-regulation	5.55814	1.594e-08

7.2 Experimentation

In this section, precise information will be presented regarding the study conducted to evaluate the quality of `MO-GENECI` compared to the set of individual techniques.

The dataset used corresponds to the full framework described in section 4.1, with the sole exception of the DREAM5 networks. These networks were excluded due to the high variability observed in their results and the previously reported concerns in the literature regarding the validity of their associated gold standards. Consequently, they were not considered in the experiments conducted from this point onward.

In these experiments, the proposed approach will be evaluated considering the 26 inference techniques described in the state of the art (section 3.1.1). However, due to the limitations and high computational costs of some of them, certain techniques have been eliminated for certain network size ranges. In this sense, all techniques are employed for networks with fewer than 25 genes. For those with a size between 25 and 110 genes, all are used except for JUMP3. For networks with sizes between 110 and 250 genes, the following techniques are removed from the list: JUMP3, TIGRESS, CMI2NI, LOCPCACMI, GRNVBEM, and NONLINEARODES. For networks larger than 250 and smaller than 2,000, the following are also discarded: PCACMI, PLSNET, INFERELATOR, GENIE3_RF, GENIE3_ET, GRNBOOST2, and MEOMI. Lastly, for networks larger than 2,000 genes, PUC and PIDC are also eliminated.

Regarding computational resources, it has been decided to parallelize the execution of all techniques, including an intelligent allocation of cores, to provide more resources to the techniques that need them most to synchronize their com-

Table 7.6: The first 5 configurations from the statistical Friedman ranking for each metric, indicating significance compared to the winner based on non-parametric Holm tests (R = Rejected and A = Accepted). The configurations are presented using the following abbreviations: PS = Population Size, CP = Crossover Probability, NP = Number of Parents, MP = Mutation Probability y MS = Mutation Strength.

Ranking	EP	GD	HV
1	PS200-CP0.7-MP0.2-NP3-MS0.3	PS300-CP0.8-MP0.2-NP3-MS0.3	PS200-CP0.7-MP0.2-NP3-MS0.3
2	PS100-CP0.7-MP0.2-NP3-MS0.3 (R)	PS300-CP0.9-MP0.2-NP3-MS0.3 (R)	PS100-CP0.7-MP0.2-NP3-MS0.3 (R)
3	PS200-CP0.8-MP0.2-NP3-MS0.3 (R)	PS300-CP0.7-MP0.2-NP3-MS0.3 (R)	PS100-CP0.8-MP0.2-NP3-MS0.3 (R)
4	PS100-CP0.8-MP0.2-NP3-MS0.3 (R)	PS300-CP0.7-MP0.1-NP3-MS0.3 (R)	PS200-CP0.8-MP0.2-NP3-MS0.3 (R)
5	PS300-CP0.7-MP0.2-NP3-MS0.3 (R)	PS300-CP0.9-MP0.1-NP3-MS0.3 (R)	PS300-CP0.7-MP0.2-NP3-MS0.3 (R)
Ranking	IGD	IGD+	SP
1	PS300-CP0.9-MP0.05-NP4-MS0.1	PS300-CP0.7-MP0.05-NP4-MS0.1	PS100-CP0.9-MP0.2-NP4-MS0.2
2	PS300-CP0.8-MP0.05-NP4-MS0.1 (A)	PS300-CP0.8-MP0.05-NP4-MS0.1 (A)	PS100-CP0.7-MP0.05-NP4-MS0.2 (R)
3	PS300-CP0.7-MP0.05-NP4-MS0.1 (A)	PS300-CP0.9-MP0.05-NP4-MS0.1 (A)	PS100-CP0.9-MP0.05-NP4-MS0.2 (R)
4	PS200-CP0.9-MP0.05-NP4-MS0.1 (R)	PS200-CP0.9-MP0.05-NP4-MS0.1 (R)	PS100-CP0.7-MP0.05-NP4-MS0.1 (R)
5	PS200-CP0.8-MP0.05-NP4-MS0.1 (R)	PS200-CP0.8-MP0.05-NP4-MS0.1 (R)	PS100-CP0.7-MP0.1-NP4-MS0.2 (R)

pletion times. Therefore, the techniques have been divided into four groups based on their resource priority, from highest to lowest. In the first group, the most resource-intensive techniques are JUMP3, LOCPCACMI, NONLINEARODES, GRNVBEM, and CMI2NI. The second group has TIGRESS, PCACMI, PLSNET, INFERELATOR, GENIE3_RF, GRNBOOST2, and GENIE3_ET. In the third group, only KBOOST and LEAP are included. In the last group, the rest of the techniques either have minimal computational cost or their implementations prevent parallelization, so they would not use the allocated cores.

7.2.1 Parameter Settings

To enhance the quality of `MO-GENECI`, a parameter setting procedure has been carried out to find the optimal combination of values for the following parameters: population size (100, 200, or 300), crossover probability (0.7, 0.8, or 0.9), number of parents (3 or 4), mutation probability (0.05, 0.1, or 0.2), and mutation strength (0.1, 0.2, or 0.3). In this exercise, it was decided to include all gene networks from the benchmark with a size of fewer than 500 genes, consisting of 82 networks. Five independent runs were performed for each parameter combination and gene regulatory network, each consisting of a total of 100,000 evaluations. The termination criterion used was PercLinksWithBestConf, with a threshold set at 0.4 (as previously mentioned during the explanation of the third fitness function).

Once all the results were obtained, a reference Pareto front was constructed for each problem by selecting the best solutions from each combination. Sub-

sequently, the reference front was compared to each independent result, and the following metrics were calculated: Epsilon (EP) [84], Generational Distance (GD) [85], PISAHypervolume (HV) [86], Inverted Generational Distance (IGD) [85], Inverted Generational Distance Plus (IGD+) [87], and Spread (SP) [79]. With these results, a statistical ranking using Friedman's test with non-parametric Holm tests was applied for each metric to determine which combination of values performed better.

The total of 162 parameter combinations with 5 independent runs for each of the 82 gene networks considered amounts to a total of 66,420 executions. The extent of this is the reason why it was decided to present the results of these executions and the subsequent statistical ranking of each metric in the documentation section of the main repository as supplementary material [2]. Nevertheless, Table 7.6 provides a summary of the results.

After reviewing the results, two candidate configurations stand out: PS200-CP0.7-MP0.2-NP3-MS0.3 and PS300-CP0.9-MP0.05-NP4-MS0.1. While the first configuration leads the ranking for the metrics Epsilon (EP) and PISAHypervolume (HV), the second one performs well in Inverted Generational Distance (IGD) and Inverted Generational Distance Plus (IGD+). Despite this information, which may make the decision seem complex, it should be noted that the first candidate ranks near the bottom for the rest of the metrics, while the second one falls in more intermediate positions. For this reason, it has been decided to select the second configuration as the optimal choice for `MO-GENECI`.

7.2.2 Experimental Procedure

After collecting all the gene networks to be inferred, selecting the individual techniques to use for each of them, and setting the parameters, an experimental procedure has been designed to assess the actual performance of the proposal concerning a large set of techniques in the state of the art.

Firstly, the corresponding techniques were executed individually in parallel for each gene expression dataset. This provides a proposed network for each technique and instance. Secondly, to consolidate the networks proposed by these techniques, for each problem, `MO-GENECI` was executed with the previously fixed parameters, setting a maximum of 250,000 evaluations. This provides a Pareto front with multiple solutions for each problem.

Thirdly, the networks proposed by the individual techniques and all those de-

[2] `https://github.com/AdrianSeguraOrtiz/MO-GENECI/tree/main/docs/parameterization`

rived from MO-GENECI (one consensus network for each solution from the front) were compared to the gold standard in the test phase. This yields an AUROC and AUPR value for each network proposal.

Fourthly, since choosing a single solution from the Pareto front is left to the domain expert, it was decided to extract two statistically significant solutions for subsequent comparison with the individual techniques. The first one was called BEST_MO-GENECI, referring to the solution from the front whose consensus network achieves the best AUPR and AUROC values. The second one was named MEDIAN_MO-GENECI, representing the solution whose accuracy is the median of the front. In this way, the goal is to demonstrate the potential of MO-GENECI when choosing the appropriate solution (BEST_MO-GENECI) and the good quality it can provide even when this choice is made without prior knowledge (MEDIAN_MO-GENECI). It is worth mentioning that during the subsequent results discussion (specifically in the explanation of Figure 7.8), a clear trend will be shown that undoubtedly facilitates the selection of a good solution from the front based on basic network characteristics such as size.

Finally, a systematic comparison procedure was initiated after collecting the precision levels for each gene network from the individual techniques and the two solutions extracted from MO-GENECI. The different proposals were compared for each size range using the standard Friedman statistical ranking and Holm's non-parametric tests. The results are presented in the following section of this manuscript.

Furthermore, once the efficiency of this approach was validated, it was decided to run MO-GENECI on a real-world dataset from melanoma patients. This same set was also employed in GENECI and this allows us to observe whether the performance improvements translate into new discoveries that did not emerge after running the preliminary algorithm.

7.3 Results and Discussion

This section analyses the results obtained by MO-GENECI for the total benchmark of 106 networks considered in this study. A chronological order based on the workflow described earlier is followed to facilitate the comprehension of this analysis. In addition to analyzing the accuracy of MO-GENECI, a couple of subsections are included to discuss the computational cost of the algorithm and the results of MO-GENECI when run on a real-world data set.

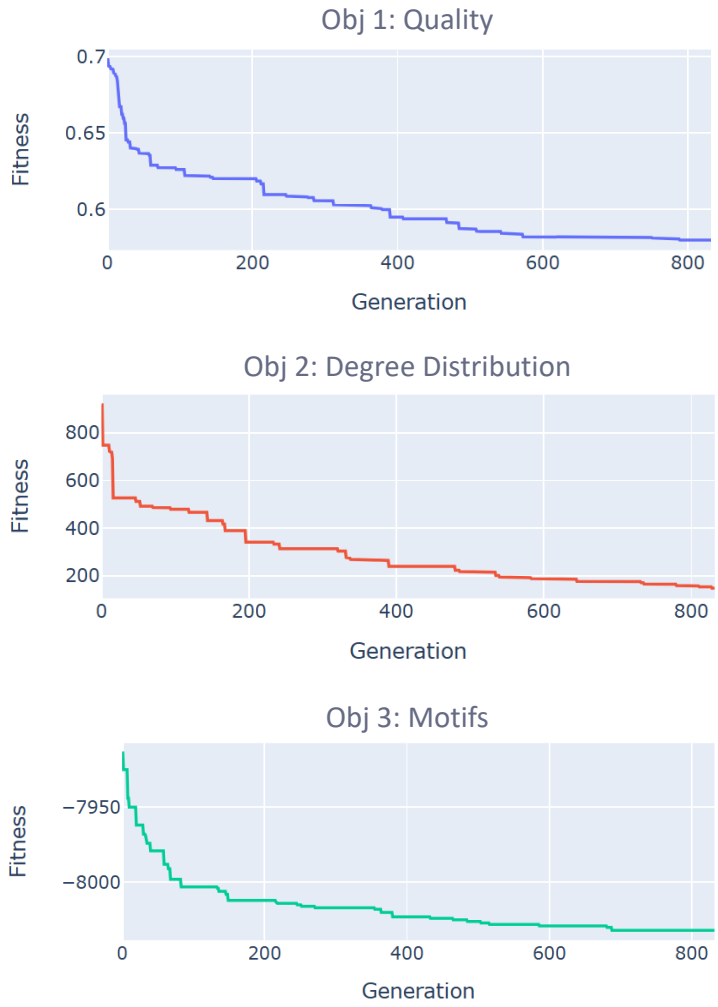


Figure 7.6: Evolution of fitness functions for the 200-node network constructed from scratch with a scale-free degree distribution and subjected to overexpression perturbation.

7.3.1 MO-GENECI internal behavior

In `MO-GENECI`, the three objective functions are optimized simultaneously over the iterations. Despite not pursuing the individual optimization of each one, it proves quite useful to graph the best individual result for each function in each generation. This allows for the verification of the algorithm's learning procedure, the proper selection of individuals, and the ability of the crossover and mutation operators to achieve uniform optimization of the three objectives, while maintaining the right balance between exploration and exploitation.

In Figure 7.6, a typical execution of MO-GENECI for the 200-node network, constructed from scratch with a scale-free degree distribution and subjected to over-expression perturbation, is depicted. The three objective functions indeed show improvement in their results as the evolutionary algorithm progresses. The slight stepwise behavior in the plots is attributed to the network's size. In small networks, this stepping is more prominent, particularly in functions related to network topology, while in large networks comprising thousands of genes, this stepping smoothens and ultimately disappears.

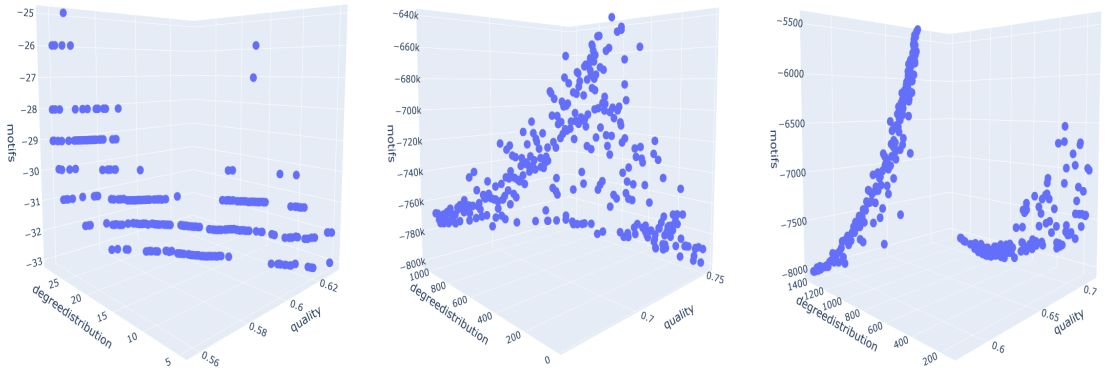
The number of generations is approximately 800, which aligns with the chosen population size of 300 individuals and a maximum of 250,000 evaluations per run. This configuration was established after verifying that the proposed approach is able to converge at this number of generations, for the three objectives, as shown in Figure 7.6, so no significant improvements were detected after this stopping condition (hence, to save on computational effort).

Concerning the visible range of values for each function, it aligns with the previously specified implementation. The *quality* values are normalized between 0 and 1, *degreedistribution* values fall within a range between 0 and ∞ due to the goodness-of-fit test, and *motif* values range between 0 and $-\infty$ due to negative motif counting.

After executing MO-GENECI, the output is obtained as the set of non-dominated solutions from the last generation (in the form of Pareto front approximation). These individuals are represented in interactive 3D plots that facilitate visualization of the obtained Pareto front for each problem. Figure 7.7 displays three specific cases of interest, allowing for the derivation of conclusions from this approach.

Firstly, subfigure 7.7a shows the obtained Pareto front for a relatively small network consisting of 13 genes. The most notable aspect of this case relates to what was mentioned previously regarding the stepping behavior. In such small networks, the *motifs* objective function has limited room for optimization, with a relatively narrow and discretized range of values. Therefore, the representation of the front appears as a set of curves parallel at different heights.

Nevertheless, in subfigure 7.7b, the opposite scenario is shown: a large network with 2,000 genes where all objective functions have significant room for optimization. Consequently, the solutions are more evenly distributed throughout the plot. As expected, three vertices stand out, corresponding to the extreme cases of each objective function, where one objective is maximized at the expense of receiving poor scores in the other two functions.



(a) 13-gene network from rattus norvegicus organism obtained from TFLink subjected to mixed perturbation.

(b) Simulated 2000-gene network by GeneNetWeaver.

(c) 200-node network constructed from scratch with a scale-free degree distribution and subjected to overexpression perturbation.

Figure 7.7: Set of non-dominated solutions in Pareto front approximation from the final population.

However, there are exceptions, as shown in subfigure 7.7c, where a network's characteristics break the opposition between the functions. In this case, it is impossible to optimize *degreedistribution* until *quality* presents extremely unfavorable values (curve with a shallower slope), and vice versa, *quality* cannot be optimized until *degreedistribution* values move away from the minimum (curve with a steeper slope).

From a different perspective, `MO-GENECI`'s output also provides a representation of parallel coordinates for the set of non-dominated solutions from the last population. However, in this case, work has been carried out with gold standards in this case in preparation for subsequent comparison between techniques. That is why the decision was made to expand the parallel coordinates initially presented by `MO-GENECI` with the accuracy values for each individual concerning the known network. Additionally, to improve visualization, apart from adding the columns AUROC and AUPR, a new column representing the average of the two previous ones has been created. This allows individuals to be color-coded with varying intensity based on their accuracy. To obtain this expanded representation, the command *evaluate dream/generic-prediction dream/generic-pareto-front* is executed, which takes a front of solutions and the known network as input.

These newly evaluated parallel coordinates serve a dual purpose. First, they allow for an analysis of the intensity and shape of the trade-offs between different objectives. Second, they help identify certain patterns regarding optimiza-

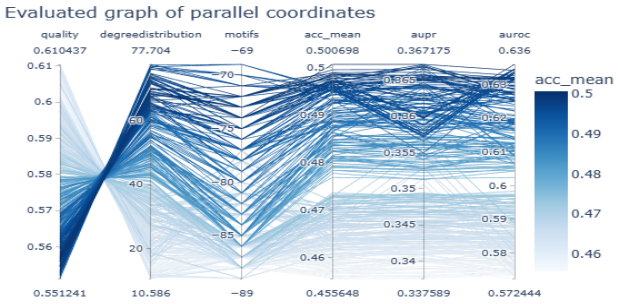
tion levels in the graphs that lead to high-quality results. In Figure 7.8, three significant cases have been represented in this sense.

In subfigure 7.8a, the "evaluated" parallel coordinates for a 20-gene network are represented, from which, four observations can be made. Firstly, the stepping behavior mentioned in the previous plots for small networks is still present in this representation. As can be seen, the *motifs* column has a limited number of values, grouping individuals at quite specific heights. Secondly, the opposition between different objectives is evident, with the differences between *quality* and *degreedistribution* being particularly noticeable in this case, showing a complete contrast. Thirdly, by analyzing the graphics, the coherence of this approach can be verified. Solutions with similar optimization profiles result in networks with similar levels of accuracy. Lastly, due to the network's size, the objective function that contributes the most accuracy to the solutions is *quality*. This is because neither *degreedistribution* nor *motifs* make sense in such small networks. There is no room for the existence of hubs or the appearance of complex motifs. Therefore, improving solution accuracy relies solely on giving more weight to the most reliable individual techniques than the others.

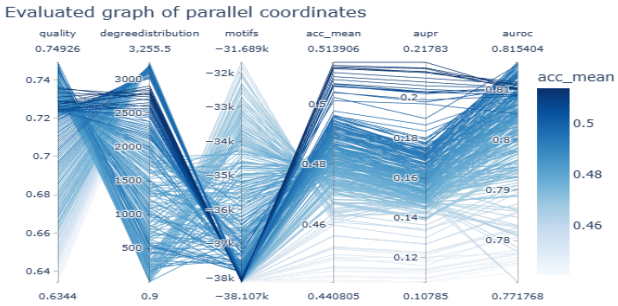
The second significant case is a medium-small network with 436 genes. Subfigure 7.8b shows that, for this case, *quality* loses the prominence mentioned for networks with few genes, *degreedistribution* still lacks significant room to contribute accuracy to the solutions. Finally, *motifs* begins to play a key role in the optimization process. In fact, it is evident how this objective begins to show greater opposition to the other ones, which was not as visible in the previous example. Once again, the coherence of this approach can be checked, with a clear gradient of color in all represented objectives.

Finally, in subfigure 7.8c, the case of a large network with 2,000 genes is shown. The increase in size compared to the previous case is the reason why *degreedistribution* gains importance in improving solution accuracy. When networks have more realistic sizes, both *degreedistribution* and *motifs* serve a more coherent purpose and, therefore, play a more significant role in the effectiveness of this approach.

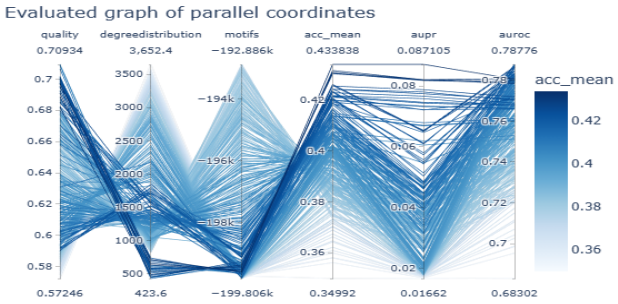
It should be noted that the last and first examples are opposite cases, and the difference in their sizes has a similar effect on the optimization profiles for higher accuracy. This is why, when explaining BEST_MO-GENECI and MEDIAN_MO-GENECI, reference was made to this section to clarify that choosing a good solution is not a complicated task considering the analysis just presented. For smaller networks, it is logical to focus on the reliability of the techniques rather than topological characteristics. In contrast, for networks of realistic size, the



(a) 20-node network with scale-free degree distribution and subjected to a knock-down perturbation.



(b) 436-gene network from the gallus gallus organism obtained from BioGrid subjected to a mixed perturbation.



(c) Simulated 2000-gene network by SynTReN.

Figure 7.8: Extended parallel coordinate plots showing Pareto front individuals. The first columns are optimization objectives, the last ones quality metrics, and color intensity encodes the mean accuracy (acc_mean).

concordance between techniques diminishes, and more biological aspects like degree distribution or motifs become more relevant.

7.3.2 Experimental comparisons

The next step in this study is the comparison of MO-GENECI with all the individual inference techniques. After evaluating all the networks generated by these techniques and extracting the aforementioned BEST_MO-GENECI and MEDIAN_MO-GENECI solutions, AUROC and AUPR values were obtained for each network and inference method. Since presenting these values for all 106 networks would be extensive, a series of visual plots are provided to summarize and facilitate understanding of the results. However, detailed data for both metrics can be found in the main project repository as supplementary material [3].

In Figure 7.9, the comparison of accuracy values for a subset of networks with a size of up to 25 genes is presented. The main objective of this proposal is fully achieved: to provide solutions that, without attempting to outperform individual techniques in their respective domains of expertise, have high accuracy for any type of network to be inferred. The existence of these specializations is also clearly exposed in the plot. Not only is there diversity in the winning individual techniques (e.g. CMI2NI, GENIE3_ET, ARACNE, JUMP3, PCACMI, etc.), but there are significant differences in their rankings depending on the case. For instance, JUMP3 is the technique with the highest AUPR value for the D4_10_3 network, yet it ranks last for the AUROC metric in the TFL-RN-M network. PUC is the AUROC winner for D3_10_E1, but it holds a rather unfavorable position in the AUROC of its neighboring BG-SV40-M. Another example is ARACNE, which seems not to excel in any network except for the exceptional case of the TFL-RN-M network. MO-GENECI surpasses these individual specializations and attains excellent rankings across all networks thanks to its adaptability and flexibility.

Furthermore, it should be noted that even when the best solution from the front is not chosen appropriately, the effectiveness of MO-GENECI remains relatively high. As can be seen, the values of MEDIAN_MO-GENECI are pretty close to those of BEST_MO-GENECI (for example, in D3_10_E1, D4_10_3 or FS-SF20-O).

It is also worth mentioning that when running MO-GENECI with all inference techniques, any information that the researcher may have about individual inference techniques is ignored. While it is true that domains of specialization are

[3] https://github.com/AdrianSeguraOrtiz/MO-GENECI/tree/main/docs/experimentation

Figure 7.9: AUROC and AUPR values for networks up to 25 genes. Each column represents a gene network, and each row represents a different metric. The upper row shows AUROC values, while the lower row displays AUPR values. Each cell contains two sets of bars stacked from lowest to highest height. In the first set of bars, the values obtained by BEST_MO-GENECI and MEDIAN_MO-GENECI are represented, while the second set contains the values from the other individual techniques. The number of individual techniques surpassed by BEST_MO-GENECI is indicated on the first set of bars, which can be visualized thanks to the dashed horizontal line that sets the threshold. The name of the winning individual technique that achieved the best results for the given network and metric is displayed on the second set of bars.

difficult to delineate, there is still some minimum knowledge that could help researchers eliminate certain techniques for their specific case. This would filter out some noise in the consensus and further improve the results obtained.

Additionally, this analysis is not only helpful in comparing the proposal of this chapter with other individual techniques but also serves as a systematic and robust comparison among them. This study can contribute to the state of the art by providing knowledge about the performance of individual techniques and their effectiveness in this field. This can be explored in more detail in section 7.3.3.

Figure 7.10 shows the AUPR and AUROC values obtained by the inference methods for a subset of networks with sizes ranging from 25 to 110 genes. The same information as in Figure 7.9 is presented for different networks and using a different visual representation.

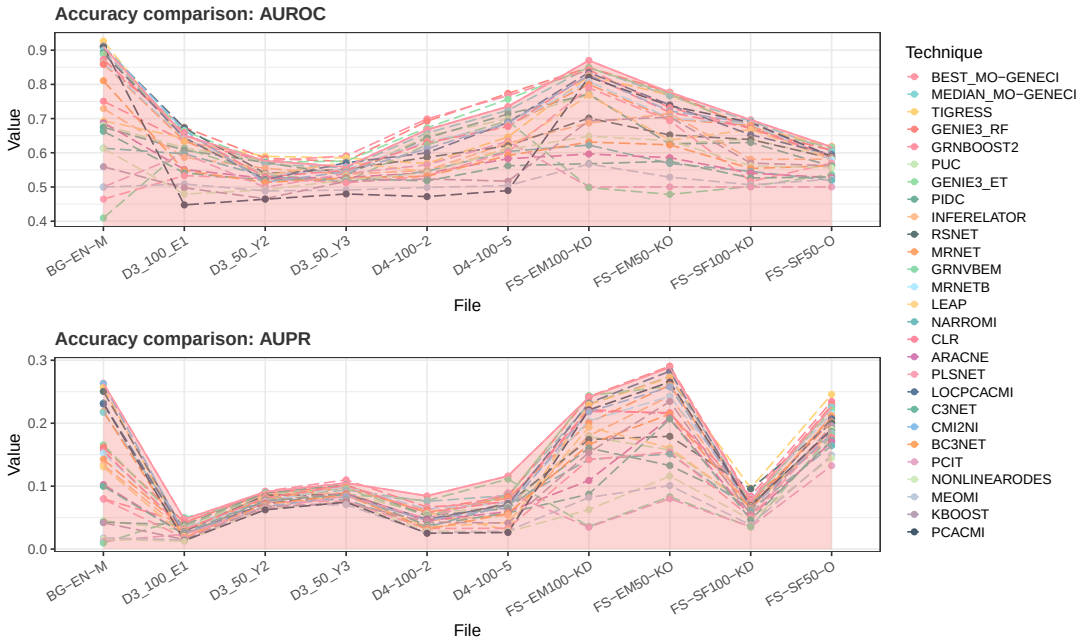


Figure 7.10: AUROC and AUPR values for networks between 25 and 110 genes in size. Each plot displays the results for a specific metric. The upper plot shows the AUROC values, while the lower one shows the AUPR values. In both plots, the vertical axis represents the values of the corresponding metric, and the horizontal axis represents the different networks considered in the figure. Each technique is assigned a color and represented by a series of points connected by dashed lines. To facilitate the comparison of BEST_MO-GENECI with the other techniques, a continuous representation of its curve and shading over its area have been used.

In this case, the fact that some techniques perform well for certain networks and worse for others is evident at the intersections between the dashed lines. The most noticeable case due to its color is PCACMI, as its low accuracy for networks derived from the DREAM challenges and its high quality for the remaining networks in the plot are clearly visible.

Once again, `MO-GENECI` gains generalization capacity and achieves good results for all cases, overshadowing the outcomes of individual techniques. However, as mentioned earlier, this proposal is eventually surpassed by techniques specialized in the domain of expertise in which the inferred gene network falls.

Figure 7.11 presents the accuracy values obtained by different techniques for a subset of networks with sizes between 110 and 250 genes. The intersections between the lines again highlight the specialization of techniques.

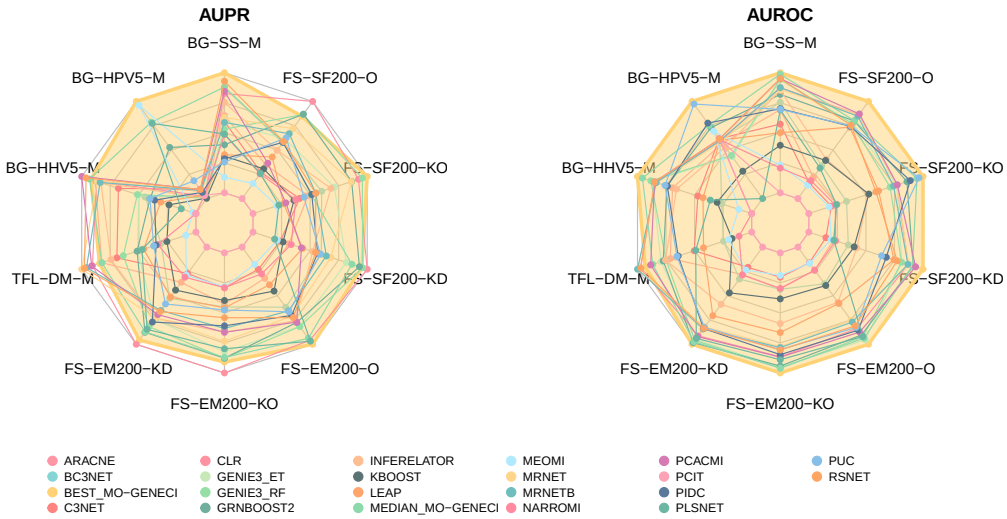


Figure 7.11: AUROC and AUPR values for networks between 110 and 250 genes in size. In this case, radar charts have been used, which clearly represent the specializations of different techniques. Once again, shading has been applied to BEST_MO-GENECI (yellow), highlighting its ability to cover all domains of expertise.

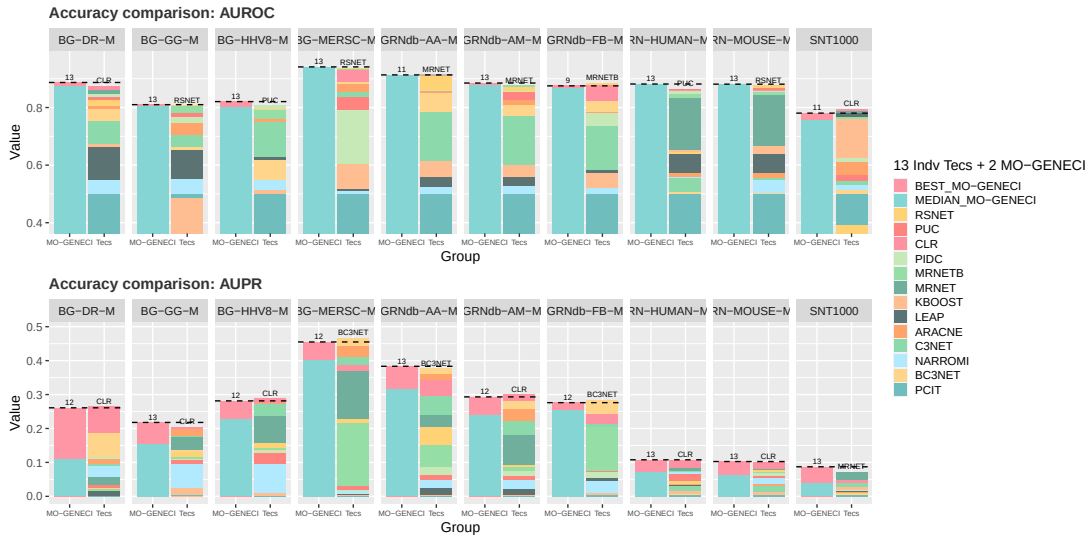


Figure 7.12: AUROC and AUPR values for networks between 250 and 2,000 genes in size. La explicación de esta representación es la misma que en la Figura 7.9

In this sense, it is worth noting the case of the simulated BioGrid network "*Human papillomavirus 5*" subjected to a mixed perturbation (BG-HPV5-M). In

this network, techniques like MEOMI, which show unfavorable results for the other networks, improve their quality. In contrast, other clearly outstanding techniques for the rest of the networks significantly reduce their accuracy. This is a clear example of the risk that a researcher runs when choosing a specific technique based on the results shown in other gene regulation networks. This is where the utility of MO-GENECI becomes evident, as it has remained immune to this change in techniques and has been able to ultimately redistribute its weights to achieve a high-quality result.

Additionally, PCIT achieves poor accuracy for all networks in this representation, yet it was declared the winner in Figure 7.9 for the AUPR metric in the D3_10_E1 network.

The last representation is shown in Figure 7.12, for a subset of networks with sizes between 250 and 2,000 genes. It is evident how the accuracy of MO-GENECI and the number of surpassed individual techniques increase with the size of the networks. Furthermore, the distances between BEST_MO-GENECI and MEDIAN_MO-GENECI are further reduced in these cases. Once again, all the observations made so far about the capabilities of MO-GENECI can be clearly confirmed. This proposal has demonstrated versatility for different domains of expertise and shows its potential to be more than competitive in networks of all sizes. The specialization by individual techniques continues to be present, as seen in the previous comparisons.

Finally, two networks with sizes exceeding 2,000 genes are left to analyze. Due to the small number of networks in this category, it has been decided to create Table 7.7 displaying the metric values explicitly instead of representing them graphically. It highlights in bold the best value achieved by individual inference techniques and the one obtained by BEST_MO-GENECI for each network and metric.

For the simulated 2000-gene network by GeneNetWeaver, it can be seen that BEST_MO-GENECI achieves a competitive AUPR value, while for the AUROC metric, it clearly outperforms the rest of the techniques (also MEDIAN_MO-GENECI). In the case of the RegulonDB network, it is quite similar to the previous one. However, this time, for the AUPR metric, the CLR technique presents a clear outlier unsupported by the rest of the techniques, which is difficult for MO-GENECI to detect. It is a clear example of a technique that surpasses the consensus due to its domain of expertise. Nevertheless, MO-GENECI's values are clearly competitive compared to the other techniques.

It is also worth mentioning that during this section, MO-GENECI has been compared to the best AUROC and AUPR values for each network, regardless

Table 7.7: AUROC and AUPR values for networks of more than 2,000 genes

Network	gnw2000		regulonDB	
Technique	AUPR	AUROC	AUPR	AUROC
ARACNE	0.3835	0.6990	0.0154	0.5739
BC3NET	0.2090	0.6342	0.0393	0.5975
C3NET	0.3183	0.6620	0.0131	0.5621
CLR	0.1782	0.8150	**0.0960**	0.9635
KBOOST	**0.3959**	0.7997	0.0064	0.6072
LEAP	0.0505	0.7052	0.0140	0.8228
MRNETB	0.0733	0.8205	0.0198	0.9591
MRNET	0.2715	**0.8270**	0.0264	0.9609
NARROMI	0.0196	0.5490	0.0341	0.6144
PCIT	0.0026	0.5000	0.0008	0.5000
RSNET	0.0564	0.8031	0.0300	**0.9664**
MEDIAN_MO-GENECI	0.1590	0.8711	0.0207	0.9681
BEST_MO-GENECI	**0.3486**	**0.8750**	**0.0353**	**0.9751**

of whether these values come from the same technique or not. In other words, for a specific network, when MO-GENECI performance is just below one technique in the AUPR metric, and below another in AUROC, it does not mean it is below both. The two metrics should be taken into account, and the fact that a technique excels in one of them but performs poorly in another is counterproductive. However, MO-GENECI has demonstrated stability for both cases with consistently high values.

7.3.3 Statistical Significance

This section presents a statistical analysis with the aim of rigorously comparing the performance of individual inference techniques, as well as BEST_MO-GENECI and MEDIAN_MO-GENECI. This analysis takes a global perspective, considering AUROC and AUPR metrics for all the networks in the dataset of this study.

Specifically, the Friedman statistical ranking with non-parametric Holm tests has been calculated for each subset of networks organized by sizes. This is because there are techniques that could not be executed for certain sizes, so they can only be compared with the other techniques that have been executed within their subset.

There are a total of 5 groups determined by the thresholds of 25, 110, 250,

Table 7.8: Friedman mean rank with Holm's adjusted p values (0.05) for the AUPR metric measured across all techniques in networks with 0 to 25 genes.

AUPR 0-25 genes		
Technique	$Friedman's Rank$	$Holm's Adj - p$
***BEST_MO-GENECI**	**4.94444**	-
CMI2NI	8.0926	1.59677e-01
MEDIAN_MO-GENECI	9.2778	1.05849e-01
INFERELATOR	10.8889	2.37812e-02
RSNET	11.1852	2.124633-02
GENIE3_ET	11.4444	2.10120e-02
LOCPCACMI	11.4815	2.10120e-02
PCACMI	11.8704	1.38443e-02
GENIE3_RF	12.1296	1.06427e-02
LEAP	13.3148	1.66457e-03
CLR	13.8519	6.93208e-04
MRNET	14.2222	3.75372e-04
MRNETB	14.5370	2.19631e-04
TIGRESS	14.8889	1.15967e-04
BC3NET	15.0185	9.52642e-05
ARACNE	15.1667	7.46093e-05
KBOOST	15.2963	6.02883e-05
GRNBOOST2	15.7407	2.41250e-05
PUC	16.0185	1.36096e-05
C3NET	16.1852	9.77589e-06
PLSNET	16.3333	7.27584e-06
PIDC	17.0926	1.20942e-06
JUMP3	17.4259	5.44535e-07
NONLINEARODES	18.3333	5.12226e-08
MEOMI	19.2963	3.48254e-09
NARROMI	20.0185	4.15433e-10
GRNVBEM	20.0370	4.08150e-10
PCIT	21.9074	9.56885e-13

and 2,000 genes. However, in the last group, this statistical study has been omitted as it consists of only two networks, which is an insufficient quantity to draw reliable conclusions. This results in a total of 8 tables, that is, two for each subset of networks, one for comparing AUPR and another for AUROC.

The first subset consists of networks with up to 25 genes. Table 7.8 shows the results for the AUPR metric, and Table 7.9 presents the results for AUROC. In this set of networks, most of the techniques participate, including the 26 individual techniques and the two extractions from `MO-GENECI`. It is observed that BEST_MO-GENECI is the best-ranked resulting technique for both metrics.

Table 7.9: Friedman mean rank with Holm's adjusted p values (0.05) for the AUROC metric measured across all techniques in networks with 0 to 25 genes.

AUROC 0-25 genes		
Technique	$Friedman'sRank$	$Holm'sAdj - p$
***BEST_MO-GENECI**	**4.57407**	-
MEDIAN_MO-GENECI	8.0185	1.23925e-01
GENIE3_ET	9.3519	6.56767e-02
CMI2NI	9.8519	5.52124e-02
GENIE3_RF	10.2778	4.33829e-02
INFERELATOR	11.1111	1.75100e-02
RSNET	12.2037	3.92833e-03
LEAP	12.8519	1.52490e-03
PCACMI	13.0926	1.13488e-03
CLR	13.5556	5.42568e-04
LOCPCACMI	13.7222	4.38619e-04
GRNBOOST2	14.0000	2.80650e-04
TIGRESS	14.2407	1.89164e-04
MRNETB	14.3148	1.76315e-04
KBOOST	14.4630	1.40115e-04
PLSNET	14.7222	8.73115e-05
MRNET	15.0926	4.19886e-05
NONLINEARODES	16.3148	2.70204e-06
ARACNE	16.3333	2.70204e-06
JUMP3	16.6481	1.31626e-06
BC3NET	16.7593	1.04978e-06
PIDC	17.1111	4.50595e-07
PUC	17.1667	4.08972e-07
C3NET	17.7037	1.03604e-07
MEOMI	18.8148	4.81678e-09
NARROMI	20.4630	3.18802e-11
GRNVBEM	20.5926	2.17753e-11
PCIT	22.6481	1.85180e-14

It's worth noting the CMI2NI technique, which is above the significance threshold in both cases. However, as seen later in the next subset regarding large-size networks, it acquires a rather unfavorable position and is surpassed by other inference techniques. In fact, CMI2NI becomes computationally infeasible for networks with more than 110 genes.

The next subset consists of networks with 25 to 110 nodes. The results for the AUPR and AUROC metrics are shown in Table 7.10 and Table 7.11, respectively. In these tables, once again, BEST_MO-GENECI leads these two statistical rankings.

Table 7.10: Friedman mean rank with Holm's adjusted p values (0.05) for the AUPR metric measured across all techniques in networks with 25 to 110 genes.

AUPR 25-110 genes		
Technique	$Friedman's Rank$	$Holm's Adj - p$
***BEST_MO-GENECI**	**3.55882**	-
GENIE3_RF	6.0882	1.88868e-01
CLR	7.6177	6.99917e-02
MEDIAN_MO-GENECI	8.2941	4.17035e-02
GENIE3_ET	8.5588	3.75826e-02
MRNETB	9.5000	1.01360e-02
TIGRESS	10.0588	4.40476e-03
MRNET	10.3235	3.08976e-03
GRNBOOST2	10.4412	2.80049e-03
LOCPCACMI	11.9118	1.28796e-04
RSNET	12.1176	8.74889e-05
PUC	12.5882	2.99879e-05
INFERELATOR	13.2647	5.53287e-06
PIDC	13.3529	4.71220e-06
CMI2NI	13.4412	3.98140e-06
BC3NET	14.8529	6.66146e-08
GRNVBEM	16.4118	3.91223e-10
ARACNE	17.0000	4.94054e-11
NARROMI	17.1471	3.02786e-11
PCACMI	17.3824	1.31674e-11
LEAP	18.0000	1.26040e-12
C3NET	18.1765	6.54808e-13
PLSNET	18.9706	2.61052e-14
KBOOST	20.0294	2.69102e-16
PCIT	20.2647	9.66022e-17
NONLINEARODES	24.2059	1.93437e-25
MEOMI	24.4412	5.32306e-26

In this case, it's worth highlighting the accuracy of techniques derived from the original GENIE3. That is the proposal that applies Random Forest regression (GENIE3_RF), the one that uses ExtraTrees regression (ET), and the one that employs the Stochastic Gradient Boosting Machine (GRNBOOST2). Although only GENIE3_RF is above the significance threshold in both cases, it can be seen that the other two also obtain good positions for the AUPR metric.

This was also observed in the previous proposal (see chapter 5), so that GENIE3 has a clear domain of specialization in the DREAM networks, and this subset of sizes contains a significantly higher concentration of networks from these

Table 7.11: Friedman mean rank with Holm's adjusted p values (0.05) for the AUROC metric measured across all techniques in networks with 25 to 110 genes.

AUROC 25-110 genes		
Technique	$Friedman'sRank$	$Holm'sAdj - p$
***BEST_MO-GENECI**	**3.73529**	-
GENIE3_RF	5.5000	4.09525e-01
GENIE3_ET	6.1765	4.09525e-01
MEDIAN_MO-GENECI	6.6471	3.91180e-01
GRNBOOST2	8.3529	6.58146e-02
TIGRESS	8.9118	3.58351e-02
CLR	9.1177	3.10503e-02
MRNETB	9.7941	1.15338e-02
CMI2NI	10.4118	4.19203e-03
MRNET	10.7353	2.48984e-03
LOCPCACMI	11.2353	9.78071e-04
PIDC	12.9118	2.05860e-05
PUC	13.1471	1.21583e-05
RSNET	13.5000	5.10683e-06
INFERELATOR	14.1176	9.68763e-07
LEAP	15.3529	2.38545e-08
PLSNET	16.2647	1.21400e-09
PCACMI	16.4412	6.97889e-10
BC3NET	18.4706	3.49502e-13
GRNVBEM	18.5000	3.27531e-13
ARACNE	19.2647	1.44156e-14
KBOOST	20.0882	4.16593e-16
NONLINEARODES	20.6471	3.43919e-17
NARROMI	21.0294	6.02465e-18
C3NET	21.4706	7.62403e-19
PCIT	21.5294	5.97097e-19
MEOMI	24.6471	4.50332e-26

challenges. GENIE3, despite obtaining good positions in the other tables, does not stand out as the best individual technique in the ranking in any other case, and in particular, for large sizes is much more computationally expensive than other techniques.

The third subset consists of networks with a size ranging from 110 to 250 genes. Table 7.12 displays the results for the AUPR metric, while Table 7.13 presents the AUROC results. In this case, it is observed that there are difficulties in confirming a statistically significant difference between the techniques, as nearly half of them are above the threshold. Nevertheless, it can be noted

Table 7.12: Friedman mean rank with Holm's adjusted p values (0.05) for the AUPR metric measured across all techniques in networks with 110 to 250 genes.

AUPR 110-250 genes		
Technique	$Friedman's Rank$	$Holm's Adj - p$
***BEST_MO-GENECI**	**2.83333**	-
CLR	5.0000	4.44534e-01
MEDIAN_MO-GENECI	6.5833	4.44534e-01
MRNET	6.6667	4.44534e-01
RSNET	7.4167	3.35301e-01
GENIE3_RF	7.9167	2.85960e-01
GRNBOOST2	8.0833	2.85960e-01
PCACMI	8.8333	1.65320e-01
MRNETB	9.1667	1.35140e-01
GENIE3_ET	10.1667	5.10336e-02
INFERELATOR	10.7500	2.82376e-02
ARACNE	11.5833	1.06108e-02
BC3NET	12.4583	3.39169e-03
PUC	13.1667	1.26136e-03
PLSNET	13.5000	8.02258e-04
PIDC	13.5833	7.51756e-04
C3NET	13.6667	7.00642e-04
LEAP	15.2500	4.78787e-05
NARROMI	17.9583	2.08922e-07
KBOOST	18.1667	1.38603e-07
MEOMI	18.9167	2.60800e-08
PCIT	21.3333	6.26486e-11

that BEST_MO-GENECI once again clearly leads both rankings. Furthermore, MEDIAN_MO-GENECI ranks second and third in the tables. It is surpassed by CLR in terms of AUPR, but this same technique ranks sixth for the AUROC metric.

It is also worth mentioning the emergence of techniques that had gone unnoticed until now and are beginning to gain momentum for the next size subgroup (of large-size nets). These are primarily MRNET and CLR, and their potential contribution to larger networks is becoming apparent.

The final subset for which this comparative study has been conducted consists of networks with a size between 250 and 2,000 genes. The results for the AUPR metric are shown in Table 7.14, and for AUROC, they can be found in Table 7.15. As has been the case in the other scenarios, BEST_MO-GENECI emerges as the top performer for network inference in this case as well.

Table 7.13: Friedman mean rank with Holm's adjusted p values (0.05) for the AUROC metric measured across all techniques in networks with 110 to 250 genes.

AUROC 110-250 genes		
Technique	$Friedman'sRank$	$Holm'sAdj - p$
***BEST_MO-GENECI**	**1.33333**	-
MEDIAN_MO-GENECI	4.1667	2.85168e-01
PCACMI	5.2500	2.79120e-01
GENIE3_ET	7.0000	1.20057e-01
GRNBOOST2	7.2500	1.20057e-01
CLR	7.3333	1.20057e-01
GENIE3_RF	7.5000	1.20057e-01
RSNET	7.8333	9.94704e-02
MRNETB	8.1667	7.95802e-02
MRNET	8.4167	6.78703e-02
PUC	10.0000	1.07847e-02
PIDC	10.0833	1.06108e-02
INFERELATOR	14.1667	1.55065e-05
LEAP	14.4167	1.04061e-05
PLSNET	14.9167	4.19078e-06
BC3NET	15.7917	7.39047e-07
ARACNE	16.3333	2.44677e-07
C3NET	17.2500	3.27207e-08
KBOOST	17.4167	2.3472e-08
NARROMI	18.7917	8.60938e-10
MEOMI	18.8333	8.15172e-10
PCIT	20.7500	5.04342e-12

Moreover, as anticipated for the previous subset, CLR and MRNET seem to yield good results for larger networks. These techniques are the only ones that surpass the significance threshold for both tables. Furthermore, by using the small group of networks with more than 2,000 genes as an extension of the current subset, it was observed in Table 7.7 how these techniques consistently achieved good results. In fact, they are highlighted in bold due to their high accuracy on a couple of occasions.

However, it should be remembered that CLR obtained a rather unfavorable position for the subset of small networks, and MRNET also did not excel until reaching larger network sizes (above 110 genes). Therefore, while they do not outperform BEST_MO-GENECI in this scenario, they also do not come close to covering the broad spectrum of domains that BEST_MO-GENECI excels in.

The fact that some techniques seem to exhibit a clear relationship between their performance and network size is a strong indication that size is a determin-

Table 7.14: Friedman mean rank with Holm's adjusted p values (0.05) for the AUPR metric measured across all techniques in networks with 250 to 2,000 genes.

AUPR 250-2000 genes		
Technique	$Friedman's Rank$	$Holm's Adj - p$
***BEST_MO-GENECI**	**2.90323**	-
CLR	3.8065	4.44083e-01
ARACNE	4.2903	4.44083e-01
MRNET	4.8387	2.65207e-01
MEDIAN_MO-GENECI	6.1936	1.50890e-02
BC3NET	6.3548	1.42610e-02
C3NET	6.3548	1.42610e-02
RSNET	6.8065	4.13002e-03
MRNETB	7.9032	8.59231e-05
PIDC	9.9355	5.38791e-09
NARROMI	10.1290	2.00219e-09
PUC	10.5484	1.86184e-10
LEAP	11.8065	5.49889e-14
KBOOST	13.1613	2.21967e-18
PCIT	14.9677	3.33603e-25

Table 7.15: Friedman mean rank with Holm's adjusted p values (0.05) for the AUROC metric measured across all techniques in networks with 250 to 2,000 genes.

AUROC 250-2000 genes		
Technique	$Friedman's Rank$	$Holm's Adj - p$
***BEST_MO-GENECI**	**2.51613**	-
MRNET	3.5161	4.44083e-01
MEDIAN_MO-GENECI	3.9032	4.44083e-01
CLR	4.3548	3.16541e-01
MRNETB	4.7097	2.18862e-01
RSNET	4.8065	2.18862e-01
PUC	7.2258	2.02893e-04
PIDC	7.4839	8.56653e-05
ARACNE	8.8387	2.08492e-07
BC3NET	10.4516	2.54677e-11
C3NET	10.5806	1.25188e-11
LEAP	11.5161	2.54894e-14
KBOOST	12.0968	4.00058e-16
NARROMI	13.2903	3.15211e-20
PCIT	14.7097	9.81590e-26

ing factor in specifying their domains of specialization. This does not mean that there are no other factors at play or that there are no techniques whose domains

are independent of network size. The way the networks have been grouped has allowed us to observe how techniques behave with respect to this specific factor. After careful analysis, it has been observed that some techniques clearly depend on network size, while others do not show a discernible pattern or trend related to this factor.

After statistically analyzing the results of this study, it can be rigorously confirmed that BEST_MO-GENECI provides the highest reliability when aiming for high-precision results, regardless of network size or specialization domain. Furthermore, it is worth mentioning that in cases where a poor choice is made on the front, MEDIAN_MO-GENECI has crossed the significance threshold in a total of 6 scenarios, falling just below it in the remaining two cases.

This study has also made it possible to observe techniques such as PCIT, MEOMI, NARROMI, or KBOOST, which consistently yield low-precision results. Researchers may consider excluding these techniques when using the `MO-GENECI` approach, which could help eliminate noise and further enhance the evolutionary algorithm's results.

7.3.4 Computational Complexity

It is evident that the complex and sophisticated strategy designed to optimally harmonize such a multitude of techniques incurs a computational cost equivalent to its precision. This computational cost depends on several factors. The first and most determining factor is the number of genes involved in the gene regulatory network that is intended to be inferred. Concerning this factor, Figure 7.13 presents the execution times in minutes required by `MO-GENECI` to achieve 250,000 evaluations in each problem, ordered by their size. These times were recorded on a machine with 500GB of RAM and 32 cores. The execution time is clearly exponential with respect to the size of the networks, starting to require significant time from 1,000 nodes onwards. After analyzing certain executions with JProfiler [263], it has been noted that the most costly part and the one that most contributes to the slowdown of the algorithm in large networks is the objective function that counts different motifs. This is to be expected; however, the improvement in precision observed compared to the previous `GENECI` proposal makes this increase in the algorithm's computational cost worthwhile.

Additionally, there is the factor of the number of evaluations. In the experimentation of this chapter, we have set a total of 250,000 evaluations for all problems. However, this quantity was set on the high side, and it has been observed that in different problems, the algorithm achieves convergence much earlier. The point at which convergence occurs for a problem does not depend

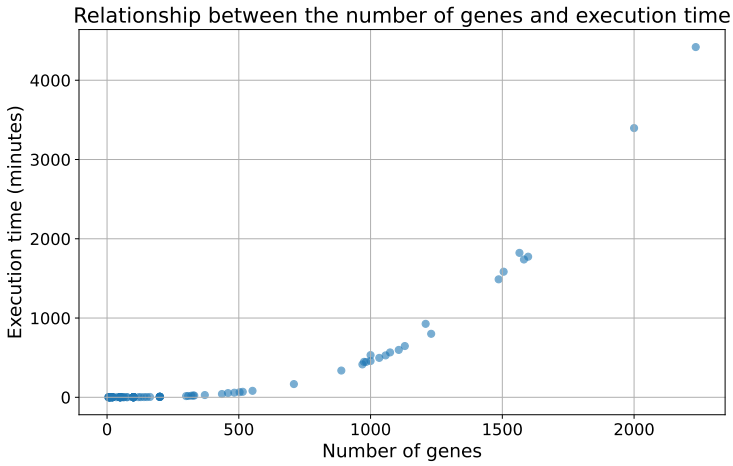


Figure 7.13: Scatter plot illustrating the relationship between the number of genes and the execution time of the algorithm. Each point represents a specific execution instance. The plot provides insight into the performance characteristics of the algorithm across different gene counts.

so much on the size of the network but rather on the coherence among the various techniques and their quantity. Therefore, these last two variables are also indirect factors that affect the execution time.

Finally, it should be remembered that `MO-GENECI` runs after collecting the results from all individual techniques. This means that comparing the computational cost of these individual techniques to this proposal is meaningless since they operate at different levels and have different purposes. The complexity of the individual techniques is relative to the task of inferring gene regulatory networks from expression data, while the complexity of `MO-GENECI` corresponds to the intelligent and optimized process of harmonizing a broad spectrum of inference techniques based on fitness functions specific to the biological domain to which this challenge belongs.

Regarding the algorithm's asymptotic time complexity (Big-O), an implementation-informed approximation is provided below:

- *Preprocessing:* run the k base methods, total $C_{\text{base}} = \sum_{i=1}^{k} C_i$.

- *Evolutionary optimization (over E evaluations):* per evaluation, (A) consensus construction $O(m \cdot k)$; (B) objectives: *Quality* $O(m \cdot k + m)$, *Degree distribution* $O(m + n \log n)$, and *Motifs* (on the binarized network) up to

$O(n^3)$ in the worst case. Summing per-evaluation:

$$O(m{\cdot}k){+}O(m{\cdot}k{+}m){+}O(m{+}n\log n){+}O(n^3) \;=\; O(m{\cdot}k{+}m{+}n\log n{+}n^3).$$

- *Overall:*

$$\boxed{C_{\text{base}} \;+\; C_{\text{MO-GENECI}}} \;\approx\; C_{\text{base}} + O\Big(E\,(m\cdot k + n^3)\Big).$$

- *Symbols:* n = genes; m = candidate interactions; k = base methods; $E \approx T{\cdot}P$ with T generations and population size P; C_{base} = total cost of running all base methods once.

7.3.5 Real-World Experimentation

After validating its efficacy, `MO-GENECI` has also been run on real-world gene expression data. Specifically, clinical data from melanoma patients were used. This data included gene expression levels collected from NanoString, specifically from the platform's immunological profiling panel. This panel was subjected to specific treatment and filtering techniques, which eliminated 35 genes showing less stable results.

This dataset was also employed in `GENECI`, and the approach taken then was to contrast the interactions with the highest confidence level of the solution with respect to the literature. However, in this case, since `MO-GENECI` is a multi-objective algorithm, we do not have a single solution, but a front of solutions. Therefore, the strategy used in this case has been to collect the most frequent and most trusted interactions taking into account their position after sorting each network by confidence level.

After carrying out this procedure, the following top 5 interactions were obtained. For each of them, both genes were entered into the STRING database [264] and the Co-Mentioned in Pubmed Abstracts section was accessed to obtain a list of publications. This list was subsequently filtered to find articles related to melanoma, cancer in general, or skin diseases.

1. **IL8 - CCL2**: In [265], the role of IL-8 as a potential predictive biomarker in head and neck cancer is studied. In their study, after analyzing patients before and after radiation therapy, it is shown that there is a strong relationship between IL-8 and CCL2 (MCP-1), which is statistically verified by Pearson's correlation coefficients. In addition, [266] establishes that CXCL8 (IL-8), CCL2, and CCL5 are three key chemokines in the invasion of tumor cells, providing evidence of their interactions through numerous experiments.

2. **OSM - IL8**: In [267], Oncostatin M (OSM) is shown to enhance skin squamous cell carcinoma (20% of skin cancer deaths) and that its expression in tumor lesions strongly correlates with that of the well-known neutrophil chemotactic factor IL-8. This relationship has been rigorously confirmed through experimentation and Spearman rank correlation calculation. Additionally, [268] studies the effect of ulipristal acetate (UPA) in the treatment of endometrial cancer. For this, the expression levels of the proinflammatory cytokines OSM, IL-6, and IL-8 were examined using quantitative real-time PCR, revealing a clear importance of the expression levels and interaction of these gene products in cancer.

3. **IL1R2 - ARG1**: This interaction was already validated in the `GENECI` publication, as it was the most reliable regulation reported by the preliminary algorithm of this project. Specifically, it was highlighted that several studies relate both genes within the context of cancer [233, 234]. Specifically, in [235] it is concluded that the amebiasis pathway could be involved in melanoma metastasis through these genes.

4. **IL8 - OSM**: This concerns the inverse regulation of interaction 2, which points to a coregulation between both genes. The justification is the same as that for the inverse regulation, as validation through literature search does not provide a sufficient level of precision to determine the direction of the regulation.

5. **NFKBIA - TNFAIP3**: In [269], the role of KLF6 in gene regulation is analyzed, focusing on patients with glioblastoma. During their study, they demonstrate that NFKBIA and TNFAIP3 are the most potent negative regulators of NF-κB induced by KLF6. The findings of the previous article coincide with what was also demonstrated for colon cancer in [270]. It shows that once again NFKBIA and TNFAIP3 are two important feedback loops of NF-κB, and that samples from patients with colorectal cancer (n = 626) show much lower gene expression of NFKBIA (0.643 times) and TNFAIP3 (0.745 times) compared to healthy controls (n = 51) according to the TCGA database. Finally, in [271], this association is also confirmed for breast cancer. Therefore, despite the lack of specific studies for skin cancer, it suggests that this interaction is likely to play a relevant role in melanoma as well.

From these, only one appeared in the top 3 provided by `GENECI` in the previous chapter. The other two are also in high positions but not notable, specifically IL18R1 - IL1RL1 is in seventh position and HLA-DQA1 - HLA-DQB1 in twelfth. This means that `MO-GENECI` continues to detect the most important interactions

provided by GENECI but adds the inference of new interactions that have also been shown to be supported by the literature.

Chapter 8

PBEvoGen: Guiding GRN inference with expert-driven preference articulation

The results of the previous proposal called MO-GENECI (see chapter 7) have shown that the best solutions tend to concentrate on specific regions of the objective space, which can be anticipated based on known properties, such as network size (see Figure 7.8).

Based on this observation, this chapter presents an approach that introduces preference-based selection [272], allowing a domain expert to define a reference point in the objective space to guide the evolutionary search. This strategy, previously applied in other fields [273, 274], is presented here for the first time in the context of GRN inference, where a new strategy regarding the selection of reference points is included, according to the characteristics of the networks to be inferred. The proposal, called PBEvoGen, is evaluated in this chapter in order to address the following research questions:

RQ1: What impact can the use of this methodology have on the biological accuracy of the inferred networks?

RQ2: Are preference zones beneficial and detectable by domain experts?

RQ3: Can integrating this selection process reduce execution costs without compromising network quality?

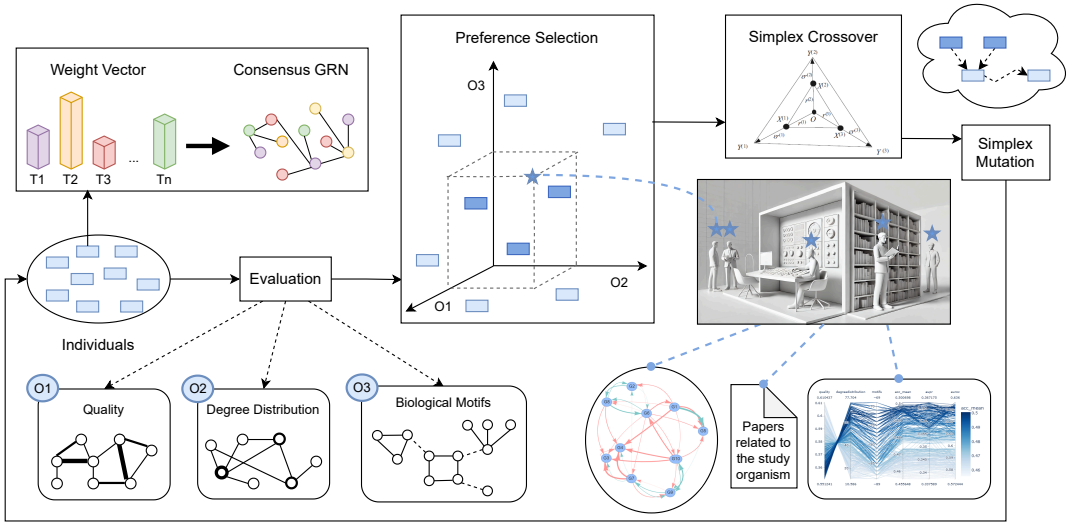


Figure 8.1: Conceptual diagram of the proposed multi-objective evolutionary algorithm. Individuals, represented as weight vectors, are converted into their respective consensus regulatory networks for subsequent evaluation. Three objectives are considered: Quality, Degree Distribution, and Motifs. Once the performance of each individual is obtained for these objectives, selection is carried out based on the reference point established by the domain expert. The expert draws from various sources and personal experience to guide the population of individuals toward a region of interest in the problem. Afterward, the selected individuals undergo crossover and mutation to form a new generation of individuals.

8.1 Proposed architecture and expert interaction

PBEvoGen is based on the multi-objective evolutionary algorithm developed in chapter 7, which is originally designed for the optimization of consensus networks in GRN inference. The incorporation of preference-based selection is illustrated in the conceptual diagram of Figure 8.1, while `PBEvoGen` implementation, based on NSGA-II, is outlined in the pseudocode of Algorithm 10. This implementation is specifically tailored to the problem through a weight vector representation, customized crossover and mutation operators, and fitness functions specifically designed to address the challenges of the biological context.

The main flow of the algorithm begins with the generation of an initial random population (line 1 in Algorithm 10), represented as weight vectors that sum up to 1. Each individual represents a weighted voting system for the selection of inference techniques, among multiple of them, hence allowing the transformation of their weights into a consensus network. Concretely, a num-

Algorithm 10 PBEvoGen Algorithm.

Require: Num of generations T, Population size P, Crossover operator $x_{simplex}$, Mutation operator $m_{simplex}$, Reference point R
Ensure: Pareto-optimal front PF
1: $P \leftarrow$ generate_random_population(P)
2: $E \leftarrow$ evaluate_population(P, F)
3: $t \leftarrow 1$
4: **while** $t < T$ **do**
5: $ranks \leftarrow$ rank_population(E)
6: $g\_dominance \leftarrow$ compute_g_dominance(E, R)
7: $selected \leftarrow$ select_population$(P, ranks, g_dominance)$
8: $offspring \leftarrow$ crossover$(selected, x_{simplex})$
9: $offspring \leftarrow$ mutate$(offspring, m_{simplex})$
10: $P \leftarrow$ replace_population$(P, offspring)$
11: $E \leftarrow$ evaluate_population(P, F)
12: $t \leftarrow t + 1$
13: **end while**
14: $PF \leftarrow$ get_pareto_front(E)
15: **return** PF

ber of 26 inference techniques taken from the current state of the art are used (described in section 3.1.1): ARACNE [96], BC3NET [126], C3NET [127], CLR [94], GENIE3_RF [9], GRNBOOST2 [129], GENIE3_ET [9], MRNET [138], MRNETB [139], PCIT [143], TIGRESS [97], KBOOST [132], MEOMI [135], JUMP3 [50], NARROMI [140], CMI2NI [95], RSNET [146], PCACMI [11], LOCPCACMI [134], PLSNET [145], PIDC [144], PUC [144], GRNVBEM [131], LEAP [133], NONLINEARODES [98] and INFERELATOR [10]. Therefore, this solution coding allows the algorithm to act as an ensemble capable of generating consensus genetic regulatory networks, oriented to reinforce those topologies that are more frequent and robust.

Based on this representation, individuals are evaluated (lines 2 and 11 in Algorithm 10) using three conflicting objectives:

1. **Quality**: This objective favors networks where a subset of interactions exhibits high confidence levels derived from a consistent weight distribution across techniques. Interactions that do not show agreement between techniques are penalized, encouraging more reliable networks.

2. **Degree Distribution**: This objective aims for networks with degree distributions that follow a power-law, a typical characteristic of biological net-

works. This criterion favors structures where most nodes have few connections, but a few act as highly connected hubs.

3. **Motifs**: This objective promotes networks that display structural patterns characteristic of regulatory networks, such as bifurcations, feedback loops, and regulatory pathways. The detection of these motifs reinforces the biological functionality of the generated networks.

The original selection process, based on binary tournament selection, has been replaced in this proposal by a preference-based selection mechanism (lines 5 to 7 in Algorithm 10). Concretely, we have adopted the reference-point method [275], which constitutes a simple way to delimit an interest region of the objective space by indicating a user-defined point. We have used the scheme existing in jMetal [276], based on the g-dominance concept [277].

The algorithm employs customized operators to ensure the feasibility of solutions. The crossover operator is based on the Simplex Crossover [253], which generates new individuals within feasible regions by combining multiple parents (line 8 in Algorithm 10). On the other hand, the mutation operator, called Simplex Mutation (see section 7.1.3), applies negative perturbations to subsets of the weight vector and redistributes those values to other subsets, maintaining the normalization of the vector (line 9 in Algorithm 10).

Thanks to the architecture finally built, the inclusion of domain knowledge in GRN consensus inference is no longer limited to the algorithm's objectives, but is now complemented by even more precise guidelines in the selection phase.

The specialization of the objectives in this algorithm within the context of biological networks makes the search space more understandable for domain experts, who can anticipate certain topological features or the presence of expected regulatory motifs. These predictions can be based on literature reviews of the organism under study, analyses of networks from similar organisms, results from experiments on simulated datasets, or knowledge of previously validated interactions. All this information can now be utilized during the algorithm's execution, guiding the search towards solutions with greater biological relevance instead of being limited to the final selection on the approximated Pareto front.

8.2 Experimentation

The experimentation in this study utilizes a set of 43 problem instances comprising gene regulatory networks with sizes of up to 370 genes. This set includes 15 networks from the DREAM3 challenge [183], 10 networks from the DREAM4

challenge [142], 12 synthetic networks generated from scratch by the SysGenSIM simulator [179] with both scale-free and EIPO modular distributions [195] ranging in size from 20 to 50 nodes, 2 instances of the yeast network from IRMA [173], and 4 additional real networks collected by TFLink [174], whose expression data were generated again using the SysGenSIM simulator. For more details on these data, see section 4.1.

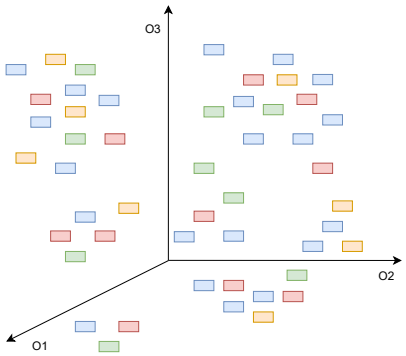
For each of these 43 instances, the phase-organized workflow shown in Figure 8.2 was executed, maintaining the same parameter values that were properly justified in the previous chapter (see section 7.2.1). First, each gene expression dataset was subjected to 15 independent runs of `MO-GENECI` (without preference articulation mechanims) to subsequently extract an initial reference front (Figure 8.2a). The resulting Pareto approximation front, already filtered to include only non-dominated solutions, was evaluated using the AUROC and AUPR accuracy metrics, which compare the consensus networks of the individuals with the corresponding gold standard of each instance (Figure 8.2b).

The selection of the reference point by a domain expert was, in this case, approximated by choosing points in the objective space close to high-accuracy solutions. To make this approximation more rigorous, different points were considered for each metric. This approach allows for the observation of whether setting reference points in regions of high AUPR leads to specific improvements in that metric, while setting them in regions of high AUROC results in improvements exclusively for the latter metric.

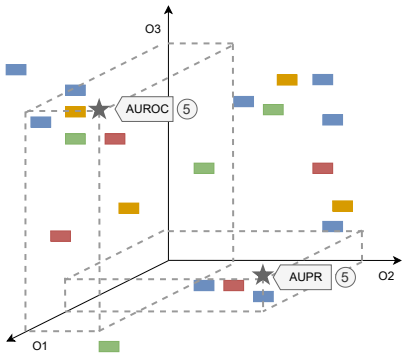
The objective is to demonstrate that the preference-based selection introduced in this chapter influences the algorithm's evolution, not only by guiding individuals towards the reference point in the objective space (spatial improvement), but also by indirectly enhancing the inference accuracy or, equivalently, the biological relevance of the solutions associated with the selected region (accuracy improvement).

To expand the experimental study, several configurations were considered for each metric used as a reference. For both AUPR and AUROC, reference points were established by calculating the maximum coordinates of the top 5, 10, and 20 best solutions from the filtered reference approximation front.

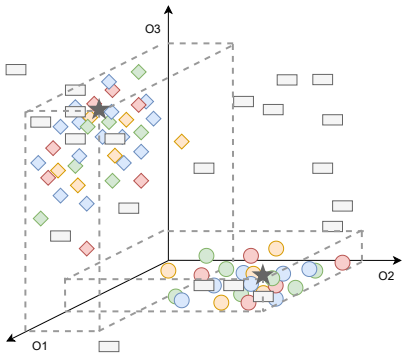
In total, six reference points were tested for each instance. Fifteen independent runs with preference-based selection were performed for each reference point (Figure 8.2c). Finally, the filtered front associated with each point was extracted and evaluated again using the AUROC and AUPR metrics (Figure 8.2d), enabling a rigorous algorithmic comparison between the different configurations and the original algorithm.



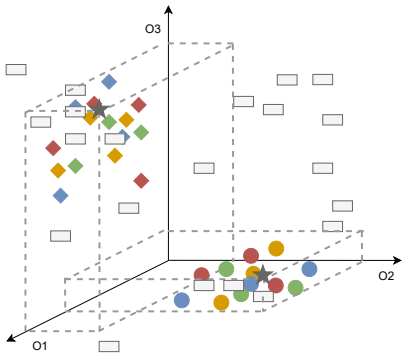
(a) **Phase 1**: 15 independent runs of `MO-GENECI`. Each rectangle represents an individual, and each color is used to represent a different independent run.



(b) **Phase 2**: Filtering of non-dominated individuals from the previous phase, validation using accuracy metrics (AUROC and AUPR), and establishment of reference points. For visual simplification, only the reference points corresponding to the top 5 individuals for each metric have been represented.



(c) **Phase 3**: 15 independent runs of the modified algorithm with preference-based selection per reference point. Original individuals appear in gray; new ones are shown as colored circles (AUPR) and diamonds (AUROC).



(d) **Phase 4**: Filtering of non-dominated individuals from the previous phase and validation using accuracy metrics (AUROC and AUPR) to enable subsequent comparison of the different populations.

Figure 8.2: Phases carried out during the experimentation of this study, illustrating the workflow from multiple independent runs and filtering of non-dominated individuals, through validation with AUROC and AUPR, to the generation and comparison of refined populations.

The algorithmic comparison focuses primarily on MO-GENECI because this approach has already been shown to statistically significantly outperform a broad set of well-established techniques in gene regulatory network inference (see section 7.3.3). Therefore, if PBEvoGen achieves better results than MO-GENECI, comparing it to these other 26 techniques becomes redundant, as its dominance would be implicitly established.

This comparison is performed using a Friedman statistical ranking with Holm's non-parametric tests for each metric [255]. Once the best configuration for AUPR and the best for AUROC are identified, and both spatial and accuracy improvements are demonstrated, an additional experimental phase (Phase 5) is conducted to validate the performance improvement offered by this proposal.

This phase consists of five additional runs for each winning configuration associated with each metric and for each of the largest networks in the benchmark ($\geq$ 100 genes). A custom observer is employed, which takes the filtered reference front (obtained by MO-GENECI, which does not uses preference-based selection mechanism) as input, restricts it to the search region defined by the reference point, and calculates in each generation the percentage of solutions from the original front that are dominated by the current population. This approach allows identifying the point at which the execution of the proposed method could be terminated, while maintaining the same optimization level as the original one, providing an estimate of how much execution time could be reduced for large networks.

8.3 Results and discussion

Following the execution of the experimentation described in the previous section, the results obtained are presented from different perspectives and through several representations to justify and demonstrate each of the three improvements pursued in this chapter: solution quality improvement, spatial improvement, and performance improvement.

8.3.1 Solution quality improvement

The completion of the fourth phase of experimentation across the entire benchmark (see Figure 8.2d) leads to the calculation of the Friedman statistical ranking with Holm's non-parametric tests for each metric, aiming to compare the original version of the algorithm with the different configurations of the proposed approach in this chapter.

Table 8.1: Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.

AUPR		
Technique	$Friedman'sRank$	$Holm'sAdj - p$
PBEvoGen AUPR-10	**2.0465**	-
PBEvoGen AUPR-5	2.3023	0.3582
PBEvoGen AUPR-20	2.7442	0.0244
MO-GENECI	2.9070	0.0060

Table 8.2: Friedman mean rank with Holm's adjusted p values (0.05) for AUROC.

AUROC		
Technique	$Friedman'sRank$	$Holm'sAdj - p$
PBEvoGen AUROC-10	**2.2791**	-
PBEvoGen AUROC-20	2.3372	1.0080
PBEvoGen AUROC-5	2.4651	1.0080
MO-GENECI	2.9186	0.0649

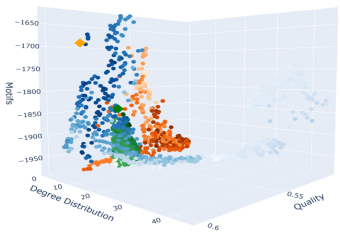
The results of this ranking for the AUPR and AUROC metrics are presented in Table 8.1 and Table 8.2, respectively. As observed, in both cases, the original algorithm ranks last, below all configurations of the proposed approach. This underperformance is accompanied by clear statistical significance for the AUPR metric, while for AUROC, the p-value is close to the commonly used threshold of 0.05. These results not only demonstrate a superiority in accuracy over `MO-GENECI` but also over the 26 inference techniques that the original proposal had already proven to outperform.

Answer to RQ1

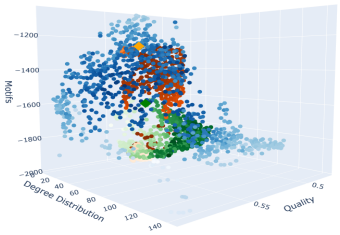
The first experimental results confirm that the proposed methodology enhances result quality, aligning with the rationale behind the accurate positioning of reference points in the 3D objective space, specifically improving the metric used for point selection.

Regarding the different configurations, it can be seen that the one considering the top 10 individuals (from the perspective of the respective metric) achieves the first position in both rankings. This outcome is consistent and reinforces the hypothesis that an intermediate distance is appropriate when establishing reference points. Specifically, a point that is too close ends up discarding other high-quality solutions, whereas a point that is too distant excessively broadens the significant search area, allowing non-relevant solutions.

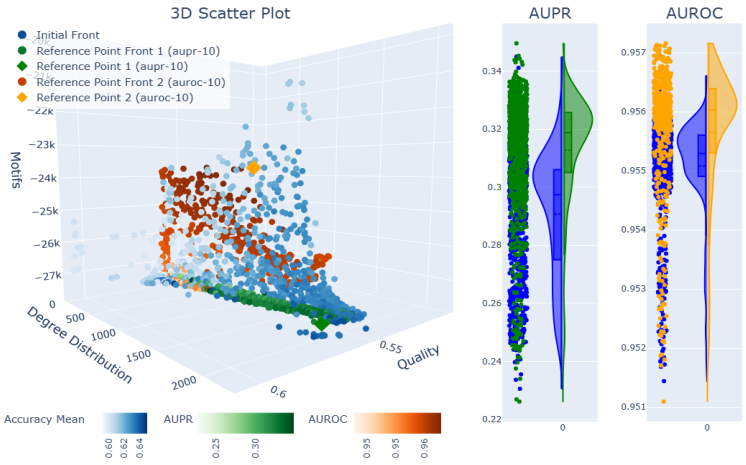
Identifying the top 10 individuals as the winning configuration enables a



(a) Second 100-node network of the *Ecoli* organism in the DREAM3 challenge.



(b) Fifth 100-node network of the DREAM4 challenge.



(c) Representation of the three populations for the *Saccharomyces cerevisiae* network from the TFLink database. In addition to what is presented in Figure 8.3a and Figure 8.3b, this case includes the legend of the three-dimensional plot and two additional violin plots corresponding to the AUPR and AUROC metrics.

Figure 8.3: Graphical representation for different gene regulatory networks of the three most relevant populations in the study. Specifically, the population of non-dominated solutions from the 15 independent runs of: (1) the original algorithm, colored based on the average of the AUPR and AUROC metrics; (2) the algorithm with preference-based selection guided by the point corresponding to the top 10 individuals with the best AUPR (green diamond), colored based on AUPR values; and (3) the algorithm with preference-based selection guided by the point corresponding to the top 10 individuals with the best AUROC (orange diamond), colored based on AUROC values.

more focused analysis in the following sections to address the two remaining improvements to be demonstrated.

8.3.2 Spatial improvement

Once the solution quality improvement has been demonstrated, it is necessary to verify whether this increase in quality is due to the proper functioning of the preference-based selection process. Figure 8.3 displays objective space generated by individuals of the population of `MO-GENECI` and those of the two winning configurations of `PBEvoGen` (one for each metric) for several GRN benckmarking instances. Several observations can be drawn from the three-dimensional plots:

- The color gradient present in the original population across all cases demonstrates that both, the biological context and the original algorithm are well-

suited for implementing the preference-based selection explored in this study. The fact that spatially close solutions exhibit similar quality metric levels makes restricting the search to specific regions an interpretable strategy for domain experts.

- The spatial positioning of the solutions guided by a reference point respects the boundaries defined by its coordinates, confirming the correct functioning of the selection process.
- The individuals generated by the algorithm with preference-based selection outperform those of the original algorithm, achieving a higher level of optimization when executed with the same number of evaluations.

Answers to RQ2

The graphical representations in this section clearly illustrate a correlation between the quality of the networks and the individuals' locations in the search space. This correlation indicates that the search constraints in specific areas are aligned with neighborhoods where individuals share similar qualities related to inference accuracy, extending beyond just the objectives. Additionally, due to the biological profile of the algorithm's objectives, selecting reference points becomes a task that domain experts can easily understand.

Additionally, for the *Saccharomyces cerevisiae* network from the TFLink database, which is the largest network in the benchmark with 370 genes, two additional violin plots have been included to represent the AUROC and AUPR values for the individuals in each population. These plots further reinforce the findings from the previous section: the solutions of the algorithm with preference-based selection are influenced not only spatially by the reference point but also by its meaning, biological relevance, and quality of solutions. This aspect is crucial for making the injection of expert knowledge truly effective.

8.3.3 Performance improvement

In the previous section, it was observed that the solutions obtained by `PBEvoGen` outperform those of `MO-GENECI`. However, identifying the point during execution when this occurs in large networks would allow for estimating the potential computational savings if the same level of optimization is to be maintained.

In the previous chapter, it is noted that execution for large networks can extend to nearly three days (see Figure 7.13). Therefore, if the region of interest within the search space is known, it would be valuable to verify whether the

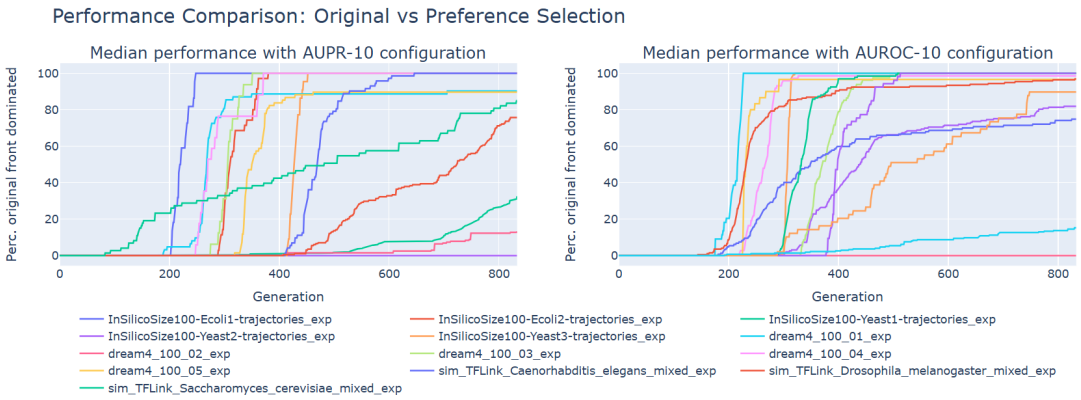


Figure 8.4: Median of the 5 independent runs of the percentage of dominated solutions from the original front (trimmed by the reference point) in each generation for large networks ($\geq$ 100 nodes) using the winning configurations (one for each metric).

proposed approach in this chapter could reduce execution time due to its ability to lower the algorithm's exploration effort.

Figure 8.4 shows that in both cases, more than 50% of the networks dominate nearly the entire original front by the midpoint of the evolutionary process. This indicates that with only half the evaluations, a similar level of optimization to that of the original algorithm is achieved, allowing for a highly significant reduction in execution time for these cases. This is a remarkable result when, as mentioned above, implementation times are in the order of days.

Answers to RQ3

By adopting the preference-based approach, computational costs for large networks can be reduced by up to half of the original computation times while still achieving solution fronts with similar accuracy.

Chapter 9

BIO-INSIGHT: Maximizing biological coverage through many-objective optimization

Despite the innovations proposed in MO-GENECI (see chapter 7), the biological complexity of GRNs requires greater attention and coverage beyond the degree distribution of their nodes and the enumeration of a predefined set of motifs. This expansion of knowledge must be carried out while maintaining the balance of opposing objectives, ensuring proper alignment with inference accuracy, and preserving computational feasibility for large-scale networks.

To overcome these challenges, this chapter introduces **BIO-INSIGHT (Biologically Informed Optimizer - INtegrating Software to Infer GRNs by Holistic Thinking)**, an innovative algorithmic approach that brings three main contributions to the current state of the art:

- **Objective space with high biological coverage**: Beyond the aspects already explored in the current literature, this proposal incorporates three main novelties: (1) the study of the structural influence of genes within the network, (2) network dynamism and stability under perturbations, and (3) analysis of the gene regulatory system to reduce non-essential interactions caused by weighting mechanisms that tend to excessively encourage regulatory redundancies.
- **Novel architecture adapted to high computational costs**: A new asynchronous and parallel evolutionary model is introduced, enabling simultaneous evaluations even across different generations. Additionally, objectives derived from previous proposals in the literature have been refactored

to eliminate unnecessary intermediate conversions and implement internal caching systems to avoid redundant computations. This, along with other refactorizations and technical strategies discussed in later sections, helps to partially offset the computational complexity introduced by adding three high-cost dimensions. As a result, the approach maintains computational feasibility for network sizes comparable to those addressed in the closest related literature, even though BIO-INSIGHT operates in an objective space twice as large.

- **Biological validation and clinical relevance of the model**: In contrast to approaches focused solely on predictive performance or algorithmic design, BIO-INSIGHT has been evaluated on real-world gene expression data from fibromyalgia and myalgic encephalomyelitis patients, demonstrating its ability to identify biologically meaningful and reproducible gene interactions. The inferred networks enabled the distinction between clinically overlapping conditions, revealed regulatory alterations consistently absent in disease groups, and uncovered condition-specific interactions with potential biomarker value. Several of these findings are supported by existing literature or experimental validation, reinforcing the usefulness of BIO-INSIGHT as a tool for uncovering disease mechanisms in complex conditions with no validated biomarkers.

In addition to these main contributions, this chapter addresses a series of research questions (RQs) that support and justify the scientific contribution of this proposal:

RQ1: Is there sufficient disparity in the networks inferred by different high-precision techniques to justify the motivation for developing new consensus methods?

RQ2: Does the consideration of multiple biological aspects still maintain the opposition of objectives at the evolutionary core of the proposal?

RQ3: Is there a correlation between the individual accuracy of inference techniques and their contribution to the optimal consensus in BIO-INSIGHT?

RQ4: Do the networks inferred by BIO-INSIGHT exhibit a biologically coherent structure in relation to known GRNs?

RQ5: Is the optimization of BIO-INSIGHT's high-biological-coverage objective space properly aligned with improving the quality of GRN inference?

RQ6: Is BIO-INSIGHT's optimization process redundant concerning the knowledge already acquired by networks during their inference through individ-

ual techniques?

9.1 Algorithmic Proposal

BIO-INSIGHT (Biologically Informed Optimizer – INtegrating Software to Infer GRNs by Holistic Thinking) introduces a novel and intelligent consensus system for GRN inference, guided by the biological context of the data. This biologically informed strategy enhances both the interpretability and reliability of the inferred gene regulatory networks, making it a major advancement over traditional approaches.

In addition to this core innovation, `BIO-INSIGHT` offers a comprehensive suite of complementary tools: interactive visualization modules, centralized access to datasets from well-known challenges, academic benchmark generation through multiple simulators, domain-specific validation metrics, and support for up to 26 well-established inference techniques.

To gain a more detailed understanding of the functionality and characteristics of this system, Figure 9.1 presents a diagram illustrating the main workflow of `BIO-INSIGHT`. This diagram corresponds to the primary command of the tool (run) and takes as input the expression dataset from which the GRN is to be inferred. The set of available techniques for the initial inference phase includes: ARACNE [96], BC3NET [126], C3NET [127], CLR [94], GENIE3_RF [9], GRNBOOST2 [129], GENIE3_ET [9], MRNET [138], MRNETB [139], PCIT [143], TIGRESS [97], KBOOST [132], MEOMI [135], JUMP3 [50], NARROMI [140], CMI2NI [95], RSNET [146], PCACMI [11], LOCPCACMI [134], PLSNET [145], PIDC [144], PUC [144], GRNVBEM [131], LEAP [133], NONLINEARODES [98] and INFERELATOR [10]. All of these techniques were described in section 3.1.1.

Gene regulatory networks are, by nature, directed graphs, as they represent causal relationships between regulators and target genes. However, since some of the techniques integrated into `BIO-INSIGHT` produce undirected graphs while others infer directionality, this property has been deliberately omitted in the consensus process. This decision enables the unification of the output representations across different methods, avoiding structural incompatibilities and ensuring that no technique is excluded based on the type of graph it generates. At the same time, it facilitates the future incorporation of new tools, regardless of whether they infer directed or undirected networks.

Designing an architecture adapted to large-scale networks requires optimizing the utilization of available resources to minimize execution time. To achieve this, in the initial phase, inference techniques are executed in parallel within

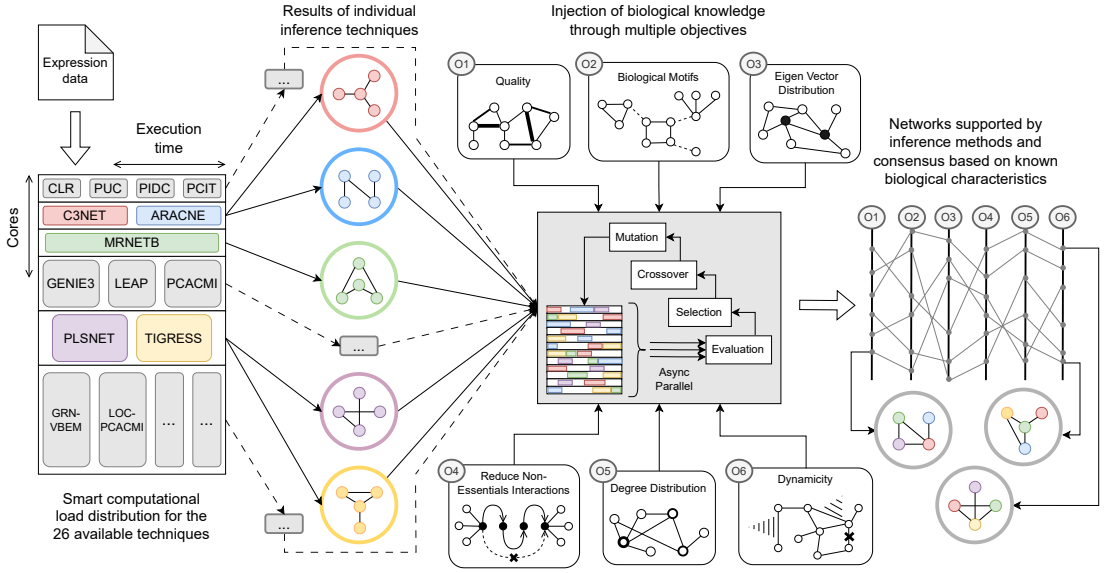


Figure 9.1: Standard workflow of BIO-INSIGHT. Starting from gene expression data, the 26 available inference techniques are executed in parallel, with computational load distributed based on their expected cost (left). Each technique infers a different gene regulatory network, producing a set of individual solutions (center-left). These networks are integrated through an asynchronous and parallel many-objective evolutionary algorithm, which optimizes a weighted voting system using six biologically driven objectives: interaction quality (O1), presence of biological motifs (O2), eigenvector centrality distribution (O3), reduction of non-essential interactions (O4), node degree distribution (O5), and dynamic stability (O6). Each individual in the population represents a set of weights for the inference techniques and is evaluated according to how well its resulting consensus network satisfies the biological objectives (center). The evolutionary process generates a Pareto front of optimal trade-offs, where each solution corresponds to a consensus network that balances different biological properties. These final networks (right) are supported both by the inference techniques and by known structural characteristics of real-world gene regulatory systems.

multiple Docker containers, each dynamically assigned a specific amount of resources based on prior computational costs and adapted to the available resources in the environment.

The results from these individual techniques are then provided as input to the algorithmic core of BIO-INSIGHT (see Algorithm 11). This asynchronous and parallel many-objective evolutionary algorithm optimizes a weighted voting system among different machine-learning techniques based on a broad set of biologically driven objectives. This algorithm, built upon MO-GENECI (chapter 7) and consequently, NSGA-II [79], has been implemented in Java using the JMetal

Algorithm 11 Evolutionary optimization of the consensus in BIO-INSIGHT

Require: Confidence matrices from inference techniques M
Ensure: Pareto front of consensus networks F
1: $P \leftarrow$ InitializePopulation()
2: **while** not TerminationCriterionMet() **do**
3: $O \leftarrow$ GenerateOffspring(P)
4: **for all** $I \in O$ **do**
5: $G \leftarrow$ BuildConsensusNetwork(I,M)
6: Parallel execution:
O1: Quality(I,G) (refactored from chapter 5)
O2: Motif(I,G) (refactored from chapter 7)
O3: EVDist(I,G) (Algorithm 12)
O4: ReduceNEInt(I,G) (Algorithm 13)
O5: DegreeDist(I,G) (retrieved from chapter 7)
O6: Dynamicity(I,G) (Algorithm 14)
7: **end for**
8: $P \leftarrow$ SelectNextGeneration($P \cup O$)
9: **end while**
10: **return** $F \leftarrow$ ExtractParetoFront(P)

framework [276].

Each individual is created as a weight vector (Line 1 in Algorithm 11) whose total sum equals one (simplex). Each vector assigns a decimal value between 0 and 1 to each inference technique to be integrated into the consensus. In every generation (Line 2 in Algorithm 11), individuals undergo crossover and mutation phases, using simplex-compatible operators described in section 7.1, ensuring that offspring remain within the solution space (Line 3 in Algorithm 11). By optimizing the objective functions detailed in the following sections, the algorithm aims to determine the optimal vote distribution to construct the gene regulatory network (Line 5 in Algorithm 11) that best satisfies all objectives (Line 6 in Algorithm 11).

Among the main contributions of this algorithm compared to existing state-of-the-art implementations, the following stand out:

- Refactorization of objectives retrieved from the literature (quality, degree distribution, and motifs), along with the implementation of three new fitness functions (eigenvector distribution, dynamicity, and reduce non-essential interactions).
- Design of an asynchronous and parallel implementation that, through the

simultaneous evaluation of individuals, even from different generations, maintains computational feasibility for large-scale networks.

- Incorporation of an external evaluation file that allows reconsidering the inclusion of prematurely discarded individuals, a practice that has already proven successful in many-objective problems [278].
- Implementation of a concurrent caching system capable of handling simultaneous reads and writes from multiple threads to reduce the number of redundant evaluations. This approach has been particularly useful for objectives where the consensus network must be binarized before computing the fitness value, as different individuals may represent the same binary network.

The output of the algorithm consists of an approximated Pareto front, where each solution represents a potential gene regulatory network inferred through the consensus of all initial techniques (Line 8 in Algorithm 11). This front is supplemented with various graphical representations of the evolutionary process that took place during execution, a parallel coordinates diagram depicting each solution in the front, as well as 2D and 3D plots resulting from grouping different objectives. These, along with the different network visualization tools implemented in the software package, allow domain experts to select the most appropriate solution based on their criteria.

The following subsections provide a detailed description of each objective function considered in BIO-INSIGHT. They include the sources from which their implementations or refactorings were obtained, and for newly created objectives, a comprehensive explanation of their design and implementation is provided.

9.1.1 Objective 1: Quality

This objective was retrieved from the first aggregative term of the proposal in chapter 5, which was later reformulated in chapter 7, simplifying its implementation. This latter implementation was retained in BIO-INSIGHT, as it demonstrated superior performance compared to the original version.

Although it is thoroughly described in section 7.1, it should be mentioned that the purpose of this function is to promote the emergence of networks whose interactions have high confidence levels, provided they are derived from individuals with consistent weight distributions, that is, individuals that assign greater importance to inference techniques whose values are more aligned with those of others.

To achieve this, the Quality function evaluates each interaction's quality based on consensus confidence and distance. The consensus is obtained through a weighted sum of the confidence values assigned by different techniques. The distance measures coherence among these techniques, penalizing significant discrepancies from the median while rewarding alignment with it. Only interactions with a quality score above the mean are used in the final calculation, which is defined as one minus the mean of these quality scores. Thus, this function minimizes this value, favouring networks with more coherent and high-quality interactions.

9.1.2 Objective 2: Motifs

The Motifs function, based on the one described in chapter 7 under the same name, aims to favour consensus networks with a high density of recurrent structural patterns that have been previously identified as characteristic of biological networks: regulatory pathways, differentiation, bifurcation, and coupling. These motifs represent specific interaction configurations between genes and are fundamental for understanding the organization and dynamics of biological systems [46]. The number of detected motifs in the consensus network determines the final score of the function, encouraging networks with biologically relevant structures.

In chapter 7, the individual's weight vector is first converted into a consensus network, which is then binarized to transform it into a directed graph using the JGrapht library [261]. The refactoring performed in `BIO-INSIGHT` optimizes this process by eliminating the intermediate step, allowing the direct conversion of confidence levels into a JGrapht graph without generating the adjacency matrix beforehand in each evaluation. This improvement has significantly reduced RAM consumption, particularly for large-scale networks. Moreover, it has been verified that this refactoring does not alter the results, ensuring that the function keeps producing the same values as in its original implementation.

Additionally, it is worth noting that implementing the concurrent caching system developed in this proposal has been particularly beneficial for this objective function. Although individuals in the population encode different weighted networks, the previously discussed binarization step can cause several of them to converge on the same binary representation, resulting in redundant assessments. Thanks to the cache system, the corresponding evaluations are now efficiently optimized, hence saving processing time and improving the algorithm's overall performance.

Algorithm 12 Third objective: Eigenvector Distribution.

Require: Consensus list with confidence values c
Ensure: Value of the fitness function $result$
1: key = getHashCode(c)
2: **if** $cache.containsKey(key)$ **then**
3: $result = cache.get(key)$
4: **else**
5: $graph$ = getWeightedGraph(c)
6: $eigenvectorScores$ = eigenvectorCentrality($graph$)
7: $result$ = goodnessFitParetoTest($eigenvectorScores$)
8: $cache.put(key, result)$
9: **end if**
10: **return** $result$

9.1.3 Objective 3: Eigen Vector Distribution

This fitness function evaluates the distribution of gene influence within the consensus networks by computing eigenvector centrality [279]. This metric measures the importance of a node in a network, assigning higher values to nodes connected to other highly influential nodes.

It is well known that the topology of biological networks often follows a power-law distribution [44], where a few nodes (genes) act as highly connected hubs, while the majority exhibit only a few connections. This pattern has also been observed for more complex centrality metrics, such as eigenvector centrality [1], where influence tends to be concentrated in a few key nodes, reflecting the typical hierarchical and modular structure of biological networks.

Unlike node degree, which is simply based on the number of interconnections a node has, eigenvector centrality focuses on the significance of these interactions, prioritizing quality over quantity. As a result, Objective 5 and this objective, although both optimizing network topology, focus on different yet complementary aspects. While the former may favour overall structural connectivity, the latter emphasizes the functional importance of nodes within the network. This distinction has been shown to create some trade-offs during optimization: nodes with low degrees can exhibit high relevance due to their connection with key genes, while conversely, nodes with many connections may have limited global influence due to the low relevance of their interactions.

The implementation of this fitness function is outlined in Algorithm 12 and analyzes the consensus network computed from the individual. First, a *hash* key is generated to check whether the evaluation has already been performed and

stored in the cache (line 1 of Algorithm 12). If the network has been previously evaluated, the result is retrieved from the cache (line 3 of Algorithm 12).

Otherwise, a weighted graph is constructed from the consensus network (line 5 of Algorithm 12), and eigenvector centrality is computed for each node using the JGrapht implementation (line 6 of Algorithm 12). Subsequently, these values undergo a goodness-of-fit test for a Pareto distribution (line 7 of Algorithm 12, implemented according to [259]). Finally, the result is stored in the cache and returned as the final value of the fitness function (lines 8 and 9 of Algorithm 12).

9.1.4 Objective 4: Reduce Non-Essentials Interactions

This fitness function evaluates the structure of the consensus networks by analyzing the edge betweenness centrality [280], with the goal of identifying and favoring networks in which non-essential interactions have been discarded. Edge betweenness centrality measures the importance of each edge in the network based on the number of shortest paths that pass through it. Edges with high centrality act as critical bridges connecting different parts of the network, whereas edges with low centrality often represent redundant or less relevant interactions.

The purpose of this function is to penalize dense gene networks containing numerous superfluous or redundant interactions (with low centrality), favouring those in which only the most essential connections (with high centrality) are retained to maintain network cohesion and functionality. Networks with fewer redundant interactions are not only easier to analyze but also tend to better reflect the true underlying regulatory architecture [127].

This fitness function is the most computationally expensive in `BIO-INSIGHT`. For this reason, two measures have been implemented:

1. **Adaptive approach in evaluation management through caching**. In the early exploratory stages of the evolutionary algorithm, the cache uses lists of confidence values from the consensus network rounded to fewer decimal places as keys. This increased rounding enhances the likelihood of reusing previous evaluations, thereby avoiding the computational cost of evaluating highly similar networks, an aspect that is not critical in this exploratory phase. As the algorithm progresses towards the exploitation and refinement stages, the precision of these values is gradually increased (using more decimal places), allowing for finer distinctions between similar solutions, where small differences may be crucial in identifying higher-quality configurations.

2. **More efficient implementation of the edge betweenness centrality met-**

Algorithm 13 Fourth objective: Reduce Non-Essential Interactions.

Require: Consensus list with confidence values c, Number of decimal places in the rounding (dependent on the evolutionary stage in which the evaluation is taking place) $decimals$
Ensure: Value of the fitness function $result$
1: key = getRoundedHashCode(c, decimals)
2: **if** $cache.containsKey(key)$ **then**
3: $result = cache.get(key)$
4: **else**
5: $graph$ = getWeightedGraph(c, decimals)
6: $edgeBetweenness$ = edgeBetweenness($graph$)
7: sort($edgeBetweenness$)
8: $mean$ = calculateWeightedMean($edgeBetweenness$)
9: $result = 1/mean$
10: $cache.put(key, result)$
11: **end if**
12: **return** $result$

ric calculation. Taking JGrapht's original implementation of this score as a reference, a less computationally expensive alternative [1] has been implemented using Dijkstra's algorithm. The results of this implementation have been verified to be identical to those obtained using the library's original version.

The implementation of this fitness function is detailed in Algorithm 13. The process begins by generating a rounded *hash* key based on the consensus network (line 1 of Algorithm 13) to check whether the result is already stored in the cache. If it is, the result is retrieved directly (line 3 of Algorithm 13).

Otherwise, a weighted graph is constructed from the consensus network (line 5 of Algorithm 13), and edge betweenness centrality is computed using Dijkstra's algorithm (line 6 of Algorithm 13). The centrality values are then sorted in ascending order (line 7 of Algorithm 13), ensuring that the least relevant interactions appear first.

Next, a weighted average of the centralities is computed (line 8 of Algorithm 13), assigning greater weight to edges with low betweenness (those at the beginning of the list). The final value of this minimization-oriented function is

[1] Optimised implementation of Edge Betweenness Centrality metric `https://github.com/AdrianSeguraOrtiz/BIO-INSIGHT/blob/main/EAGRN-JMetal/src/main/java/eagrn/fitnessfunction/impl/topology/EdgeBetweennessCalculatorDijkstra.java`

obtained as the inverse of this weighted average (line 9 of Algorithm 13), favoring networks in which even the least relevant interactions have high centrality values. Finally, the result is stored in the cache and returned (lines 10 and 11 of Algorithm 13).

9.1.5 Objective 5: Degree Distribution

The Degree distribution function is retrieved from chapter 7 and aims to promote the creation of consensus networks that follow a scale-free degree distribution, a typical pattern in biological networks. In these networks, most nodes have only a few connections, while a few hubs concentrate most interactions [44].

To achieve this, the function calculates the degree of each node by summing the decimal confidence levels of the interactions in which the gene acts as either a source or a target. Then, the goodness-of-fit test described in [259] is applied to assess how well the degree distribution of the consensus network aligns with a Pareto distribution, which is characteristic of scale-free networks. The result of this test, which reflects the probability that the network follows this distribution, is the value returned by the function.

9.1.6 Objective 6: Dynamicity

This fitness function evaluates the dynamic stability of the consensus networks. To achieve this, it models the temporal evolution of gene activity using a system of nonlinear ordinary differential equations (ODEs) based on the model presented in [281], applying certain simplifications to formulate it directly from the adjacency matrix of the inferred network.

Given a set of N genes in the network, each node i is represented by a variable $y_i(t)$, which denotes its expression level over time. The temporal evolution of each node is governed by the differential Equation (9.1):

$$\frac{dy_i}{dt} = f\left(\sum_j A_{ji} y_j\right) - y_i \tag{9.1}$$

Where A_{ji} is the weight of the connection between nodes j and i in the inferred adjacency matrix, $f(x)$ is a nonlinear activation function that models gene regulation, and the term $-y_i$ represents the natural degradation of gene expression, preventing the system from growing without constraints.

The activation function used in this model is the Hill function [282] shown in Equation (9.2), due to its widespread and traditional use in modeling transcriptional regulation interactions in GRNs [283, 284]:

$$f(x) = \frac{x^n}{k^n + x^n} \tag{9.2}$$

Where n controls the degree of nonlinearity in the system by modulating the cooperative response of the network, and k adjusts the threshold at which the input signal significantly affects activation. This formulation introduces saturation in the response of each node, reflecting that gene regulation does not respond linearly to stimuli, but instead exhibits threshold behaviours and cooperation among regulators.

In this model, after several experimental tests, the values $n = 2$ and $k = 0.5$ have been set to capture a biologically plausible dynamic from a general perspective. On the one hand, a value of $n = 2$ balances a gradual response and cooperative behaviour, ensuring that activation does not occur linearly or abruptly. On the other hand, the value $k = 0.5$ allows the system's response to be triggered by moderate signals without requiring extreme stimuli or overreacting to small fluctuations.

The dynamic stability of a biological network is a key indicator of its functional robustness. Stable networks tend to maintain their structure and functionality in response to internal or external perturbations. This fitness function is designed to favour individuals whose consensus networks exhibit well-defined temporal evolution and quickly converge to a state, avoiding chaotic systems or those highly sensitive to perturbations.

In this case, it was decided not to use the caching system because the computational cost of this objective function is not as high as that of other functions that require complex calculations on networks using JGrapht. Performance analysis showed that storing all evaluations in a cache would result in higher memory and resource consumption than simply executing the function whenever needed.

The implementation of this fitness function is described in Algorithm 14. First, the consensus network is transformed into a weighted adjacency matrix (line 1 of Algorithm 14). This matrix is then used to construct a system of nonlinear ordinary differential equations (ODEs) that simulates the system's dynamics (line 2 of Algorithm 14).

The Dormand-Prince 5(4) integrator [285] is employed to solve the ODE system with high precision (line 3 of Algorithm 14). Homogeneous initial condi-

Algorithm 14 Sixth objective: Dynamicity

Require: Consensus list with confidence values c
Ensure: Value of the fitness function $result$
1: $adjacencyMatrix$ = getFloatMatrix(c)
2: $model$ = createODEModel($adjacencyMatrix$)
3: $integrator$ = DormandPrince54Integrator()
4: $initState$ = [1.0, 1.0, ..., 1.0]
5: $finalState$ = integrate($integrator$, $model$, $initState$)
6: $result$ = average($finalState$)
7: **return** $result$

tions are set, where all nodes start with a value of 1.0 (line 4 of Algorithm 14), and the system's evolution is simulated (line 5 of Algorithm 14).

Finally, the stability score is computed as the average of the final values of the nodes (line 6 of Algorithm 14). This score reflects the system's ability to maintain stable behaviour under the influence of the nonlinear interactions defined by the adjacency matrix. A value close to the initial state indicates stability, whereas significant deviations suggest instability or complex dynamic behaviours. The final value is returned as the fitness function's result (line 7 of Algorithm 14).

As seen in Algorithm 14, the implementation of this last objective is simple and designed to be as generic as possible, making it applicable to any network, regardless of its size and structure. This initial approach enables the exploration of the stability of consensus networks, with promising results that encourage future refinements in this aspect.

9.2 Experimentation

The first phase of the experimental design in this study aims to demonstrate that incorporating all the biological objectives described in the previous section not only improves the quality of the consensus networks compared to the state of the art, but also maintains computational feasibility, even for large-scale networks, thanks to the proposed architecture.

To achieve this, the first experiment is designed for a fair and rigorous comparison with the most recent and directly related algorithm, `MO-GENECI` (chapter 7), which has been shown to outperform a total of 26 widely used individual inference techniques. Since all the objective functions from `MO-GENECI` have been incorporated into this proposal, replicating its experimental conditions as

closely as possible allows verification that the observed accuracy improvements are solely due to the integration of new biological knowledge introduced through the additional objectives.

For this reason, the same dataset used in its experimentation is employed in this phase, as it is also considered the most extensive and diverse academic benchmark constructed to date for the field of GRN inference. This benchmark consists of a total of 106 networks, with sizes of up to 2,000 genes, sourced from up to ten different origins, including well-recognized challenges in the field (DREAM3 [183] and DREAM4 [142]), simulators (SysGenSIM [179], SynTReN [180], Rogers [181], and GeneNetWeaver [182]), in vivo networks such as IRMA [173], and a broad set of databases compiling verified GRNs through various procedures (TFLink [174], RegulonDB [175], RegNetwork [176], BioGRID [177], and GRNdb [178]). The specifications of this diverse dataset are detailed in section 4.1.

The inference techniques to be integrated into the consensus are the same as those considered in MO-GENECI, specifically the 26 available techniques listed in the previous section. This ensures that the observed accuracy improvements are exclusively attributable to the algorithmic design and objective formulation of BIO-INSIGHT, rather than to the quality of the initial networks provided by the inference techniques. Similarly to what is stated in chapter 7, some of these techniques are restricted to certain ranges of network size due to their high computational cost. This ensures that both proposals start from exactly the same baseline, ruling out the possibility that BIO-INSIGHT's improvements could be attributed to the initial quality of the inferred networks. Since both approaches share the same individual representation, the same crossover and mutation operators have been adopted, maintaining identical parameter settings and fixing the same values for population size (300) and number of evaluations (250,000). This guarantees that differences in accuracy between the two proposals are not due to the over-evolution of either one.

In addition to BIO-INSIGHT and MO-GENECI, other consensus strategies are incorporated into the comparison. These strategies should be capable of integrating any set of techniques and performing consensus in an unsupervised manner. That is, to ensure a fair comparison, the same inference techniques should be combined using a procedure that does not rely on labelled data. Otherwise, if accuracy is evaluated against the gold standard, such strategies would have a significant advantage by having prior, partial, or full access to that reference. Moreover, they are not conceptually comparable to this proposal, as supervised strategies are restricted to academic settings and cannot infer real-world, unexplored gene networks.

This excludes most of the proposals mentioned in the state of the art, either due to their supervised nature [150, 149, 148] or their lack of flexibility in accommodating any initial individual inference method [147, 286]. A promising alternative would be EnsInfer [16]; however, despite its description not mentioning any supervised process, its algorithm requires confirmation of the existence of each interaction[2] as well as partitioning data into training, validation, and test sets.

Finally, due to the scarcity of feasible consensus proposals for algorithmic comparison, several simple strategies have been implemented as reference baselines: mean, median, weighted mean, and Bayesian fusion. The latter is inspired by the EnsInfer implementation, serving as an approximate substitute given the impossibility of its direct use.

Thus, the final comparison will be conducted between `BIO-INSIGHT`, `MO-GENECI`, and the remaining simple consensus strategies. This comparison also indirectly evaluates the set of 26 individual inference techniques, as `MO-GENECI` has already demonstrated superiority over all of them. The evaluation is performed by validating each resulting consensus network against the gold standards using AUPR and AUROC metrics. In the case of evolutionary algorithms that produce an approximated Pareto front, the median-quality solution and the best solution from the front will be considered representative samples.

To statistically validate the superiority of `BIO-INSIGHT`, a Friedman ranking test was applied over the benchmark of 106 gene networks, followed by Holm's non-parametric post-hoc procedure to adjust p-values and determine the significance of differences between methods. This methodology ensures a robust comparative analysis without assuming data normality, and it highlights which proposals achieve statistically better performance across the entire benchmark.

The second phase of the experimental design focuses on assessing the individual contribution of each objective function integrated into `BIO-INSIGHT`, in order to evaluate the innovation brought by the proposal and to test whether the simultaneous optimization of biologically driven yet potentially conflicting objectives leads to superior inference performance.

To this end, an ablation study was conducted by executing `BIO-INSIGHT` on all networks in the benchmark with fewer than 1000 genes (to ensure computational feasibility) under ten different configurations:

- The complete `BIO-INSIGHT` proposal, optimizing all objectives jointly in a

[2]Link to EnsInfer repository: `https://github.com/IcyFermion/network_inference_ensemble/tree/main`

many-objective framework.

- Nine mono-objective variants, each optimizing independently one of the biological aspects considered in the objective space.

It is worth noting that the *Motifs* objective actually integrates four distinct regulatory patterns (*regulatory pathways*, *differentiation*, *bifurcation*, and *coupling*) which are typically combined in a single score. However, for this ablation study, each motif type was optimized individually, in order to isolate their specific contributions. As a result, the number of mono-objective variants increases from six (one per original objective) to nine.

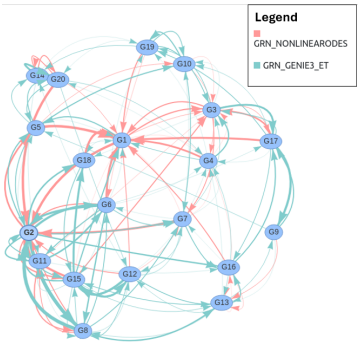
Each configuration was executed on the same subset of networks using identical inference inputs, parameters, and evolutionary operators, ensuring that the only variable factor was the objective function. The resulting consensus networks were evaluated using AUROC and AUPR against the gold standards. These metrics were then subjected to Friedman statistical ranking and Holm's post-hoc non-parametric tests to assess whether the full multi-objective optimization significantly outperforms any of the individual objective configurations.

Finally, the third experimental phase aims to demonstrate the real-world clinical applicability of this proposal once its accuracy in the academic domain has been validated in the previous phase. To achieve this, real-world gene expression data from patients with various pathologies, including fibromyalgia, myalgic encephalomyelitis, and the co-diagnosis of both diseases, have been used. The first step consists of dividing this dataset into four distinct groups: one for each pathology and a control group. Then, `BIO-INSIGHT` is executed on each subset, extracting an approximated Pareto front for each pathology. To reduce each front to a single representative consensus network, all the networks in the front were merged, storing for each interaction its frequency within the front and its average confidence score.

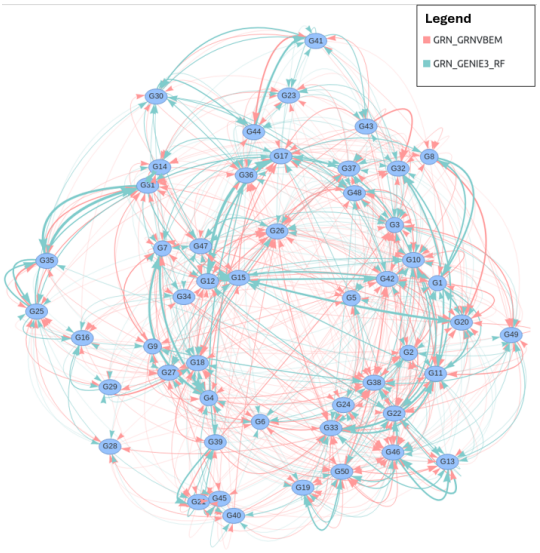
After obtaining the consensus network for each pathology, the designed analysis compares the presence or absence of interactions between different groups. In other words, it aims to observe, for example, whether there are interactions that are inferred in a specific clinical pathology, but not in the control group.

9.3 Results and Discussion

The initial execution of individual inference techniques provides the input for the evolutionary algorithm in this proposal. Therefore, for each problem, the gene regulatory network inferred by each technique is obtained.



(a) Simulated 20-node network with scale-free topology and overexpression perturbation. Confidence levels above 0.3.



(b) First yeast network from the DREAM3 challenge with a size of 50 nodes. Confidence levels above 0.2.

Figure 9.2: Comparison of the networks inferred by the two most accurate techniques for a given problem. Each technique is represented by a different color, arrows indicate the direction of regulation, and their thickness reflects the confidence level of the interaction.

9.3.1 Performance Analysis

Although the consensus of inference networks through `BIO-INSIGHT` does not require the prior computation of each technique's accuracy, this information is highly useful for making observations and future comparisons.

A noteworthy initial observation is the surprising disparity among the inferred networks, even when they exhibit similar accuracy. Across the results obtained from the 26 individual techniques, markedly different interactions can be identified, highlighting an apparent inconsistency among them. To illustrate this discrepancy, Figure 9.2 presents the overlap of the two most accurate techniques in two specific problems.

First, Figure 9.2a compares the network inferred by NONLINEARODES with the one obtained by GENIE3_ET in a simulated 20-node network. The difference between them is evident: while GENIE3_ET favours a more dispersed topology, NONLINEARODES models gene regulation through a central hub, which is en-

tirely ignored by the first technique.

Second, Figure 9.2b presents the comparison between GRNVBEM and GENIE3_RF in a 50-node network from the DREAM3 challenge. Although both networks appear to share a similar topological structure, the proposed interactions occur between different genes.

Answer to RQ1

Yes, the results show a significant disparity among the networks inferred by different high-accuracy techniques. The differences in topology and inferred interactions suggest a lack of coherence that could impact biological interpretation. This justifies the need to develop consensus methods integrating information from multiple approaches to obtain more robust and reliable representations.

After executing `BIO-INSIGHT` to consolidate the networks inferred by individual techniques, an approximated Pareto front is obtained for each problem. In scenarios with a high number of objectives, such as the one in this study, ensuring an adequate trade-off between them is crucial. The presence of overlapping objectives can hinder the effective exploration of the solution space, bias the dominance of specific objectives, and compromise the diversity of the Pareto front.

By visualizing various fronts obtained by `BIO-INSIGHT` through interactive parallel coordinate plots, a correct partial or even total conflict between the objectives of this proposal has been observed. However, statically representing this phenomenon in a single figure is challenging. Therefore, a specific sample has been selected, and a visualization has been designed to illustrate the opposition among all objectives in a single image.

Figure 9.3 presents a chord diagram for the approximated Pareto front obtained for one of the networks from the DREAM4 challenge. In this figure, the subset of individuals with the worst fitness values is selected for each objective. This allows for observing how sacrificing a specific objective enables individuals to achieve better fitness scores in the remaining objectives. For example, considering the worst-performing individuals in the Dynamicity objective, highlighted in dark purple, it is evident that they occupy highly optimized positions in other objectives, such as Reduce Non-Essential Interactions or Eigen Vector (metric) Distribution.

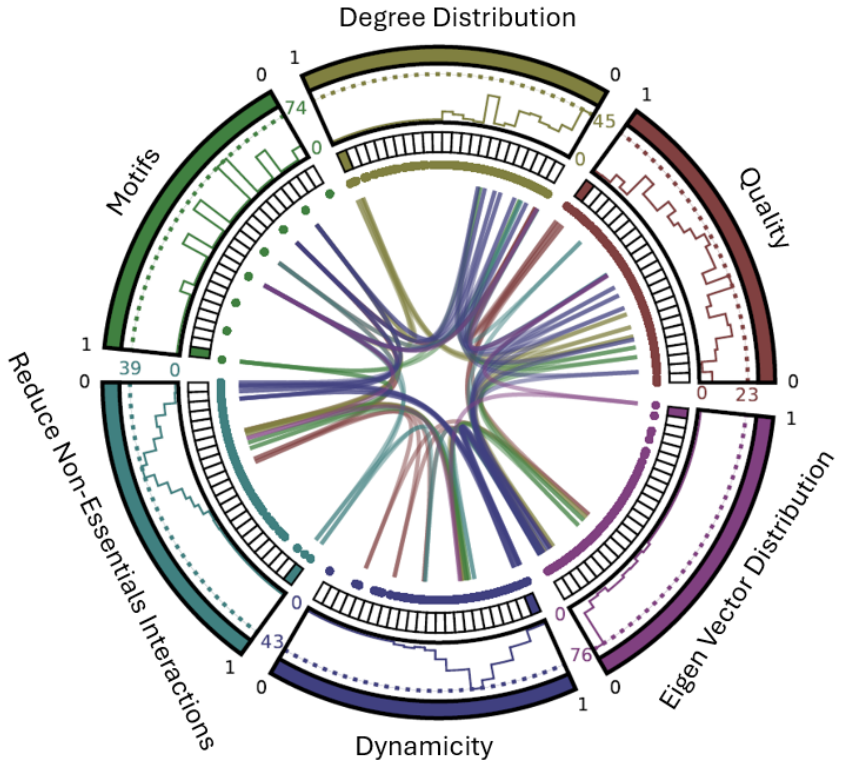
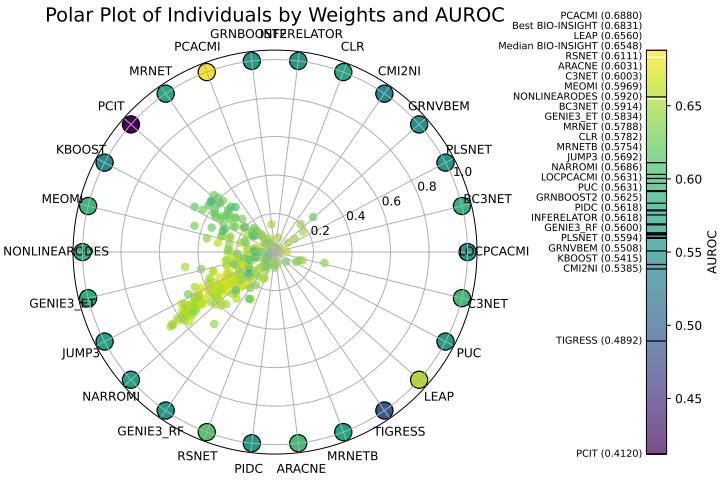


Figure 9.3: Chord diagram for the solution front obtained by BIO-INSIGHT after optimizing the consensus of the techniques applied to the first network from the DREAM4 challenge, with a size of 10 nodes. In this diagram, each objective is represented as a circular trapezoid. Inside the trapezoid, a histogram illustrates the solutions' distribution across the corresponding objective's normalized values. Dashed lines indicate the maximum and minimum values within the histogram. Small boxes at the base of the circular trapezoid represent subsets of individuals grouped based on their proximity in the objective score. These boxes are initially interactive and allow the selection of individuals to be displayed in the core of the diagram. Since this document is static, a snapshot has been taken, selecting the worst-performing individuals for each fitness function to highlight the opposition between the objectives of the algorithm appropriately.

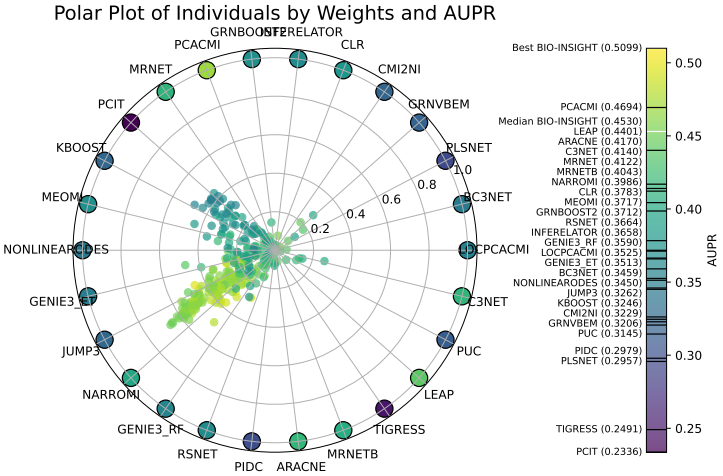
Answer to RQ2

Yes, the consideration of multiple biological aspects does not compromise the trade-off between objectives at the evolutionary core of the proposal. The exploration of the obtained fronts confirms the presence of a partial or total conflict among the objectives, ensuring a balanced and diverse optimization.

Although it has already been demonstrated that, due to specialization do-



(a) Polar chart for AUROC



(b) Polar chart for AUPR

Figure 9.4: Polar plot locating each solution from the front obtained by `BIO-INSIGHT` for the second yeast network from the DREAM3 challenge, with a size of 10 nodes. In these plots, individuals are represented by points positioned according to the weights assigned to different techniques and are coloured based on their accuracy level. Additionally, larger markers represent simulated solutions (not part of the front), corresponding to assigning all the weight to a single technique. To the right of the radar, a sidebar displays the colour gradient associated with the accuracy metric in question. In this bar, black markers indicate the accuracy values of each technique, while white markers highlight the accuracy of `BIO-INSIGHT`'s best solution as well as the median of the front.

mains, no inference technique is generally more accurate than others [16], re-

searchers might still be tempted to perform consensus using only techniques that have yielded good accuracy in other datasets. Therefore, it is interesting to analyze the weights assigned by the highest-accuracy individuals in the front to the different techniques and examine whether there is any correlation between these weights and the individual accuracy of the techniques.

It is natural to assume that BIO-INSIGHT would assign higher weights to the techniques with greater individual accuracy. However, this is not necessarily the case.

Figure 9.4 presents a polar plot with the individuals from the front of one of the networks from the DREAM3 challenge, positioned based on the weights assigned to the different techniques and coloured according to their accuracy level. As observed, the highest-quality individuals (more yellow) are not necessarily concentrated near the techniques with the highest individual accuracy for the two accuracy metrics. This suggests that the algorithm adopts a holistic perspective, where consensus quality is not merely the sum of the individual accuracies of each technique. Thus, a balanced selection of less accurate techniques can generate high-quality networks, whereas consensus among highly accurate techniques does not necessarily guarantee higher accuracy.

Answer to RQ3

No, the results indicate that the optimization of consensus in BIO-INSIGHT does not align with an individualistic approach in which the accuracy of the techniques correlates with their level of participation in the consensus. Instead, the algorithm appears to adopt a more holistic perspective, where the combination of lower-accuracy techniques can lead to higher-quality consensus networks due to their complementarity.

After analyzing the fitness values of the individuals and the weights they assign to the different techniques, it is essential to examine the characteristics of the consensus networks constructed from the individuals in the Pareto approximation front. Although the algorithmic comparison discussed later quantifies the accuracy of these networks against the gold standards, it is also important to verify that, in addition to being accurate, the consensus networks exhibit biologically coherent properties that ensure clear biological interpretability.

Figure 9.5 presents the most accurate consensus gene regulatory network obtained by BIO-INSIGHT for one of the networks from the BioGRID repository. This network displays several characteristics that align with expectations in this domain: a scale-free topology with high connectivity in the core, a modular structure that enables the distinction of multiple functional communities, and a

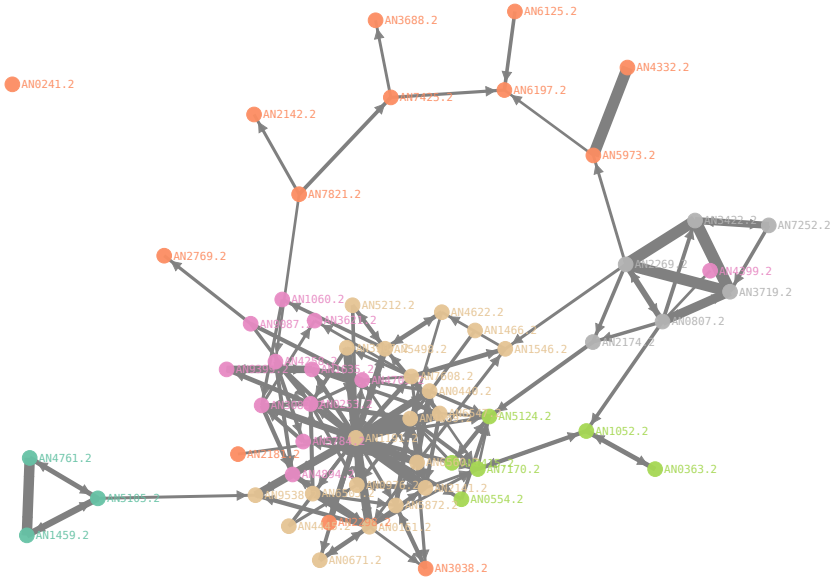


Figure 9.5: Representation of the best solution obtained by BIO-INSIGHT for the *Emericella nidulans FGSC A4* network extracted from the BioGRID repository. Genes are coloured based on their neighbourhood, arrows indicate the direction of regulation, and their thickness represents the confidence level of the interaction.

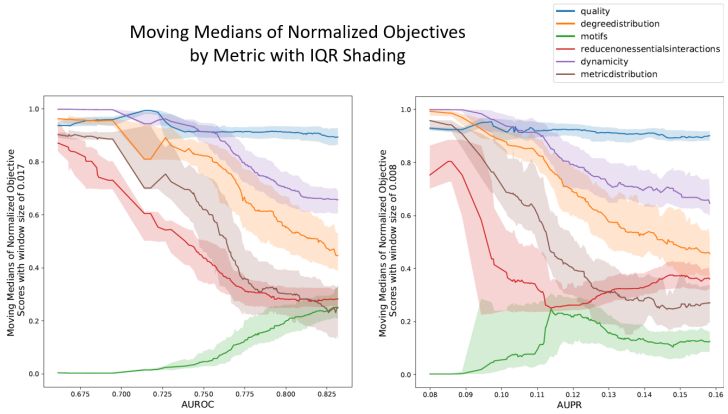
strong presence of regulatory motifs, such as a regulatory pathway (in orange), a clear feedforward loop (in light blue), and a biparallel motif (in grey).

The presence of these characteristics in high-accuracy consensus networks inferred by BIO-INSIGHT validates the hypothesis of this study: an intelligent consensus of individual techniques, guided by a biologically comprehensive objective space, not only brings the results closer to biologically plausible scenarios, but also leads to a significant accuracy enhancement that cannot be achieved through the purely mathematical satisfaction of current literature proposals.

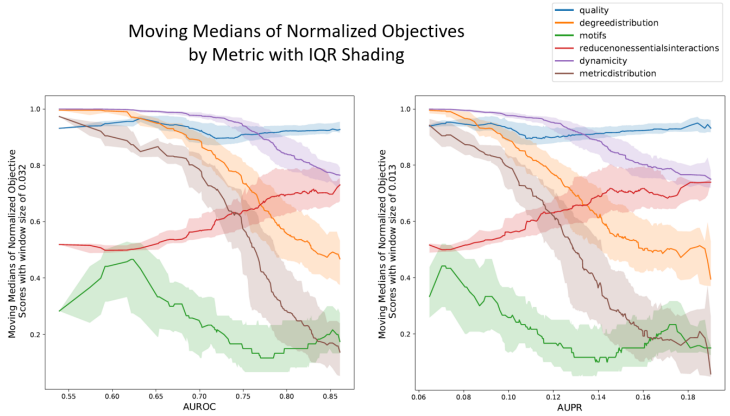
Answer to RQ4

Yes, the networks inferred by BIO-INSIGHT exhibit a biologically coherent structure consistent with current knowledge of GRNs. Their characteristics, such as topology, structure, and patterns, indicate that consensus guided by biologically comprehensive objectives not only enhances accuracy, but also generates networks with greater interpretability, aligning with the underlying biology.

The simultaneous occurrence of high accuracy and biologically coherent properties in the consensus networks generated by BIO-INSIGHT, along with their role



(a) Network *Glycine max* extracted from the BioGRID repository.



(b) Network *Strongylocentrotus purpuratus* extracted from the BioGRID repository.

Figure 9.6: In these plots, the moving medians of the normalized objective values for the individuals forming the front obtained by `BIO-INSIGHT` are represented and sorted for each accuracy metric. Additionally, each moving median is shaded by the interquartile range, allowing an outline of the diversity of objective values at each accuracy level.

in validating the hypothesis of this study, can only be explained in one way: the biological objectives designed in this chapter are somehow aligned with inference accuracy.

To demonstrate this, Figure 9.6 presents the moving medians of the objectives sorted for each accuracy metric in the fronts of two specific BioGRID networks. In both cases, for the *Glycine max* network (Figure 9.6a) and network *Strongylocentrotus purpuratus* (Figure 9.6b), a clear relationship exists between the level

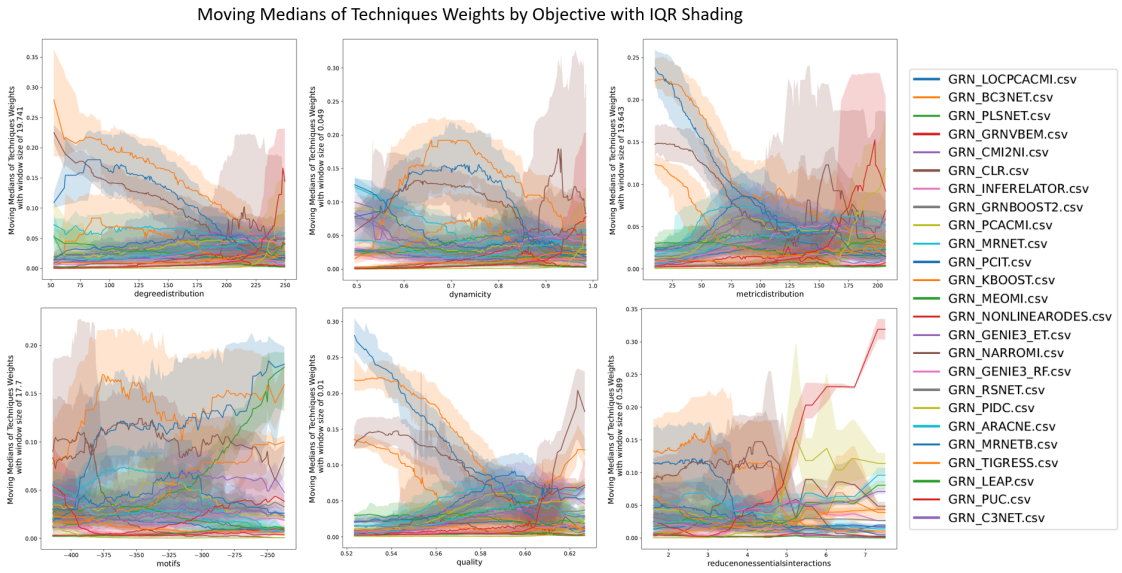


Figure 9.7: In these plots, the moving medians of the weights assigned to each technique in the front obtained by BIO-INSIGHT for the *Glycine max* network from BioGRID are represented and sorted for each objective of the algorithm. Additionally, as in Figure 9.6, each moving median is shaded by the interquartile range, which in this case outlines the diversity of the weights assigned to each technique at each objective function value.

of optimization of the different objectives and the AUPR and AUROC values of the individuals.

Answer to RQ5

The conflicting objectives of BIO-INSIGHT have demonstrated that, despite not using labelled data at any point, their optimization is clearly related to inference accuracy. This helps validate the design of the objective functions, their appropriate directionality within this field, and their feasible use in real-world settings where the networks to be inferred are yet to be discovered.

It is essential to show that the accuracy enhancement, and thus the optimization performed by BIO-INSIGHT, does not duplicate or interfere with the initial learning occurring during the execution of individual inference techniques. Although these techniques are primarily based on mathematical principles that differ significantly from BIO-INSIGHT's objectives, some indirect alignment could still exist.

To rule out this possibility, Figure 9.7 presents the moving medians of the

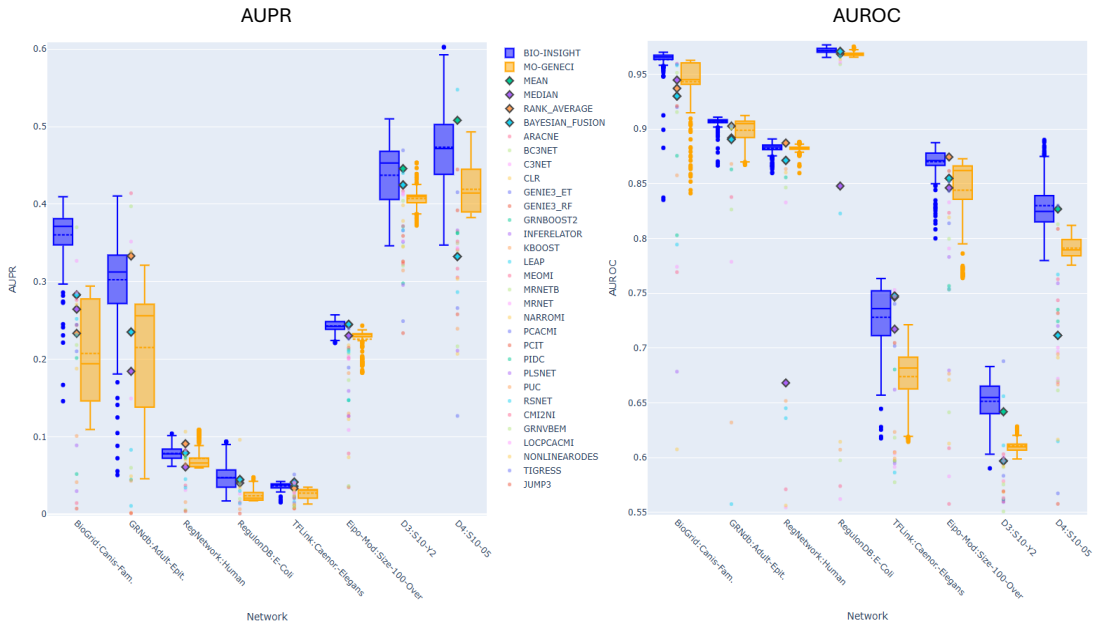


Figure 9.8: Comparison of the AUPR and AUROC metrics obtained by BIO-INSIGHT (blue box) for a diverse set of networks from different sources, in relation to MO-GENECI (orange box), other consensus strategies (diamonds), and individual techniques (circles).

weights assigned to each technique, sorted for each objective of the algorithm, in the front of one of the BioGRID networks. This representation reveals the wide interquartile ranges and the lack of directionality in the curves. This noise confirms that the objectives do not favour any particular technique individually. In other words, no techniques are specialized in a specific objective, nor do they individually contribute the biological knowledge that the objective represents.

Answer to RQ6

The carefully designed objective functions in this study have successfully demonstrated their complete independence from the methodologies integrated into the individual techniques. This means that applying BIO-INSIGHT after executing these techniques to optimize their consensus provides entirely novel information.

Answering each research question has justified the motivation and validated the design of the algorithmic proposal. However, to demonstrate the scientific contribution of this research, it is essential to conduct a rigorous accuracy comparison with existing state-of-the-art methodologies.

Table 9.1: Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.

AUPR		
Technique	$Friedman'sRank$	$Holm'sAdj - p$
BEST_BIO-INSIGHT	**1.51887**	-
BEST_MO-GENECI	2.8396	8.6584e-05
MEAN_WEIGHTS	4.8349	1.2970e-22
RANK_AVERAGE	4.8962	3.1183e-23
MEDIAN_BIO-INSIGHT	5.0896	1.0415e-25
MEDIAN_WEIGHTS	5.4151	2.6040e-30
BAYESIAN_FUSION	5.5802	9.0595e-33
MEDIAN_MO-GENECI	5.8255	1.1532e-36

Table 9.2: Friedman mean rank with Holm's adjusted p values (0.05) for AUROC.

AUROC		
Technique	$Friedman'sRank$	$Holm'sAdj - p$
BEST_BIO-INSIGHT	**1.69811**	-
BEST_MO-GENECI	2.9387	2.2685e-04
MEDIAN_BIO-INSIGHT	4.2359	9.2290e-14
RANK_AVERAGE	4.8443	2.6078e-20
MEAN_WEIGHTS	5.1179	1.1488e-23
MEDIAN_MO-GENECI	5.3774	3.9178e-27
BAYESIAN_FUSION	5.7689	6.4410e-33
MEDIAN_WEIGHTS	6.0189	6.7039e-37

In Figure 9.8, the AUPR and AUROC values are represented for a diverse set of networks inferred by `BIO-INSIGHT`, `MO-GENECI`, several consensus strategies, and the 26 individual inference techniques. The results show that `BIO-INSIGHT` clearly dominates in both accuracy metrics, consistently outperforming all other evaluated methodologies.

A Friedman statistical ranking with Holm's non-parametric tests has been conducted on the academic benchmark of 106 gene networks to corroborate this superiority in a statistically rigorous manner. Since `MO-GENECI` had already demonstrated in its results (see section 7.3) that it outperforms the 26 individual techniques using this same statistical test, these techniques have been excluded from this analysis, focusing instead on comparing consensus approaches.

In Table 9.1 and Table 9.2, the statistical test results for the AUPR and AUROC metrics, respectively, are presented. BEST_BIO-INSIGHT achieves the highest

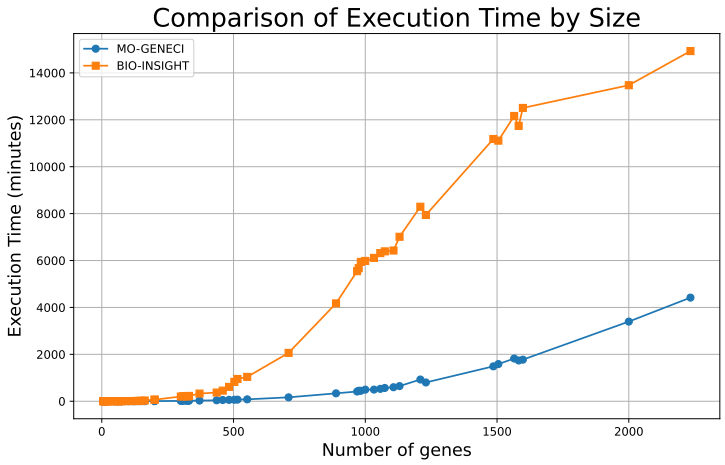


Figure 9.9: Comparison of execution time depending on the network size for MO-GENECI (blue) and BIO-INSIGHT (orange).

average ranking in both metrics, significantly outperforming all other methodologies. The statistical comparison further confirms that this difference is significant, with p-values for the other techniques being extremely low and far from the commonly recognized 0.05 threshold.

Notably, BEST_MO-GENECI, the second-best method in both cases, is at a considerable distance from BIO-INSIGHT. Given that previous studies had already demonstrated that MO-GENECI outperforms the 26 individual inference techniques, these results indicate that BIO-INSIGHT surpasses it even more significantly. Therefore, any additional comparison with the individual inference techniques would be redundant, as their inferiority has already been indirectly established through the significant and consistent superiority of BIO-INSIGHT over MO-GENECI. Additionally, traditional consensus strategies, such as mean, median, and rank average, fall considerably behind, sometimes even being outperformed by a random selection from the BIO-INSIGHT front rather than a proper selection made by a domain expert.

To improve the transparency of the study, the AUROC and AUPR values obtained for each inferred network, using both the individual techniques and the consensus strategies including BIO-INSIGHT, have been provided as supplementary material.

The accuracy improvements introduced by this proposal involve a considerably high algorithmic complexity. Although the necessity of its implementation has been thoroughly justified through the resolution of various research ques-

tions, the computational cost of its execution remains a relevant aspect.

For this reason, from the beginning of BIO-INSIGHT's design, numerous measures were taken to minimize the impact of its complexity on execution time: the implementation of an asynchronous parallel model, the caching system, the elimination of unnecessary intermediate steps, the adaptive approach to managing evaluation storage, the efficient implementation of graph-based scoring methods from the literature, and more.

Although these optimizations cannot fully compensate for adding three extra dimensions to the search space, they have allowed BIO-INSIGHT's operational range to remain comparable to the closest previous strategy.

Figure 9.9 compares the MO-GENECI algorithm and BIO-INSIGHT, clearly showing the exponential increase in BIO-INSIGHT's execution time as network size increases. Both algorithms were analyzed using the same number of evaluations (250,000) and executed under the same computational resources (500 GB of RAM and 32 cores).

Regarding the algorithm's asymptotic time complexity (Big-O), an implementation-informed approximation is provided below:

- *Preprocessing:* run the k base methods, with total $C_{\text{base}} = \sum_{i=1}^{k} C_i$.

- *Evolutionary optimization (over E evaluations):* in each evaluation, (A) consensus construction costs $O(m \cdot k)$; (B) objectives include: *Eigenvector distribution* (power iteration) $O(t\,(n+m))$ plus sorting $O(n \log n)$; *Reduce non-essential interactions* (weighted edge-betweenness) $O(nm + n^2 \log n)$ plus ranking $O(m \log m)$; *Degree distribution* $O(m + n \log n)$; *Motifs* up to $O(n^3)$ (amortized by caching); and *Dynamicity* via nonlinear ODE integration $O(S\,n^2)$ with S adaptive steps. Summing per-evaluation:

$$O(m \cdot k) + O\Big(t(n+m) + n \log n\Big) + O\Big(nm + n^2 \log n + m \log m\Big) + O(m + n \log n) + O(n^3) + O(S\,n^2).$$

  A compact bound is

$$O\Big(m \cdot k + t(n+m) + n \log n + nm + n^2 \log n + m \log m + m + n \log n + n^3 + S\,n^2\Big).$$

- *Overall:*

$$\boxed{C_{\text{base}} \;+\; C_{\text{BIO-INSIGHT}}} \;\approx\; C_{\text{base}} + O\Big(E\,\Big(nm + n^2 \log n + S\,n^2 + n^3\Big)\Big),$$

Table 9.3: Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.

AUPR		
Configuration	$Friedman's Rank$	$Holm's Adj-p$
BEST_BIO-INSIGHT	**1.3779**	-
O6: Dynamicity	5.1337	4.1353e-16
O2-1: MotifsDifferentiation	5.5581	2.7626e-19
O3: EigenVectorDistribution	5.6454	7.2146e-20
O2-2: MotifsRegulatoryRoute	5.7616	8.8521e-21
O4: ReduceNEInteractions	6.0233	4.1025e-23
O1: Quality	6.0407	3.3516e-23
O2-3: MotifBifurcation	6.1570	2.9065e-24
O2-4: MotifCoupling	6.6512	2.9530e-29
O5: DegreeDistribution	6.6512	2.9530e-29

Table 9.4: Friedman mean rank with Holm's adjusted p values (0.05) for AUROC.

AUROC		
Configuration	$Friedman's Rank$	$Holm's Adj-p$
BEST_BIO-INSIGHT	**1.6570**	-
O6: Dynamicity	5.1744	2.5713e-14
O4: ReduceNEInteractions	5.3488	2.5701e-15
O3: EigenVectorDistribution	5.4942	2.8512e-16
O5: DegreeDistribution	5.8605	3.4785e-19
O1: Quality	5.9361	9.4989e-20
O2-3: MotifBifurcation	6.0581	9.2336e-21
O2-2: MotifsRegulatoryRoute	6.0930	5.1874e-21
O2-1: MotifsDifferentiation	6.1861	8.2152e-22
O2-4: MotifCoupling	7.1919	3.7083e-32

where edge betweenness, dynamicity, and (if heavily used) motifs tend to dominate for large n.

- *Symbols:* n = genes; m = candidate interactions; k = base methods; t = power-iteration steps; S = ODE solver steps; $E \approx T \cdot P$; C_{base} = total cost of running all base methods once.

9.3.2 Objective Function Ablation Study

Tables 9.3 and 9.4 present the results of the ablation study based on the Friedman ranking and Holm's post-hoc procedure, considering AUROC and AUPR as

evaluation metrics. In both cases, the configuration that jointly optimizes all objectives (**BEST_BIO-INSIGHT**) obtains the best average rank with statistically significant differences (adjusted $p < 0.05$) with respect to all mono-objective variants.

Only the best-performing individual from the Pareto front is considered in this comparison. The median-quality solution was deliberately excluded, as it would correspond to an uninformed random selection within the front, and would therefore be at a disadvantage compared to mono-objective variants that always return the single optimal solution found.

These results confirm that the simultaneous optimization of multiple biologically grounded objectives leads to more accurate consensus networks than optimizing each one in isolation. This supports the main hypothesis of this study, which advocates for a holistic and multi-faceted approach to GRN inference.

In addition, the rankings provide valuable insights into the relevance of each objective. Notably, several of the best-ranked mono-objective variants correspond to newly proposed objectives not inherited from previous literature, such as *Dynamicity* (O6), *Reduce Non-Essential Interactions* (O4), and *Eigenvector Distribution* (O3). These consistently outperform classical objectives like *Quality* (O1) or *Degree Distribution* (O5), reinforcing the contribution and novelty of this proposal in terms of biological interpretability and inference accuracy.

9.3.3 Real-world clinical application

The performance of `BIO-INSIGHT` was also evaluated using non-simulated gene expression data from 43 female subjects: 8 diagnosed with Myalgic Encephalomyelitis/Chronic Fatigue Syndrome (ME/CFS), 10 with Fibromyalgia (FM), 16 with both ME/CFS and FM (co-diagnosed from now on), and 9 healthy controls (GSE269048 dataset) [287]. ME/CFS and FM are chronic conditions characterized by fatigue, pain, and other disabling symptoms for which no validated biomarkers exist. Gene expression levels were measured using custom Affymetrix HERV-V3 microarrays [205], targeting 1,559 genes related to immunity, inflammation, cancer, central nervous system functions, differentiation, telomere maintenance, chromatin structure, and gag-like genes. Data preprocessing and normalization steps prior to analysis are detailed in [287].

Using these data, `BIO-INSIGHT` predicted pairs of interacting genes for each condition. After applying the approximate Pareto front reduction discussed in Section 9.2, it could be observed that all inferred interactions for the same pathology appear in 100% of the solutions of the corresponding front (frequency

Figure 9.10: Differential gene enrichment and unique interactions across conditions. (A) Pathway enrichment analysis comparing gene expression between control and ME/CFS, FM, and co-diagnosis study groups. Dot size represents the number of genes involved while colour indicates the -log10(FDR) significance. (B) Venn diagrams displaying the overlap of predicted gene interactions across study groups. (C) Unique pathway interactions for the control and ME/CFS groups, highlighting significantly enriched biological processes.

= 1), which is evidence of the stability and consistency of the predictions. Results were compared between disease groups and controls by calculating differences in the average weight of interactions. A threshold of $|0.4|$ was applied to identify significant differences (Table S1 in Suplementary Material). In the case of duplicate interactions, where genes could act as both regulators and regulated, only the pair with the highest average weight difference was retained (Table S2 in Suplementary Material). When each disease study group was compared to the healthy control group, 32 additional and 35 absent gene interactions were identified in ME/CFS, while fewer interactions appeared related to FM and the

co-diagnosed groups, most seemingly absent with respect to controls. Specifically, 40 gene regulations were found absent in FM, with only 2 additional for this condition, and 41 were absent in the co-diagnosed group, which displayed 6 disease-associated additional interactions. Some of these genes, i.e. CD74 and TAB2 in the ME/CFS group, were also found to be differentially expressed [287], in support of potential biological relevance of the predicted interactions (Table S2 in Suplementary Material). Overrepresentation analysis (ORA) of the interacting gene sets for each condition, using Gene Ontology terms for Biological Processes, revealed that cytoplasmic translation, immune response and programmed cell death pathways could be affected across all disease groups (Figure 9.10A). These findings align with the existing literature, supporting a role for immune abnormalities in ME/CFS and FM [288]. Furthermore, alterations in apoptotic signalling [289, 290] and protein synthesis [291, 292, 293] can lead to immune dysregulation, potentially contributing to the clinical manifestations observed in these patients.

Given the overlapping clinical features of ME/CFS and FM, and the lack of specific biomarkers, we expanded the analysis to compare potential gene interactions across ME/CFS, FM, and co-diagnosed regulatory networks. This approach identified significant gene-gene interactions, particularly in the ME/CFS study group, most absent in the FM (36/38) or co-diagnosed (38/41) groups (Table S2 in Suplementary Material). To further elucidate disease-specific or commonly absent interactions across all groups, the intersection between group comparisons is obtained. Venn diagrams are used to illustrate the specific or common regulatory interactions across these conditions (Figure 9.10B). A set of 25 gene-gene interactions are predicted to be missing across all disease groups (Table S3 in Suplementary Material, control unique intersections), including gene pairs such as IL6-CCL8, TNF-TAB2, or RPS10-RPL19, involved in critical pathways like cytokine signalling, immune activation, and protein synthesis, respectively (Figure 9.10B-C). Conversely, 27 gene-gene interactions related to programmed cell death are uniquely present in ME/CFS (Table S3 in Suplementary Material and Figure 9.10B-C), suggesting distinct regulatory alterations specific for this condition. Among them the CD74-EIF4G2 interaction, involving a cell surface protein that participates in several immune processes, including inflammatory or autoimmune diseases [294] and a protein involved in the regulation of protein synthesis, leading to immune dysregulation when its function is impaired [291, 292, 293] should be highlighted for the available validating information. In addition to CD74, a human endogenous retrovirus (HERV) (the MLT1_5q32 element) encoded in one of its introns, were found differentially expressed in ME/CFS, supporting its potential implication in this disease [287]. Furthermore, CD74 gene's potential role linking monocyte functioning and neurological symptoms in

ME/CFS, with biomarker value, had been previously described by [295]. On another hand, the physical interaction between CD74 and EIF4G2 is experimentally confirmed by [296] using Affinity Capture-MS, further validating BIO-INSIGHT's predictive capacity and underscoring its utility in identifying clinically relevant gene predictions. These results demonstrate that BIO-INSIGHT provides a valuable tool for understanding the molecular underpinnings of ME/CFS and FM, offering insights into gene regulatory networks that could inform future therapeutic strategies.

Chapter 10

MOEBA-BIO: Flexible framework for self-constructing evolutionary biclustering in biological domains

This chapter presents MOEBA-BIO (Multi-Objective Evolutionary Biclustering Algorithm for BIOmedical applications), a new biclustering framework designed to make better use of biological domain knowledge in order to maximize learning during the algorithm's execution and extend its capabilities to areas not addressed by other approaches, such as parameter self-configuration and the self-determination of the number of biclusters.

The designed framework proposes a new broader-perspective encoding in which each individual represents a complete set of biclusters equivalent to a final solution to the problem (see Figure 3.1 for a better understanding). In this codification, the number of biclusters is not predefined and becomes part of the algorithm's learning process. Unlike previous approaches, this representation deletes the need for subjective post-processing or stochastic combinations of partial solutions.

From a general application perspective, the proposed representation in MOEBA-BIO opens the door to the integration of global objective functions that evaluate aspects such as the distribution and differentiation of the biclusters as a whole, rather than just individual qualities. This is particularly useful in biomedical problems, where interpreting the results requires a more holistic data structure perspective. Moreover, when biclustering is focused on a specific problem, it allows for the inclusion of objectives that leverage this global vision of the problem and address domain-specific qualities, enabling a more in-depth analysis tailored

to the nature of the data.

The contributions to this chapter are:

1. **New biclustering framework for biomedical data**: `MOEBA-BIO` stands out in the current state of the art as the first framework for designing evolutionary biclustering algorithms specialized in the biomedical field. It incorporates up to seven well-known multi-objective meta-heuristics (including their subparameters), multiple biclustering objectives, various crossover and mutation operators, the two studied encodings (the traditional and the proposed one), and a wide range of observers that enable accurate monitoring of the evolutionary process throughout execution.
2. **New codification with a holistic perspective**: The integrated encoding within the framework provides a more realistic perspective on the problem by directly incorporating domain-specific biological knowledge.
3. **Self-learning of the number of biclusters**: Context-guided self-determination of solution size, free from post-processing and redundancies.
4. **Context-driven automatic design of algorithmic proposals**: Implementation of a sophisticated self-configurator that not only adjusts traditional parameters based on technical metrics like hypervolume [297], but also employs supervised metrics directly related to the application domain. This enables the selection of metrics tailored to the data context, ensuring that the objectives and their self-configured subparameters are correctly aligned with the particularities of the dataset. In this way, `MOEBA-BIO` is pre-configured using representative academic data, ensuring that the resulting configuration is adapted for execution on real-world data. Thus, self-configured parameters accurately and confidently reflect biomedical domain knowledge, providing precise and reliable tuning for real-world problems.
5. **New objectives of global perspective and general purpose**: To showcase the potential of the complete encoding even in a general context, this study proposes two new objectives, previously unattainable with traditional encoding. These objectives utilize knowledge from other biclusters within the individual to enhance the global coherence of the solution: *Adaptive bicluster size* (Adaptive bSIZE) and *Bicluster differentiation* (bDIFF).

10.1 Methods

MOEBA-BIO implements the evolutionary metaheuristics scheme by making available a wide range of variants and configurations, allowing the context-driven

Algorithm 15 Self-configuring scheme of evolutionary metaheuristics offered by `MOEBA-BIO`.

Require: Problem p, Objectives o, Max evaluations $maxEvals$
Ensure: Pareto front approximation $front$
1: $population \leftarrow generate(p)$
2: $evaluations \leftarrow 0$
3: **while** $evaluations < maxEvals$ **do**
4: $\quad evaluated \leftarrow evaluate(population, o)$
5: $\quad selected \leftarrow select(evaluated)$
6: $\quad offspring \leftarrow crossover(selected)$
7: $\quad mutated \leftarrow mutate(offspring)$
8: $\quad population \leftarrow update(population, mutated)$
9: $\quad evaluations \leftarrow evaluations + |mutated|$
10: **end while**
11: $front \leftarrow nonDominated(population)$
12: **return** $front$

self-design of complex algorithmic proposals. This scheme, compatible with all the algorithms considered and selectable within the framework, is detailed in Algorithm 15. Each phase of this scheme has multiple options in `MOEBA-BIO`, made selectable through a clear hierarchy of parameters and subparameters, detailed in subsequent sections.

The first version of the framework presented in this chapter only includes objectives for solving biclustering problems on numerical data. However, its implementation has been left open to facilitate the future incorporation of objectives associated with heterogeneous data. In fact, from the initial version, users are required to specify the data type stored in each column of the input matrix. This information is currently provided to all the objective functions implemented in this framework [1].

Since the objective functions included in the framework are freely chosen and configured, `MOEBA-BIO` does not specialize in detecting any specific pattern in the data. However, in the first phase of experimentation, the goal is to detect constant biclusters using the most traditional functions to validate the new representation.

Additionally, it is worth mentioning that numerical data are pre-normalized to ensure the normalization of the objective functions, allowing the integration of

[1] Java class available in: `https://github.com/AdrianSeguraOrtiz/MOEBA-BIO/blob/main/src/main/java/moeba/fitnessfunction/FitnessFunction.java`

any multi-objective meta-heuristics (including MOEA/D [81]). Moreover, overlapping is avoided in one of the two dimensions, which is a common constraint in biclustering proposals [124, 101, 298], enabling a more focused encoding design.

MOEBA-BIO is a framework built on top of jMetal [276], extended with custom parameters, encodings, operators, and objective functions specifically designed for biclustering problems in biomedical applications. Additionally, several observers have been included to facilitate the experimental tracking of important aspects of this problem, such as the number of biclusters [2]. Moreover, the framework's functionality has been extended with the implementation of a specific self-configurator for this new environment [3].

All of these elements are selectable in the tool's configuration, including the optimization algorithm to use and traditional parameters, such as crossover probability, mutation probability, population size, etc. Given this wide range of possibilities, Table 10.1 lists all the top-level configurable parameters in `MOEBA-BIO` (without delving into subconfigurations due to space limitations). Following a general overview of these parameters, subsequent subsections focus on the most critical ones or those where this framework has significantly contributed to better highlighting its applicability to biclustering on biomedical data.

As shown in Table 10.1, the only two mandatory arguments are the input data matrix in CSV format and a complementary JSON file specifying the data type stored in each column (for future non-numerical implementations). The traditional representation, called "PARTIAL", was initially implemented for experimental purposes and remains available alongside the new "COMPLETE" representation introduced in this thesis, which is explained in the following subsections. The complete representation includes two optional parameters that define the range for the number of biclusters in the algorithm's initial population. By default, this range is set to a reasonable range between 5% and 25% of the total number of rows. While this range is not fixed during execution, as the number of biclusters in individuals can vary throughout the evolutionary process, it does serve as an initial seed for the genetic content.

The number and combination of objective functions are entirely flexible. On the one hand, functions specific to the complete representation, where there is a global perspective, cannot be used if the partial encoding is selected. On the other hand, and related to the next parameter in Table 10.1, traditional

[2]Java class available in: `https://github.com/AdrianSeguraOrtiz/MOEBA-BIO/blob/main/src/main/java/moeba/utils/observer/impl/BiclusterCountObserver.java`

[3]Java class available in: `https://github.com/AdrianSeguraOrtiz/MOEBA-BIO/blob/main/src/main/java/moeba/parameterization/ParameterizationRunner.java`

Table 10.1: Configurable parameters of MOEBA-BIO framework.

Parameter	Description
-input -dataset	Path to the input CSV dataset for biclustering. **Default value:** N/A
-input -column -types	Path to the input JSON file specifying the column names and their data types. **Default value:** N/A
-representation	Type of representation to use. **Valid values:** COMPLETE, PARTIAL. **Default value:** COMPLETE
-complete -initial -min -num -bics	Initial minimum number of biclusters (only for COMPLETE representation). **Valid values:** Integer. **Default value:** 5% of the number of rows
-complete -initial -max -num -bics	Initial maximum number of biclusters (only for COMPLETE representation). **Valid values:** Integer. **Default value:** 25% of the number of rows
-str -fitness -functions	Fitness objectives to optimize, separated by semicolons. Sub-parameters can be specified in brackets after the identifier. Objectives marked with an * are only available for the COMPLETE representation. **Valid values:** BiclusterSizeNormComp, BiclusterVarianceNorm, RowVarianceNormComp, MeanSquaredResidueNorm, BiclusterSizeNumBicsNormComp*, DistanceBetweenBiclustersNormComp*. **Default value:** BiclusterSizeNormComp;MeanSquaredResidueNorm
-summarise -individual -objectives	Method to summarize the overall solution quality from the individual bicluster qualities. Applicable only to COMPLETE. **Valid values:** Mean, HarmonicMean, GeometricMean. **Default value:** HarmonicMean
-population -size	Population size. **Valid values:** Integer. **Default value:** 500
-max -evaluations	Maximum number of evaluations. **Valid values:** Integer. **Default value:** 150000
-str -algorithm	Algorithm to use. Sub-parameters can be specified in brackets after the identifier. **Valid values (Single-objective):** GA-AsyncParallel, GA-SingleThread. **Valid values (Multi-objective):** NSGAII-AsyncParallel, NSGAII-SingleThread, MOEAD-SingleThread, MOCell-SingleThread, SPEA2-SingleThread, IBEA-SingleThread, NSGAIII-SingleThread, MOSA-SingleThread. **Default value:** NSGAII-AsyncParallel
-crossover -probability	Crossover probability. **Valid values:** Decimal between 0 and 1. **Default value:** 0.9
-mutation -probability	Mutation probability. If a progressive mutation is desired, specify a range (e.g. 0.3->0.05). **Valid values:** Decimal or range. **Default value:** 0.1
-crossover -operator	Crossover operator. The operator chosen depends on the representation type. Sub-parameters can be specified in brackets after identifier. **Valid interfaces combination (COMPLETE):** RowPermutationCrossover; BiclusterBinaryCrossover; CellBinaryCrossover. **Valid interfaces combination (PARTIAL):** RowColBinaryCrossover. **Default value (COMPLETE):** PartiallyMappedCrossover; BicUniformCrossover; CellUniformCrossover
-mutation -operator	Mutation operator. The operator chosen depends on the representation type. Sub-parameters can be specified in brackets after the identifier. **Valid interfaces combination (COMPLETE):** RowPermutationMutation;BiclusterBinaryMutation;CellBinaryMutation. **Valid interfaces combination (PARTIAL):** RowColBinaryMutation **Default value (COMPLETE):** SwapMutation; BicUniformMutation; CellUniformMutation
-observers	List of observers separated by semicolons. **Valid values:** BiclusterCountObserver, FitnessEvolutionMinObserver, FitnessEvolutionAvgObserver, FitnessEvolutionMaxObserver, NumEvaluationsObserver. **Default value:** BiclusterCountObserver; FitnessEvolutionMinObserver; NumEvaluationsObserver
-num -threads	Number of threads to use. **Valid values:** Integer. **Default value:** Number of available processors
-output -folder	Output folder path. **Default value:** N/A

objectives aimed at evaluating a single bicluster can be used in the complete encoding, as long as a joint quality summary strategy is specified. Available strategies include the arithmetic mean, geometric mean, and harmonic mean. These strategies penalize individual quality heterogeneity to a lesser or greater degree, respectively. The aim is to prevent a solution from containing biclusters whose overall quality is good solely due to each individual's effort to excel at a specific objective (another flaw of the traditional representation).

The algorithms available in `MOEBA-BIO` are diverse but have been previously used for biclustering purposes [19]. In concrete, it considers: NSGAII [79], NSGAIII [80], MOEA/D [81], MOCell [82], SPEA2 [83], IBEA [78], and MOSA [299]. Other well-known algorithms, such as SMPSO [252], had to be discarded due to their incompatibility with the designed representation, as it does not follow the evolutionary scheme.

Regarding crossover and mutation probabilities, it is worth mentioning that a progressive mutation option has been implemented in addition to providing static values. This common practice favors exploration in the early stages of the evolutionary algorithm and subsequent exploitation in later stages of higher convergence.

As for the crossover and mutation operators, the decision was made to preestablish them in the simplest way possible to avoid interfering with subsequent comparisons of encodings. To this end, the most standard operators for the nature of each part of the encodings were chosen. On the one hand, the partial representation consists of two binary encoding sections, one for rows and one for columns. Therefore, the most common approach is to use the most popular binary crossover (uniform crossover) and the most established binary mutation operator (uniform mutation). On the other hand, something similar has been implemented for the complete encoding. Although this will be explained in more detail in the following subsection, it should be mentioned that the complete encoding is divided into three parts: one permutation and two binary sections. Therefore, the same operators as before are used for the binary sections, while for the permutation, Partially Mapped Crossover and Swap mutation are employed. However, the implementation of new operators is open thanks to a sophisticated interface system, allowing the design of domain-specific operators that can also inject knowledge during the evolution of generations.

Additionally, it is worth noting that a list of observers is available for those who wish to log various aspects of the evolutionary history during execution. If an algorithm supporting multi-threaded execution is specified, limiting CPU usage during the run will also be possible.

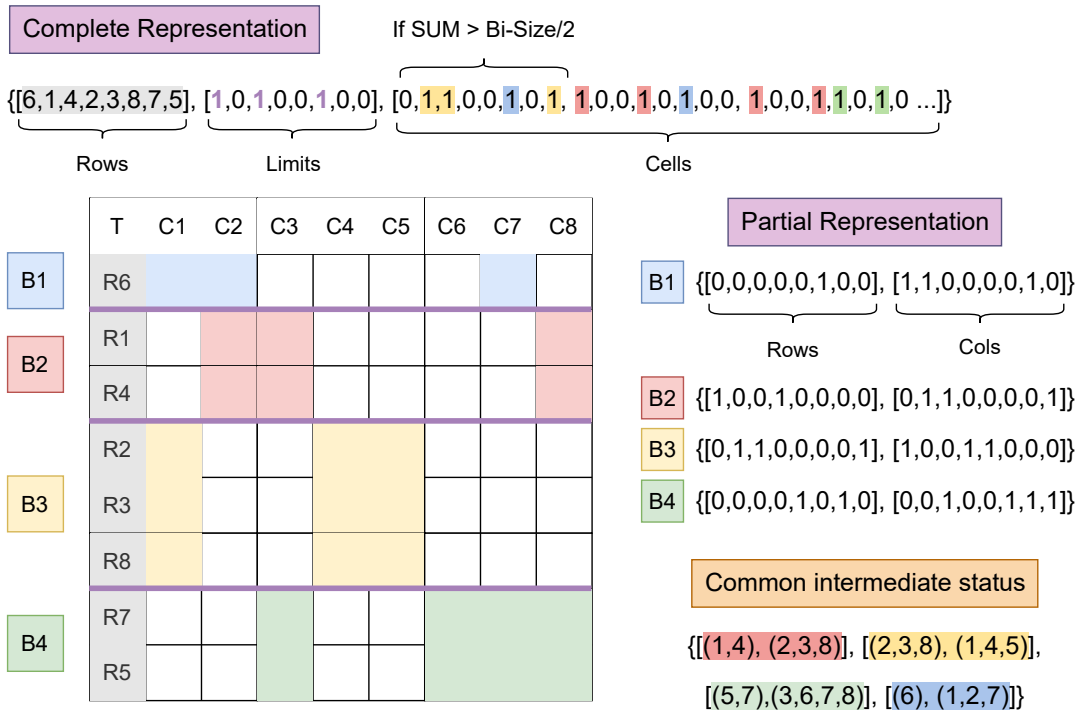


Figure 10.1: An example of the COMPLETE and PARTIAL encodings, as well as their common intermediate state, on a simplified 8x8 matrix. A solution consisting of 4 biclusters with overlapping columns is shown.

Finally, regarding second-level parameters, it should be mentioned that, as explained in several sections of Table 10.1, they pertain to specific configurations of certain first-level options that can be specified directly in parentheses. The most significant cases are algorithm-specific parameters, parameters for certain objective functions, or parameters related to crossover and mutation operators. All of these will be considered during the self-configuration phase.

10.1.1 Representation

MOEBA-BIO introduces a new global perspective encoding, that allow to stablish a direct equivalence between the algorithmic individual and a biclustering solution to the real problem. This is achieved because each individual represents an exhaustive set of biclusters, whose quantity is variable and part of the algorithm's learning process. This perspective helps in designing objective functions that evaluate new aspects of biclustering, as well as realistic functions specialized in the biological domain of the data.

This encoding, named "COMPLETE," is illustrated in Figure 10.1 to facilitate understanding. As can be seen, the encoding requires three parts:

1. **Rows**: The first part of the encoding is a permutation that represents the order in which the different rows of the data matrix are arranged (Column T in Figure 10.1).

2. **Limits**: It consists of a binary vector of the same length as the previous one, specifying the horizontal boundaries of the biclusters (purple lines in Figure 10.1).

3. **Cells**: A binary vector with the same size as the data matrix specifying each cell's activation state. Unlike the previous vector, its interpretation does not depend on the permutation. That is, the first value refers to the activation state of the cell (R1, C1), regardless of its position according to the permutation or which bicluster it belongs to based on the boundaries. If most of the cells in the same column within a bicluster are activated, that column is activated for the bicluster in question. This approach allows for a progressive learning process of column activation and deactivation, where columns with more activated cells are more likely to remain activated. In comparison, those with fewer activated cells are more likely to be deactivated.

The example proposed in Figure 10.1 is a simplified case of 4 biclusters in an 8x8 matrix. As can be seen, all the biclusters in the solution are represented by a single individual in the case of the complete representation, while 4 independent individuals are needed to capture all this information with the partial representation (each individual is a partial solution to the problem).

This representation is also associated with a set of features that align with observations seen in biclustering problems within the biological domain. First, this representation does not allow overlap in one of the two dimensions (if row overlap and not column overlap is desired, the input numeric matrix needs to be transposed). This constraint is also observed in other well-known algorithms, such as Cheng and Church's Algorithm (CCA) [101] or the Binary Inclusion-Maximal Biclustering Algorithm (Bimax) [124], which have been extensively applied to gene expression data.

Although it is not a restriction, this representation tends to group all elements of the non-overlapping dimension. In Figure 10.1, it can be seen that all rows end up belonging to a bicluster. However, if a bicluster deactivates almost all columns or contains only one row (as in B1 in Figure 10.1), the framework itself

will disregard these biclusters, leaving those rows ungrouped [4]. This design decision aims to facilitate coverage of the data matrix and eliminate the risk of repetitive exploitation of the same areas. This is another limitation of the partial representation that was intended to be overcome, where the dominant quality of one bicluster over others draws all partial solutions towards it.

It is clear that handling an individual is more complex and resource-intensive in the case of the complete representation than in the partial one, a fact supported by the amount of information stored in each case. However, to minimize the memory resources required, the Java BitSet class [5] has been used to store the binary vectors. This allows significant space savings without adding computational cost when manipulating individuals.

Although the complete representation is one of the major contributions of this thesis, the `MOEBA-BIO` framework is designed for continuous extension thanks to the flexibility and modularity of its implementation. This means that integrating new encodings into `MOEBA-BIO` is extremely simple [6]. Thanks to the transition through a common intermediate state (also illustrated in Figure 10.1), the evaluation process is completely independent of the encoding of the individuals. Therefore, the only necessary design elements (apart from the encoding itself) are the transition to this intermediate state (i.e., the interpretation of the representation) and the crossover and mutation operators, which can initially be formed by combining operators already implemented in JMetal [276].

10.1.2 Objectives

The fitness functions currently available in `MOEBA-BIO` cover traditional individual-focused objectives and global vision objectives of a generic nature for biclustering. To ensure compatibility between the traditional individual-centered objectives and the new framework and representation, several strategies have been implemented to summarize individual qualities. Some of these strategies include penalties that account for heterogeneity among those qualities (see Table 10.1).

In order to use the full range of available algorithms, all fitness functions must be normalized between 0 and 1. Therefore, traditional biclustering objectives had to be reformulated to fit this range, made possible by the prior nor-

[4] Java class available in: `https://github.com/AdrianSeguraOrtiz/MOEBA-BIO/blob/main/src/main/java/moeba/fitnessfunction/FitnessFunction.java`

[5] `https://docs.oracle.com/javase/7/docs/api/java/util/BitSet.html`

[6] Java class available in: `https://github.com/AdrianSeguraOrtiz/MOEBA-BIO/blob/main/src/main/java/moeba/representationwrapper/RepresentationWrapper.java`

Algorithm 16 Adaptive bSize fitness function.

Require: Bicluster b, Set of all biclusters less b B, Data matrix D, Coherence weight α, Row weight β
Ensure: Value of the fitness function $score$
1: $maxSize \leftarrow rows(D) \times columns(D)$
2: $bSize \leftarrow rows(b) \times columns(b)$
3: $numBiclusters \leftarrow size(B) + 1$
4: $parcelSize \leftarrow maxSize/(numBiclusters^2)$
5: $normWeightedSize \leftarrow calcNormSize(b, D, \beta)$
6: $sizePenalty \leftarrow min(1, abs(parcelSize - bSize)/parcelSize)$
7: $score \leftarrow (1 - \alpha) \times normWeightedSize + \alpha \times (1 - sizePenalty)$
8: **return** $score$

malization of the numerical data. Furthermore, since MOEBA-BIO is an extension of JMetal, all objectives must be oriented towards minimization. However, for individual quality summary strategies like the harmonic or geometric mean to be effective, the individual qualities of the biclusters must be oriented towards maximization. As a result, the formulas for individual biclusters are first oriented towards maximization, and after obtaining the joint quality, the complementary value of the result is calculated.

Since traditional objectives for biclustering have been extensively discussed in the literature [19], the subsequent sections focus on the implementations of the new objectives proposed here.

Adaptive bicluster size

This fitness function is designed to evaluate the size of the biclusters in relation to the total number of biclusters in a solution. It combines two main components: the normalized size of the biclusters and an adaptive penalty based on the difference between the actual size of a bicluster and a realistic maximum size, dynamically adjusted according to the number of biclusters in the solution. The balance between these two factors is controlled by a configurable coherence weight, allowing adjustment of the relative importance of size versus homogeneity within the set of biclusters. This encourages solutions that not only maximize the size of the biclusters, but also maintain structural coherence based on the total number of generated biclusters. Unlike the traditional *Bicluster Size (bSIZE)*, it has a complete problem perspective by accounting for the number of biclusters, so its convergence is not focused on obtaining a single bicluster that covers the entire data matrix.

The implementation of this fitness function is represented in the pseudocode shown in Algorithm 16. First, the maximum size a bicluster can have within the data matrix and the actual size of the bicluster under evaluation are calculated (lines 1 and 2 in Algorithm 16). Next, the recommended maximum size of a bicluster is calculated, which is obtained by dividing the maximum size of the matrix by the square of the number of biclusters (lines 3 and 4 in Algorithm 16). This allows the penalty to be dynamically adjusted according to the total number of biclusters present in the solution.

The weighted normalized size of the bicluster is obtained using the traditional bSize function, which takes into account both, the size of the bicluster and the weight assigned to the rows and consequently to the columns (line 5 in Algorithm 16).

Additionally, a penalty is calculated based on the difference between the actual size of the bicluster and the recommended maximum size (line 6 in Algorithm 16). This penalty adapts the size of the bicluster according to the total number of biclusters in the solution.

Finally, the fitness function score is calculated by combining the normalized size of the bicluster and the size penalty, weighted by the coherence parameter α (line 7 in Algorithm 16). The function returns this value as the final fitness result for the evaluated bicluster (line 9 in Algorithm 16).

Bicluster differentiation

Leveraging the no-row-overlap constraint, this objective focuses on ensuring that each bicluster is correctly defined, without excluding rows that fit better with the averages of their columns than with others, nor including rows that are more aligned with other biclusters. The function compares the bicluster under evaluation with the nearest one, determined by the number of shared columns.

For each row of the nearest bicluster, two distances are calculated: one to the values of the evaluated bicluster and another one to the values of its own bicluster (adjusted to reflect the exclusion of the row in question). If the distance from the row to the evaluated bicluster is less than that of the adjusted averages of its own bicluster, the evaluated bicluster is penalized for excluding an aligned row.

This function rewards biclusters whose rows are better represented within them than in other biclusters, improving differentiation between biclusters and the overall coherence of the solution. Thus, the biclusters reflect more accurately

Algorithm 17 Bicluster Differentiation fitness function.

Require: Bicluster b, Set of all biclusters less b B, Data matrix D
Ensure: Value of the fitness function $score$

1: $closest \leftarrow findClosest(b, B)$
2: $meanB \leftarrow calcMean(b)$
3: $meanClosest \leftarrow calcMean(closest)$
4: $totalScore \leftarrow 0$
5: **for** each row in $closest$ **do**
6: $\quad riClosest \leftarrow getCells(row, cols(closest))$
7: $\quad riB \leftarrow getCells(row, cols(b))$
8: $\quad distBMean \leftarrow calcDist(riB, meanB)$
9: $\quad adjMeanClosest \leftarrow calcAdjMean(meanClosest, riClosest)$
10: $\quad distAdjMean \leftarrow calcDist(riClosest, adjMeanClosest)$
11: $\quad fitScore \leftarrow distBMean/(distBMean + distAdjMean)$
12: $\quad totalScore \leftarrow totalScore + fitScore$
13: **end for**
14: **return** $totalScore/size(closest)$

the patterns of the rows they contain, contributing to greater biclustering precision without encouraging size reduction (as seen in other traditional functions).

The implementation of this fitness function is represented in the pseudocode shown in Algorithm 17. First, the bicluster closest to the one under evaluation is identified, using the number of shared columns as the criterion (line 1 in Algorithm 17).

Next, the column averages for both the evaluated bicluster and the nearest bicluster are calculated (lines 2 and 3 in Algorithm 17). These averages are essential for comparing how well the rows of the nearest bicluster fit their own average compared to the average of the evaluated bicluster.

In the loop (lines 5-12 in Algorithm 17), the rows of the nearest bicluster are iterated over. For each row, the values of the columns in both, the nearest bicluster and the evaluated bicluster are retrieved (lines 6 and 7 in Algorithm 17). Then, the distance of these values to the averages of the evaluated bicluster is calculated (line 8 in Algorithm 17), along with the distance to the adjusted averages of the nearest bicluster, which reflects the exclusion of the given row (lines 9 and 10 in Algorithm 17).

The fitness score is calculated as the ratio of the distance from the row to the evaluated bicluster to the sum of both distances (line 11 in Algorithm 17).

This metric indicates whether the row fits better in the evaluated bicluster or the nearest bicluster. If the distance to the evaluated bicluster's averages is smaller, the evaluated bicluster is penalized with a lower score, accumulating this value into the total (line 12 in Algorithm 17).

Finally, the total score for the evaluated bicluster is obtained as the average of the fit scores calculated for each row of the nearest bicluster (line 13 in Algorithm 17).

10.1.3 Parameter autoconfigurator

In addition to the new representation and the objectives designed thanks to its global perspective, `MOEBA-BIO` also facilitates the injection of domain-specific biomedical knowledge through its parameter self-configurator. Parameter self-configuration using wrapper evolutionary algorithms has already been validated in other proposals in the literature [300]. However, in this case, a self-configurator has been specifically designed for the biclustering problem when applied to particular domains.

Figure 10.2 shows the diagram of the proposed self-configurator. It consists of two simple genetic algorithms instantiated with the most common parameter values, which wrap the evolutionary algorithm to be configured. It takes as input a series of data matrices associated with a specific biomedical context, the files with the reference biclusters for each matrix (gold standards), a file with the supervised parameters to be configured, and another for the unsupervised parameters, specifying the set of possible values for each one [7]. As output, it provides the best parameter configuration obtained. Additionally, tracking files are provided for the evolution of the external wrapper population and the evolution of the internal wrapper population corresponding to the external winner.

- **Supervised Phase**: This corresponds to the outer wrapper genetic algorithm (the green one in Figure 10.2). In this phase, different combinations of objectives (with varying numbers of dimensions as long as $n \geq 2$) and values of their subparameters are tested to determine which configuration minimizes a validation metric related to the gold standards of the benchmark representing the biomedical domain of the data. However, since different configurations at this level imply different search spaces with varying numbers of dimensions, it would be unwise to set the same values for the other parameters in all cases. Therefore, given a configuration at this level, it is necessary to extract the best values for the remaining

[7] Configuration file examples in: `https://github.com/AdrianSeguraOrtiz/MOEBA-BIO/tree/main/parameterization`

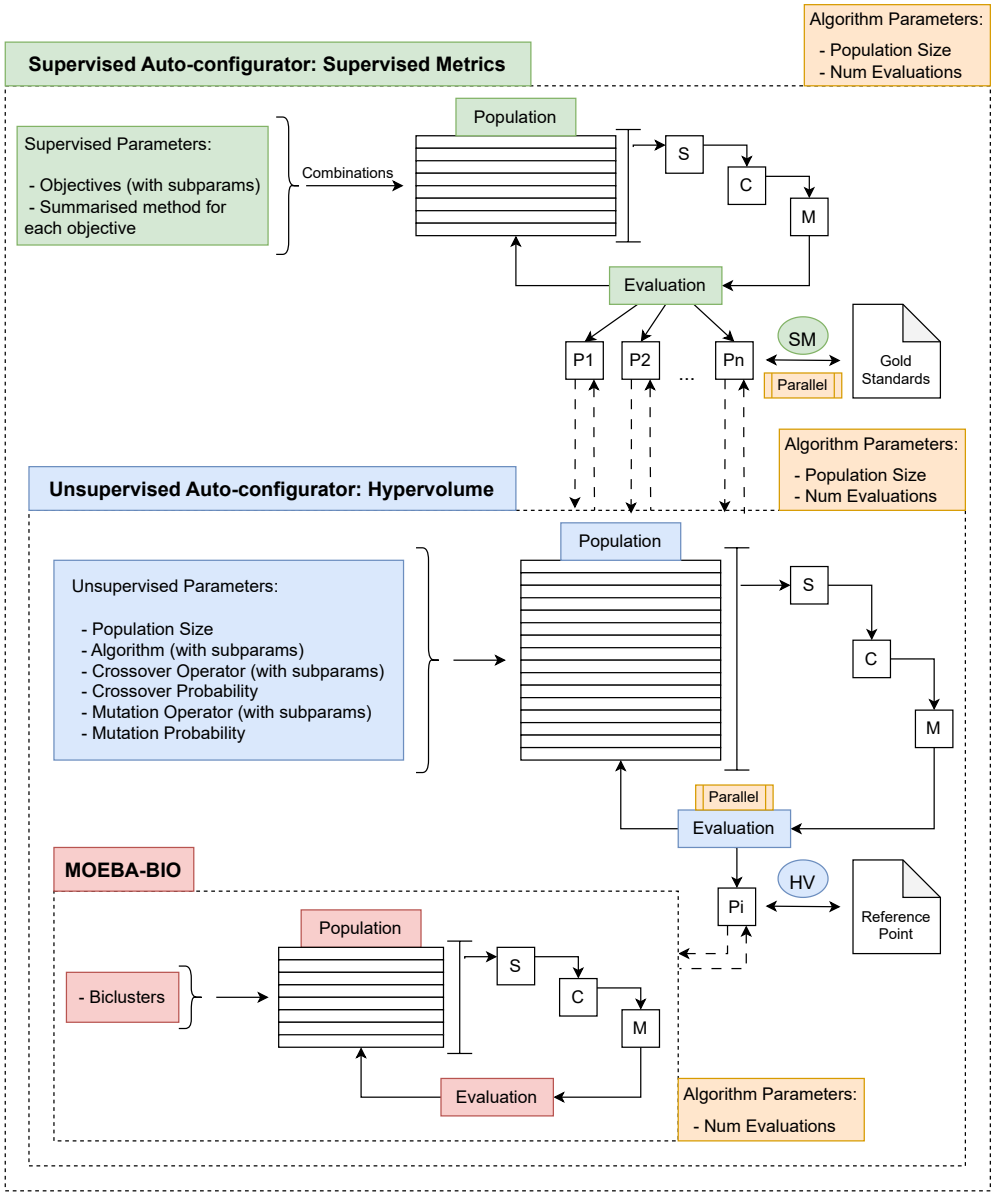


Figure 10.2: Structure of the specific self-configurator in the MOEBA-BIO framework. It consists of two wrapper genetic algorithms. The outer one handles the self-configuration of objectives through a supervised evaluation that depends on the gold standards of the input data. The inner wrapper handles the self-configuration of the remaining parameters, based on unsupervised metrics such as hypervolume.

parameters in the next phase before proceeding with its evaluation. The evaluation is performed after obtaining the best individual from the unsu-

pervised phase. For each data matrix, the 5 best individuals obtained on the MOEBA-BIO Pareto approximation front according to the selected supervised metric are taken, and the final score is returned as the average across all matrices.

- **Unsupervised Phase**: This corresponds to the inner wrapper genetic algorithm (the blue one in Figure 10.2). Once the objectives of the problem and their corresponding subparameters have been set in the external individual, this phase optimizes the values of the remaining parameters, including crucial ones, such as: the algorithm to use with its corresponding subparameters, the operators, the probabilities for each phase, the population size, and more. This optimization, now performed on comparable Pareto fronts, can be based on standard unsupervised metrics such as hypervolume, which, given that all objectives are normalized between 0 and 1, can use the value 1 in each dimension to form its reference point. Again, for each data matrix, the metric to be minimized is calculated, and the average across all matrices is returned.

It should be noted that, despite its stochastic nature, no repetitions have been implemented for the internal execution of the algorithm being configured for each parameter combination. This decision is based on the fact that, due to the redundancy of individuals in such small search spaces, it is considered that the potential variations dependent on the executions can be effectively covered by this redundancy. This measure has been adopted to ensure the computational feasibility of the self-configurator, and its validity will be demonstrated in the subsequent experimentation.

To clarify the evaluation process carried out by the self-configurator, Figure 10.3 illustrates an example of how MOEBA-BIO evaluates each candidate configuration. Specifically, the figure depicts how each outer individual defines a combination of objective functions and subparameters, which is then internally assessed by an inner evolutionary layer using different algorithmic setups. The final quality of each outer individual is computed based on the performance of the inner layer across multiple datasets.

It is worth noting that both evolutionary algorithms that make up the self-configurator have been designed as single-objective optimization processes. The algorithm in the external (supervised) phase is guided exclusively by a gold standard-dependent metric (specifically, the clustering error), while the algorithm in the internal (unsupervised) phase uses only the hypervolume as its objective function. Therefore, the self-configurator does not involve any weighting or prioritization among multiple objectives, nor does it need to resolve conflicts

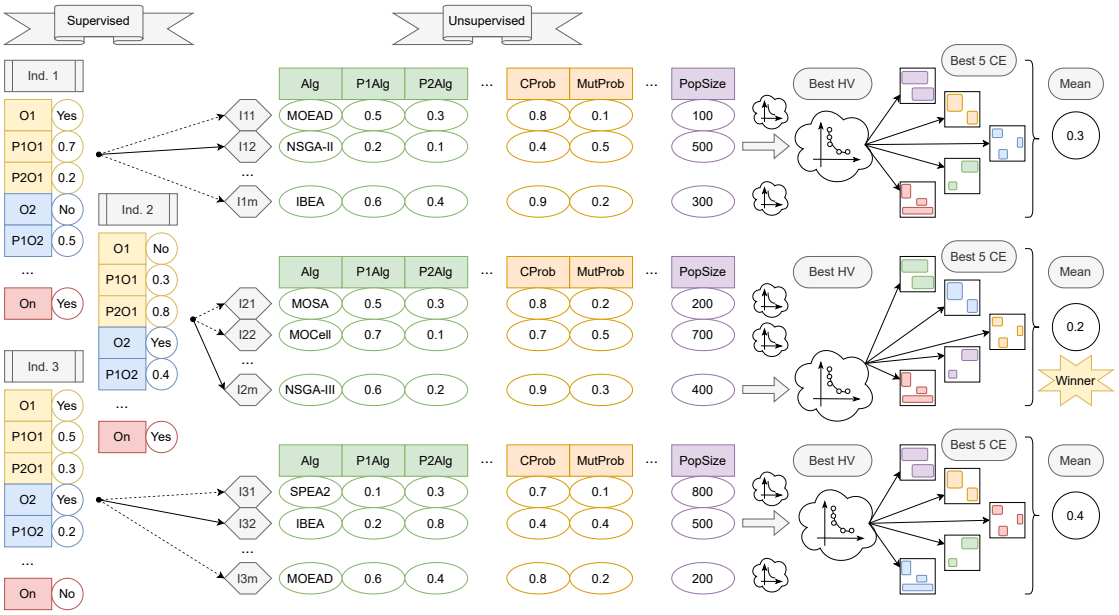


Figure 10.3: Visual representation of the MOEBA-BIO self-configuration mechanism. The process is structured in two nested evolutionary phases: a supervised outer layer (left) and an unsupervised inner layer (right). Each individual in the outer population encodes a specific combination of objective functions and subparameters. For each one, the inner layer determines the best technical configuration (algorithm, operator probabilities, population size, etc.) based on hypervolume (HV). The final evaluation of each outer individual is computed by averaging the clustering error (CE) of the top 5 Pareto-optimal solutions obtained for each dataset. The best outer individual (i.e., configuration) is selected as the winner.

between them. Its role is to identify the combination of objectives (and associated parameters) that yields the best results in a supervised context, optimizing the remaining parameters in an unsupervised context. This structure helps maintain interpretability throughout the process and facilitates its adaptation to other domains where the set of objectives or evaluation metrics may vary.

10.1.4 Methodological Comparison

Once all the methodological components of MOEBA-BIO have been presented, it is appropriate to contrast them with the most closely related state-of-the-art approaches that also use non-traditional encodings: BI-MOCK, PBD-SPEA2, and BiClustSMEA.

Unlike these approaches, MOEBA-BIO not only modifies the representation of individuals, but also proposes a complete evolutionary framework that enables

Table 10.2: Methodological and implementation-level comparison of MOEBA-BIO with representative non-traditional encoding proposals.

Implementation aspect	BI-MOCK	PBD-SPEA2	BiClustSMEA	MOEBA-BIO (this proposal)
Control over number of biclusters	Automatically determined thanks to a newly proposed variable string length encoding scheme	User-defined, must be specified.	Generated during execution using a random variable	The number of biclusters is variable and is fully self-learned via evolutionary convergence
Aggregation of fitness across biclusters	Simple mean of the MSR and bSize over the set of biclusters encoded in the solution	Fitness is likely evaluated per bicluster	Mean across all δ-biclusters present in a chromosome for objectives like MSR, row variance and volume	Harmonic amn Geometric mean available to summarize individual bicluster qualities. Penalized heterogeneity.
Final solution selection	A post-processing step is proposed to select the set of final biclusters among Pareto front individuals.	Post-processing across individuals; biclusters from different individuals might be combined.	Sequential selection from the Pareto optimal set based on the lowest MSR value.	Each individual in the Pareto front represents a complete solution to the biclustering problem; no aggregation required.
Extensibility of objective functions	Not explicitly designed for easy extension based on the description.	Likely hard-coded within the algorithm's structure.	Requires modification to the algorithm's core to incorporate new objectives.	Plugin-based, fully decoupled objective function design; researchers have complete freedom to implement new objectives or use predefined ones.
Domain integration capacity	Only traditional objectives like bSize and MSR are mentioned.	Not considered, homogeneity and size are used as primary objectives.	Focuses on general quality metrics.	Native support for domain-specific global objectives (e.g., Regulatory Coherence for gene co-expression).
Open source / Reproducibility	No public availability.	No public availability.	No public availability.	Yes (publicly available framework): `https://github.com/AdrianSeguraOrtiz/MOEBA-BIO`.

the representation of complete solutions, the definition of objectives from both a global perspective and an application-specific standpoint, and the application of structural penalization mechanisms. In addition, it integrates both supervised and unsupervised self-configuration systems, allows for decoupled objective design, and facilitates its extension to other biomedical contexts.

To highlight these differences beyond the encoding itself, Table 10.2 summarizes the key methodological and implementation-related aspects that distinguish MOEBA-BIO.

10.2 Experimentation

The experimentation in this study aims to validate the new complete problem encoding proposed in this chapter and the general context objective functions implemented thanks to it. To achieve this, conducting a fair, rigorous comparison within the most generic context is necessary.

Recent proposals in the literature that used a partial representation were analyzed to conduct this comparison. After reviewing the recent survey [19], which focuses on analyzing all biclustering algorithms with a multi-objective evolutionary approach, a list of 22 candidates was obtained: TSTP [301], BOBEA [160], MMco-Clus [302], β-SMOB [158], AMOSAB-PS [156], BP-NSGA2 [155], PBD-SPEA2 [171], AMOSAB [161], SPEA2B-δ [159], HMOBI [303], SMOB-VE [168], MOBI [163], SPEA2B [304], MOGAB [305], MOM-aiNet [306], MOA-COB [307], AMOPSOB [308], CMOPSOB [309], MOPSOB [310], MOFBA [311], SMOB [312] and MOEAB [58]. However, since none of the papers on these algorithms provide public access to their software, a partial representation was integrated into `MOEBA-BIO` exclusively for experimental purposes. This approach enables an alternative yet fair comparison.

The first step is to set the objectives to be used during the comparison. In [19], it is shown that the most commonly used combination is Mean Squared Residue (MSR) and Bicluster Size (bSIZE). Therefore, these two functions will be established as the main objectives in this phase of experimentation. The goal is to evaluate each contribution separately, resulting in 5 configurations:

- Partial (bSIZE + MSR).
- Complete (bSIZE + MSR).
- Complete replacing bSIZE by Adaptive bSIZE with a coherence weight value of 0.25 (Adaptive bSIZE + MSR).
- Complete adding bDIFF (bSIZE + MSR + bDIFF).
- Complete representation but adding both new objective functions at the same time (Adaptive bSIZE + MSR + bDIFF).

Both complete (proposal) and partial (traditional) encoding configurations are compared within the same `MOEBA-BIO` framework using the standard operators for each representation. To provide a fairly diverse and flexible genetic content in this first experimental environment, the population size has been set at 500 individuals. Regarding the rest of the parameters, all have been set to their most common and reasonable values: the chosen algorithm is NSGAII [79], with a total of 100,000 evaluations, a crossover probability of 90%, and a slightly

higher than normal mutation probability set at 10% due to the size of the data matrices.

The dataset used corresponds to the matrices described in section 4.2.1 that were artificially generated by the G-bic tool after giving different values to a given set of parameters.

When running `MOEBA-BIO` with each configuration, an approximate Pareto front is obtained for each input matrix. In the case of the partial configuration, the entire front is combined to form the real solution to the problem. For the complete encoding configurations, each individual in the front can be directly compared with the gold standard. Therefore, for the partial representation, the quality score is the comparison between the solution formed by the entire front and the gold standard, while for the complete configurations, the median of the quality scores from the front is calculated. This allows for the subsequent calculation of a Friedman statistical ranking complemented by Holm's non-parametric tests [313] to provide statistical rigor in the comparison of configurations.

Regarding the evaluation of the biclustering results, the clustering error [314] has been selected as the main comparison metric across biclustering algorithms. This is a robust metric that evaluates the complete biclustering solution by penalizing both the redundancy between biclusters and the difference between the number of inferred biclusters and the actual number in the gold standard. It is also a metric that has been used in many recent biclustering studies as well as in the context of evaluating evolutionary biclustering algorithms [19]. The clustering error should be minimized such that a lower value indicates a better biclustering solution. Therefore, for this phase of experimentation and the subsequent statistical significance analysis, the complementary value of the result is calculated.

This evaluation strategy ensures that the benefits obtained from the complete configuration compared to the partial one are solely due to the encoding of the individuals. Furthermore, it allows for the determination of whether each new objective individually improves upon the base use of the new encoding and whether their combined use provides greater benefits than each individual.

Additionally, notice that other biclustering metrics such as Recovery, Relevance [315] and Ayadi's score [316] are also considered in our further experiments as they are widely used in the specialized literature.

Finally, it is worth mentioning that the validation of the constructed autoconfigurator will be addressed in the next chapter during the self-construction of a specific algorithm for the field of gene co-expression.

Table 10.3: Friedman mean rank with Holm's adjusted p values (0.05) for Clustering Error.

Clustering Error		
Technique	$Friedman's\{Rank\}$	$Holm's\{Adj-p\}$
***Complete-AdapBS(0.25)+BDiff**	**1.42**	-
Complete+BDiff	2.29	6.71e-03
Complete-AdapBS(0.25)	2.85	1.69e-05
Complete	3.44	1.15e-09
Partial	5.00	4.88e-28

10.3 Results and discussion

Table 10.3 shows the results of applying a Friedman statistical ranking with Holm's non-parametric tests to the clustering error values obtained after running `MOEBA-BIO` on the entire benchmark of 48 matrices generated by G-Bic, using each of the configurations to be compared. As can be seen, the configuration associated with the partial encoding ranks last, resulting in a lower rank than the complete encoding when none of the new objectives is applied. In fact, aside from this distance in the overall ranking, a Wilcoxon test between these two configurations yields a p-value of 7.11e-15, which is significantly below the commonly accepted threshold of 0.05. This allows us to affirm that, under the same experimental conditions (traditional objectives, environment, data, and basic crossover and mutation operators), the use of the complete encoding offers a statistically significant improvement in result accuracy compared to the traditional encoding.

Additionally, by observing the intermediate positions in the Friedman ranking, it can be affirmed that both, the inclusion of bDIFF and the replacement of the traditional bSIZE function with Adaptive bSIZE separately outperform the basic complete configuration. In other words, both functions individually contribute to improving the accuracy of the results. Finally, the top-ranked configuration confirms that the new objectives, besides offering benefits individually, provide even better improvement when used together, and this improvement is also statistically significant compared to the other configurations in the ranking, according to Holm's non-parametric tests.

With all these observations, it can be confirmed that the complete encoding not only increases accuracy on its own, but also enables the inclusion of holistic perspective knowledge which, even in a general context, has demonstrated further improvements in the algorithm's results. However, to better understand the reasons behind these results, a deeper analysis has been conducted to ensure that the success of this encoding is due to the premises that initially motivated

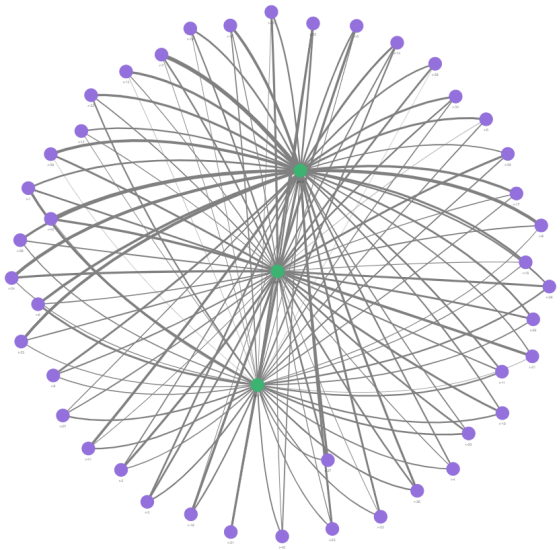
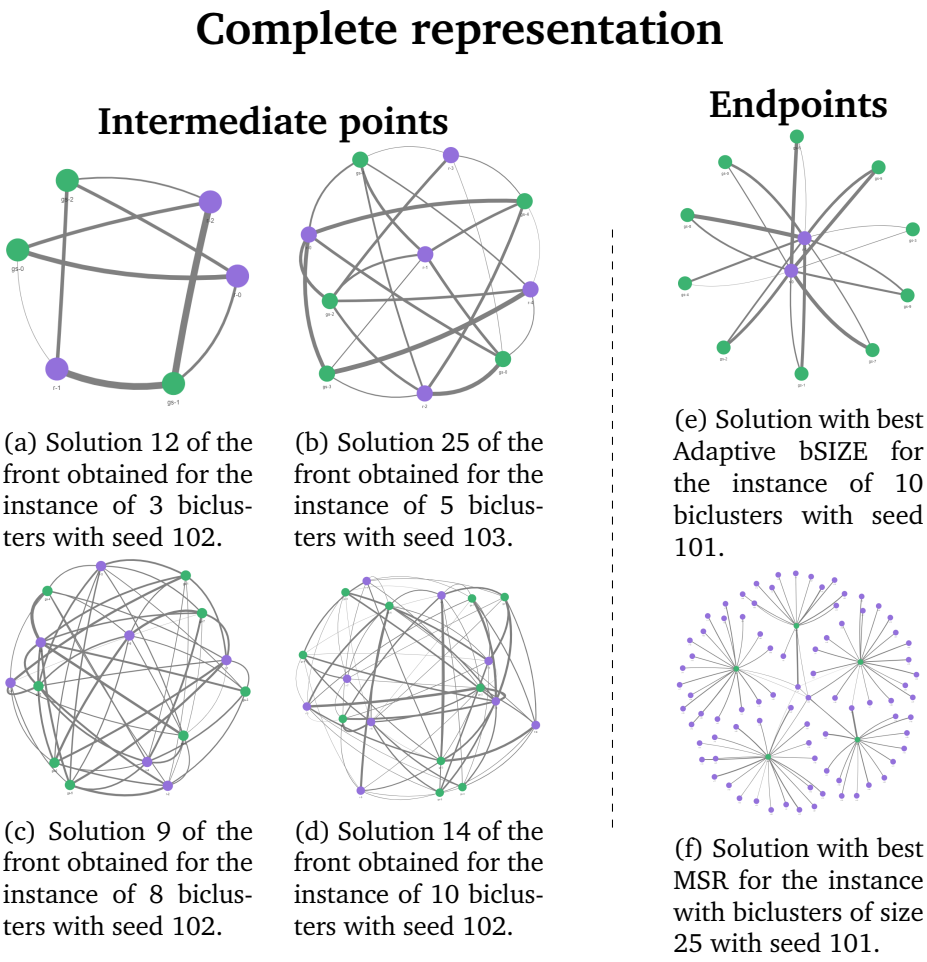


Figure 10.4: Biclustering solution obtained by MOEBA-BIO using the partial representation for the simulated dataset with 3 biclusters and seed 102. The green nodes represent the biclusters from the gold standard. Each purple node in this encoding represents a partial solution from the front obtained by the algorithm since, in this case, the combination of all biclusters in the front constitutes the real biclustering solution to the problem. The graph's edges refer to the intersection between biclusters, with their thickness increasing in proportion to the number of shared rows and columns. To avoid adding noise to the graph, intersections between nodes of the same color caused by possible overlaps between biclusters have been ignored.

its implementation.

In this regard, the first motivation behind the complete encoding was to overcome the fact that the partial representation tends to converge towards a large number of redundant biclusters, which also exhibit significant heterogeneity in their qualities. To verify this, solutions obtained using the partial encoding and winning complete configuration are shown in Figure 10.4 and Figure 10.5, respectively, compared to the corresponding gold standard. In concrete, Figure 10.4 shows a solution from the partial encoding (obtained as the union of all individuals in the front), where the number of biclusters clearly exceeds the actual number of biclusters in the data matrix. This redundancy is not typically penalized by traditional supervised metrics [19], where each bicluster in the gold standard is only compared to the most similar bicluster in the front. In a real-world scenario, where there is no prior information about the biclusters, this



(a) Solution 12 of the front obtained for the instance of 3 biclusters with seed 102.

(b) Solution 25 of the front obtained for the instance of 5 biclusters with seed 103.

(c) Solution 9 of the front obtained for the instance of 8 biclusters with seed 102.

(d) Solution 14 of the front obtained for the instance of 10 biclusters with seed 102.

(e) Solution with best Adaptive bSIZE for the instance of 10 biclusters with seed 101.

(f) Solution with best MSR for the instance with biclusters of size 25 with seed 101.

Figure 10.5: Examples of solutions obtained by `MOEBA-BIO` during the first phase of experimentation using the complete representation and the new objective functions of adaptive bicluster size and bicluster differentiation. The interpretation of the graphs is the same as discussed in Figure 10.4, except that in this case, all purple nodes belong to a single solution from the front obtained by the algorithm, as in the complete encoding, a solution from the front is equivalent to a real solution to the problem. On the left side of the figure, solutions that are balanced across the three objectives are presented. On the right side, extreme solutions from the front are shown, specifically one for the case of extreme optimization of Adaptive Bicluster Size and another for the case of extreme optimization of MSR.

excessive number of redundant clusters would add counterproductive noise.

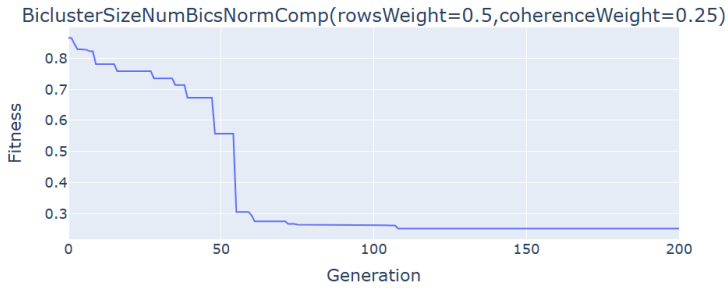
Second, Figure 10.5 presents several solutions from the complete encoding. On the left side of the figure, solutions with an intermediate position in the Pareto front are shown. This position, defined as a coherent balance between

objectives, seems to correspond to solutions with a number of biclusters that are quite similar to the real number. This observation leads to two conclusions: the number of biclusters is part of the algorithm's learning process (which does not happen with the partial encoding), and the noise caused by excessive and redundant biclusters is clearly minimized.

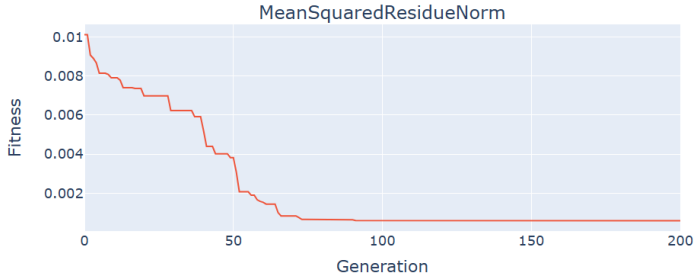
However, it remains to demonstrate the ability of the complete encoding to ensure that the biclusters exhibit qualities consistent with the position in the front of the solution to which they belong. This demonstration is shown on the right side of Figure 10.5. In Figure 10.5e, the biclusters obtained from a solution located at the extreme of the Pareto front with the best optimization of the Adaptive bSIZE objective are displayed. Since the coherence weight has been set to a value of 0.25, the normalized size of the biclusters still carries more weight. This implies that, although the convergence of this function does not directly lead to a single bicluster occupying the entire matrix, the optimization of this function still tends toward large biclusters with low coherence. This is what is observed in Figure 10.5e, where the solution with the best Adaptive bSIZE indeed contains few large biclusters. Meanwhile, Figure 10.5f shows a solution with the best MSR value in its front. Similarly to what was observed with Adaptive bSIZE, the solution contains biclusters consistent with its position in the front, in this case, a large number of small biclusters with high internal coherence.

In addition to analyzing the algorithm's results, it is crucial to examine the evolution of the populations of individuals during execution. For this reason, Figure 10.6 shows the evolution of the different objectives for a specific run of the winning complete configuration. Specifically, it displays the minimum value found in each generation for each fitness function. In Figure 10.6a, we can observe how the Adaptive bSIZE objective, being the simplest one and independent of the data content, exhibits a more abrupt and decisive learning curve. Meanwhile, in Figure 10.6b, although MSR converges at a point quite similar to the previous objective, its learning process seems more gradual and progressive. Finally, the complexity of the new global perspective objective, bDIFF, is shown in Figure 10.6c, where its learning process requires a larger number of generations to reach convergence. Nevertheless, it has already been demonstrated that including this objective provides significant benefits to the algorithm.

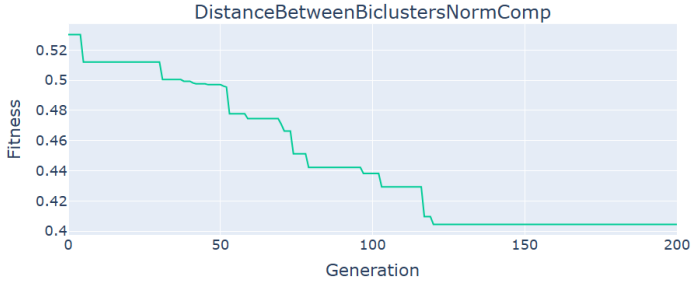
Finally, Figure 10.7 represents, for the same run as in Figure 10.6, the distribution of the number of biclusters contained in the individuals for each generation. This allows the definitive conclusion that the number of biclusters is part of the algorithm's learning process. Despite not being an explicit objective of the algorithm (as the correct number of biclusters is unknown), the curve pre-



(a) Minimum values per generation for Adaptive bSIZE.



(b) Minimum values per generation for MSR.



(c) Minimum values per generation for bDIFF.

Figure 10.6: Evolution of the minimum fitness values of each objective during the MOEBA-BIO run using complete representation and the new objective functions on the simulated dataset with an overlap level of 20 and seed 101.

sented by the distributions clearly mirrors the curves traced by the evolution of the different objectives. In other words, there is a clear direct learning process, whereby the joint optimization of objectives translates into convergence in the number of biclusters present in the individuals of the population.

For each instance, the convergence of the different objectives and the implicit learning of the number of biclusters leads to an approximate Pareto front. These fronts provide the domain expert with various complete solutions to the biclustering problem on the input dataset. Figure 10.8 shows different Pareto fronts obtained by MOEBA-BIO in the first phase of experimentation with the new encoding and proposed objectives. As can be seen, the shape and distribution of

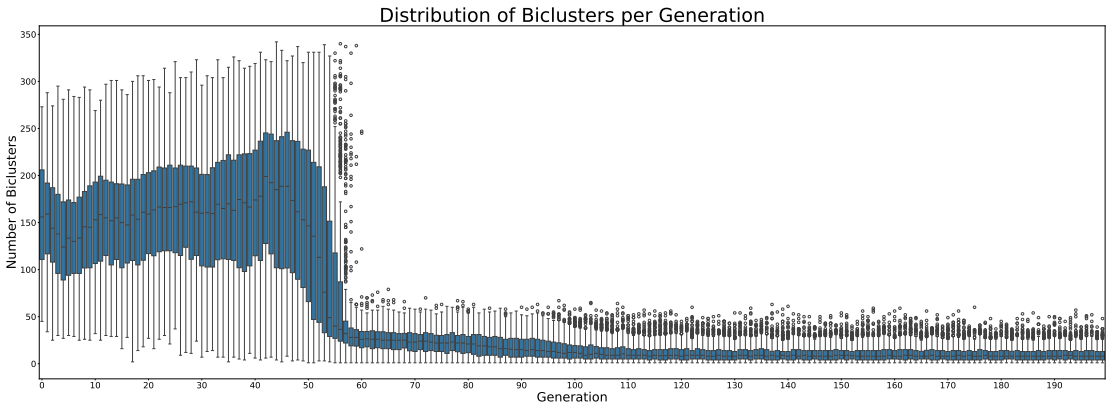
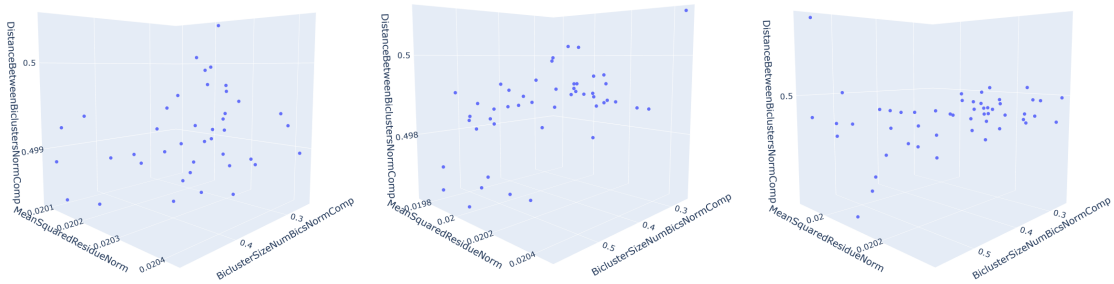


Figure 10.7: Distribution of the number of biclusters contained in each individual per generation in the MOEBA-BIO run using complete representation and the new objective functions on the simulated dataset with an overlap level of 20 and seed 101.



(a) Noise level of 20 and seed 102.

(b) Num of biclusters set to 3 and seed 103.

(c) Overlap level of 20 and seed 102.

Figure 10.8: Set of non-dominated solutions in Pareto front approximation from the final population obtained by MOEBA-BIO using complete representation and the new objective functions on the simulated dataset.

solutions vary across different instances.

Chapter 11

MOEBA-BIO-CoExp: Context-guided evolutionary biclustering for gene co-expression analysis

This chapter introduces MOEBA-BIO-CoExp, a specialized algorithm derived from the MOEBA-BIO framework for addressing the specific challenges of biclustering in the context of gene co-expression. While the previous chapter presented MOEBA-BIO as a general-purpose evolutionary framework for biomedical biclustering, this chapter highlights how its modular and extensible design enables adaptation to domain-specific problems. In particular, the integration of biological knowledge related to regulatory networks is achieved through the implementation of a new global objective: Regulatory Coherence.

The aim of this chapter is twofold: (1) to illustrate the process of auto-constructing a domain-adapted algorithm by leveraging MOEBA-BIO's self-configuration capabilities, and (2) to evaluate the resulting specialized algorithm within a relevant biological context, specifically the detection of co-expressed gene groups.

11.1 Methods

To auto-construct a new biclustering algorithm tailored to gene co-expression analysis, the MOEBA-BIO framework was extended with a novel global objective function named Regulatory Coherence. This function evaluates the biological plausibility of the inferred biclusters in terms of their consistency with the underlying gene regulatory network (GRN) structure.

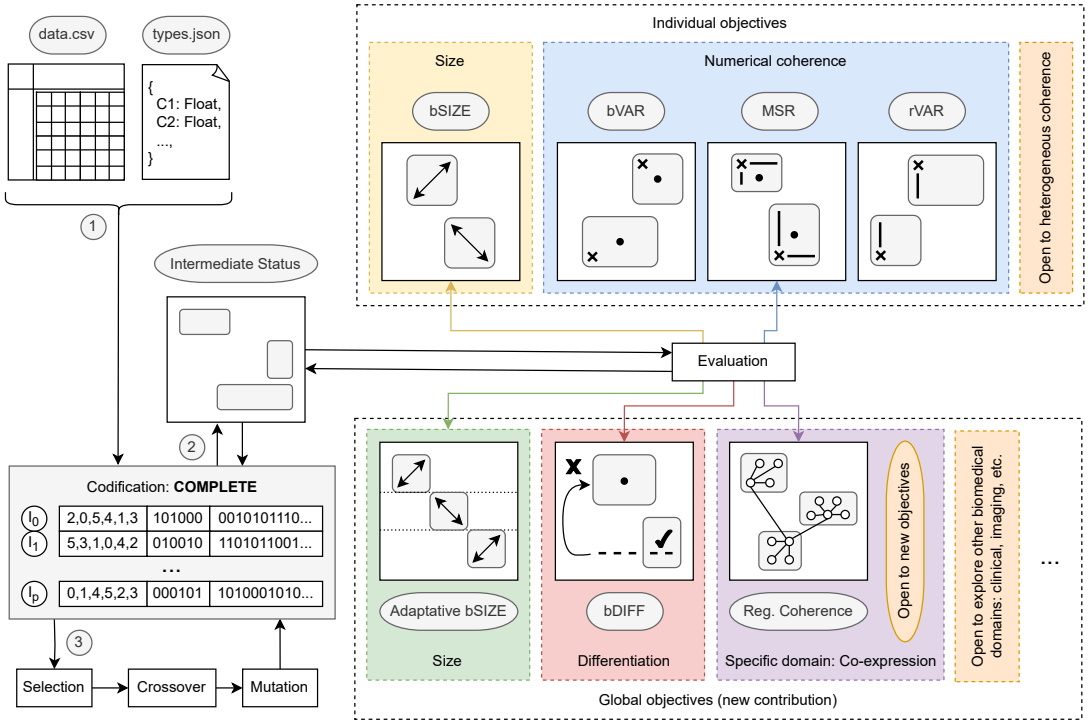


Figure 11.1: The extended framework of MOEBA-BIO outlines its main phases and contributions. It starts with the input of the data matrix and the specification of the type of each column (1). Based on the selected encoding strategy (in this case, COMPLETE), MOEBA-BIO initializes the population for the evolutionary algorithm. The execution then proceeds through an iterative process until the stopping condition, defined as reaching a maximum number of evaluations, is met. Each individual is translated into an intermediate representation common to all encodings (2), enhancing interpretability and facilitating evaluation. Individuals are assessed according to the selected objective functions, with normalized fitness scores ranging from 0 (best case) to 1 (worst case). These scores are stored in the original individual, after which the standard evolutionary steps (selection, crossover, and mutation) are applied (3), depending on the chosen algorithm and the operators defined during parameter configuration.

Figure 11.1 presents the updated MOEBA-BIO architecture for this specialization. The process begins with the input of a gene expression matrix and the specification of each column's data type. After initializing a population using the COMPLETE representation, each individual is translated into an intermediate structure that decouples the evaluation logic from the representation logic. Fitness values for each individual are computed using both general-purpose and domain-specific objectives. The evolutionary algorithm then iteratively performs selection, crossover, mutation, and replacement, guided by these fitness scores.

In this chapter, the new Regulatory Coherence objective was added to this process to guide the learning toward biologically coherent bicluster structures in the context of gene regulation.

11.1.1 New objective: Regulatory coherence

This fitness function focuses on evaluating the regulatory coherence of biclusters within the context of gene co-expression, a specific biomedical domain. Unlike previous fitness functions that evaluate each bicluster (whether they have a global perspective of the solution or not), Regulatory Coherence assigns a global score to the complete solution without the need for summary strategies. This function aims to demonstrate MOEBA-BIO's capability to design algorithms tailored to specific biological domains, facilitating knowledge injection through the global perspective complete representation and the direct equivalence between individual and solution.

The foundation of this function is based on studies indicating that co-expressed genes tend to be regulated by the same transcription factors [317], and often, due to the typical scale-free topology of gene regulatory networks [20], they belong to the same communities (see Figure 2.2). The function measures the modularity of the partition into communities that emerge from the individual's biclusters on the gene regulatory network inferred by the GENIE3 tool [9], based on the provided input data.

The final score reflects the modularity of the solution, where a value close to 0 indicates high coherence (genes well-grouped by their common regulators), and a value close to 1 indicates low coherence. This allows for evaluating how well the co-expressed genes in a bicluster are regulated by the same factors in the inferred network, providing an accurate and biologically meaningful metric to assess the biclustering solution in gene expression data.

The implementation of this fitness function is represented in the pseudocode presented in Algorithm 18. First, each gene is assigned to a bicluster, mapping the genes within the groups formed by the biclusters (line 1 in Algorithm 18).

Next, a cumulative variable called sum is initialized to store the result of the regulatory coherence calculated between the genes within each bicluster (line 2 in Algorithm 18). Then, all pairs of genes present in the regulatory network inferred by GENIE3 are iterated over (lines 3-7 in Algorithm 18).

Within this double loop, for each pair of genes g_i and g_j, it is checked whether both belong to the same bicluster (line 5 in Algorithm 18). If they do so, the cumulative sum is updated by adding the regulatory confidence value between the

Algorithm 18 Regulatory Coherence fitness function.

Require: Set of all biclusters B, Gene regulatory network inferred by GENIE3 G
Ensure: Value of the fitness function $score$

1: $rowBics \leftarrow assignBiclusters(B, G)$
2: $sum \leftarrow 0$
3: **for** each gi in G **do**
4: **for** each gj in G **do**
5: **if** $rowBics[gi] = rowBics[gj]$ **then**
6: $sum \leftarrow sum + G.conf(gi, gj)$
7: $sum \leftarrow sum - \frac{G.outDegree(gi) \times G.inDegree(gj)}{G.totalWeight}$
8: **end if**
9: **end for**
10: **end for**
11: $modularity \leftarrow sum/G.totalWeight$
12: **return** $1 - (modularity + 1)/2$

two genes in the network (line 6 in Algorithm 18) and subtracting an adjusted term based on the out-degrees and in-degrees of both genes, normalized by the total weight of the network (line 7 in Algorithm 18).

After processing all gene pairs, the modularity value is calculated by dividing the cumulative sum by the total weight of the regulatory network (line 8 in Algorithm 18). The final score is normalized to a range between 0 and 1 using a linear transformation (line 9 in Algorithm 18), where a value close to 0 indicates high coherence, while a value close to 1 indicates low regulatory coherence. Since this fitness function does not require a summary strategy, it is directly oriented towards minimization.

11.2 Experimentation

The experimentation in this study is divided into five main phases.

The first phase of the experimentation focuses on using the self-configurator of `MOEBA-BIO` framework on a dataset specific to the biomedical domain. To this end, the simulated data generated by the FABIA package, as described in Section 4.2.1, are used.

The self-configurator has considered all possible values for both first-level and second-level parameters. For both wrapper genetic algorithms, a population size of 50 individuals has been set, with 1500 evaluations for the outer loop and

2000 for the inner loop, as the latter has a larger search space. These values are based on those tested in the reference meta-optimizer [300], which have also proven to be sufficient for the convergence of both algorithms. Additionally, the number of MOEBA-BIO evaluations has been reduced to 25,000, a quantity that demonstrated the ability to capture the most significant progress of the populations in the first phase of experimentation. Furthermore, clustering error [314] has been chosen as the supervised metric, and the hypervolume value, changed to a negative sign with unitary reference, has been chosen as the unsupervised metric.

The second phase tests the effectiveness of the self-configurator. To achieve this, the winning configuration from the previous phase is compared with other candidates from both the supervised and unsupervised phases. This process allows for verifying whether the clustering error metric has been minimized while simultaneously improving the hypervolume value. Each configuration is executed five times with 150,000 evaluations, allowing for a more robust comparison and justifying the decision to avoid repetition in the self-configuration process, thereby prioritizing computational feasibility without damaging the quality of the solutions.

The third phase, once the best configuration of MOEBA-BIO has been identified for the simulated benchmark that represents the gene co-expression context, evaluates the developed evolutionary algorithm against other proposals through a technical validation. For this comparison, widely used algorithms of various types have been selected from the literature [318] (see section 3.2.1). Specifically, CCA [101], OPSM [120], xMOTIFs [12], ISA [13], LAS [125], Bimax [124], BiBit [14], Plaid [113], and Spectral [118] have been selected. For their execution, the biclustlib Python library [123] was used, providing them with the data matrices previously generated by the FABIA package [209] in R.

It should be emphasized that the CCA, Bimax, Plaid, xMOTIFs, and LAS algorithms require identifying the number of biclusters beforehand. This gives them a comparative advantage over MOEBA-BIO, where this information is unknown and part of the algorithm's learning process. To make the comparison fairer, the exact number was not provided; instead, an approximate value was calculated as 10% of the average between the number of rows and columns.

In the dataset generated by FABIA, this means the approximate number of biclusters is slightly higher than the actual number. This allows these algorithms the possibility of identifying all biclusters from the gold standard, while also testing their ability to avoid grouping unrelated genes in expression patterns when the researcher lacks an accurate estimate of the number of biclusters, a

situation also evaluated in MOEBA-BIO and, therefore, important for validating these algorithms.

In addition to the clustering error [314], supervised individual level precision metrics such as Recovery, Relevance [315] and Ayadi's score [316] are used. These metrics do not consider redundancies and select the best-inferred bicluster for each reference bicluster in the gold standard. This approach eliminates the penalty for redundancy to demonstrate that the quality of this proposal is not due, in any case, to poor guidance from algorithms requiring knowledge of the number of biclusters beforehand. If MOEBA-BIO achieves better results in these metrics, it would highlight its ability to manage redundancy effectively when the number of biclusters is unknown. It also means that found biclusters surpass the best match achieved by the algorithms that fail to identify the correct number of biclusters.

After execution, for each state-of-the-art algorithm result, the metric value is calculated directly, while for MOEBA-BIO, the results from the previous phase are used to select the best solution and the overall median of the distribution from the 5 runs.

Additionally, to achieve a more comprehensive technical comparison, the execution times of each algorithm are measured. This aims to assess the computational performance of each proposal by analyzing both the accuracy of the obtained results and the time required to achieve them.

The fourth phase extends the validation process of this proposal to a more biological level by utilizing real-world gene expression data. Specifically, the benchmark dataset described in 4.2.2 is used, which gathers 17 time-series gene expression matrices from several sources. Since these data are unlabeled, they do not have a gold standard for conducting a technical comparison as in the previous experimental phase. Instead, a biological validation is performed based on the functional enrichment analysis of the biclusters.

For this phase, the Python library biclustlib[1], an extension of the homonymous project introduced in [123] (used in the previous phase), is employed. This version has been enriched with the functionalities of the well-known GOATOOLS library [319]. This software enables the integration of the same methodologies previously considered (CCA [101], OPSM [120], xMOTIFs [12], ISA [13], LAS [125], Bimax [124], BiBit [14], Plaid [113], and Spectral [118]), as well as the use of the mentioned dataset[2] to assess the biological relevance of the bi-

[1]Available on PyPI: https://pypi.org/project/biclustlib/

[2]Datasets available at: https://github.com/nikitasigal/biclustlib/tree/main/src/biclustlib/benchmark/data/jaskowiak

clusters obtained by each approach through functional enrichment analysis with GOATOOLS.

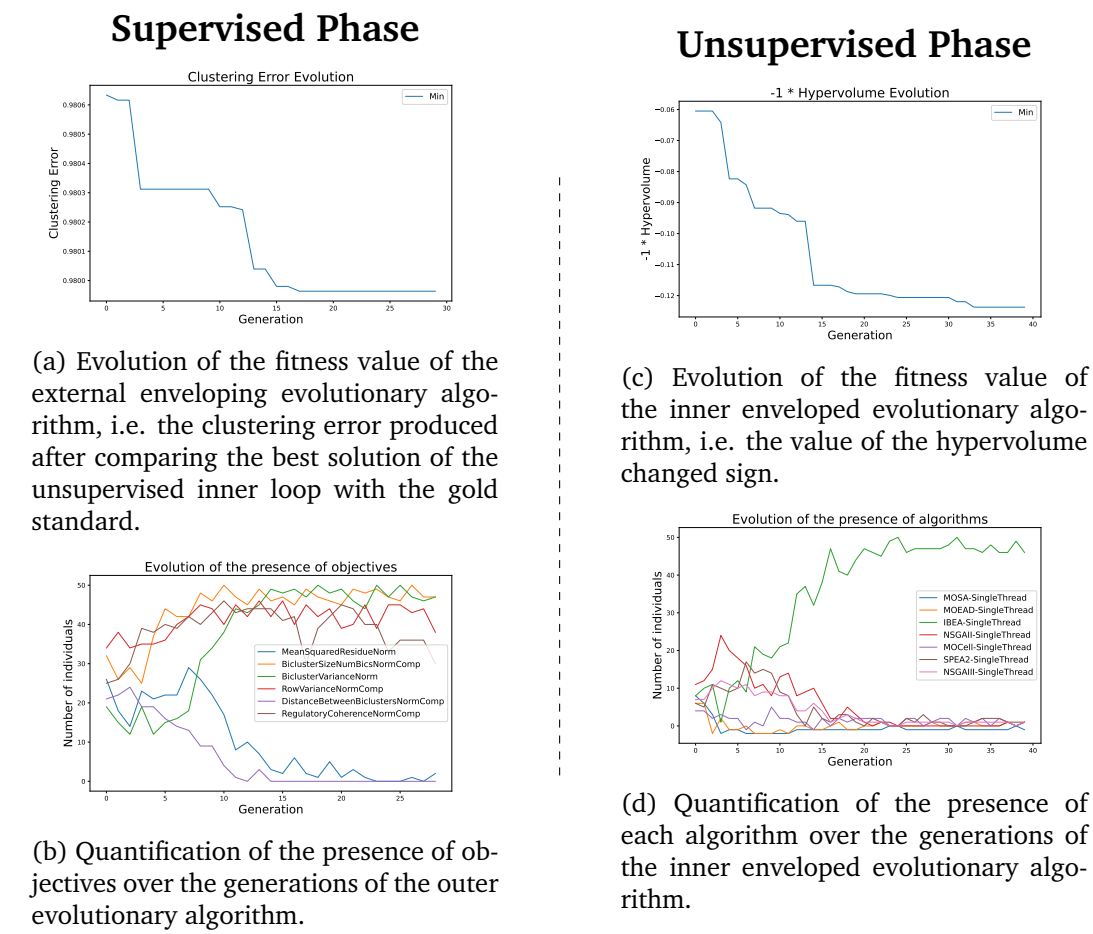
The functional enrichment analysis is based on identifying Gene Ontology (GO) [320] terms that are significantly represented among the genes grouped within each bicluster. To achieve this, an over-representation test is applied, comparing the proportion of genes annotated with a GO term within a bicluster to the expected proportion in the reference population. Specifically, a Fisher's exact test [321] is used, with significance values corrected using the Benjamini-Hochberg procedure to control the false discovery rate (FDR) [322].

For each evaluated algorithm, the proportion of enriched biclusters is measured at different significance levels ($\alpha \in \{0.05, 0.005, 0.00001\}$). A bicluster is considered enriched if at least one of its associated GO terms has a corrected p-value below α. The higher this proportion for a given methodology, the stronger the evidence that the inferred biclusters capture functionally coherent groupings of genes, reflecting biologically relevant relationships in the gene expression data.

Once the proportions of enriched biclusters have been obtained for each methodology, dataset, and significance level, a Friedman statistical ranking is conducted along with non-parametric Holm tests [313] for each α value. This analysis allows verifying whether the biological relevance of the biclusters inferred by `MOEBA-BIO` is statistically superior to that of the other methodologies.

The fifth phase involves analyzing how the size of the input matrix affects the execution time of the algorithm constructed in the previous phases. For this purpose, multiple synthetic datasets with 100 columns were generated, progressively increasing the number of rows from 100 to 5000 in steps of 100. Since this dimension does not involve overlap and tends toward full coverage by the algorithm, it enables an accurate assessment of the scalability of the approach, allowing observation of whether the growth in execution time is linear, quadratic, or follows another pattern, and identification of potential bottlenecks resulting from increased data size.

It should be emphasized that the generated data are completely random, with no internal structure or presence of biclusters. As a result, the outcomes of the algorithm in this phase have no analytical or interpretative value; the purpose of the experiment is strictly limited to the analysis of computational cost. To ensure representative results, the algorithm was executed five times on each matrix, and the median execution time was subsequently calculated.



(a) Evolution of the fitness value of the external enveloping evolutionary algorithm, i.e. the clustering error produced after comparing the best solution of the unsupervised inner loop with the gold standard.

(b) Quantification of the presence of objectives over the generations of the outer evolutionary algorithm.

(c) Evolution of the fitness value of the inner enveloped evolutionary algorithm, i.e. the value of the hypervolume changed sign.

(d) Quantification of the presence of each algorithm over the generations of the inner enveloped evolutionary algorithm.

Figure 11.2: Tracking the execution of the autoconfigurator for gene co-expression data. The evolution of the outer loop (supervised phase) is shown on the left and the inner loop (unsupervised phase) on the right.

11.3 Results and discussion

To facilitate the understanding of this document, the results of each phase of the experimentation are presented in the same order as detailed in the previous section.

11.3.1 Autoconfiguration

The first phase of experimentation in this study focuses on the execution of the self-configurator on a dataset specific to the domain of gene expression. The

run on a machine with 64 cores and 700GB of RAM took approximately 6 days and 14 hours. Although it takes a considerable amount of time, it is considered a valuable investment in exchange for identifying the subset of objectives and parameter values that best explain the biological domain of application.

Figure 11.2 provides information about the evolution of the external population of the self-configurator and the internal population of the winning supervised configuration. First, regarding the supervised phase, Figure 11.2a shows the evolution of the clustering error metric as different sets of objectives and their subparameters are evaluated. The convergence of this curve indicates that the outer wrapper algorithm has reached a sufficient number of evaluations. Additionally, for this same phase, Figure 11.2b shows the evolution of the objectives as a global count of their presence in the individuals of each generation. Two observations can be extracted from this graph: the convergence of the clustering error metric aligns perfectly with the clarification of the self-determined objectives, and the combined use of the Adaptive bSIZE, bVAR, rVAR, and Regulatory Coherence functions manage to maximize the precision of the solutions.

The appearance of the rVAR objective and the exclusion of bDIFF makes sense given the type of pattern typically observed in co-expression data. In the context of gene co-expression, it is common for co-expressed genes to maintain a constant relationship across experimental conditions, but with different levels of activation. This implies that the bicluster's rows may be at different levels, but maintain a consistent proportion across the columns of the bicluster. This characteristic makes the maximization of rVAR particularly suitable for this domain, while bDIFF, which favors rows with similar cell values, does not fit as well in this case.

Secondly, on the right side of Figure 11.2, information about the unsupervised phase is shown, specifically the evolution of individuals in the internal population of the self-configurator corresponding to the winning individual from the supervised phase. In Figure 11.2c, the evolution of the signed change in the normalized hypervolume value with unitary reference is plotted, as different algorithms, parameters, and unsupervised subparameters are evaluated. Once again, the convergence of this curve ensures a sufficient number of evaluations in the internal wrapper algorithm of the self-configurator. Finally, from all the self-configured parameters, Figure 11.2d depicts the evolution of the different candidate algorithms within the population. Of course, although two combinations may choose the same algorithm, they can have completely different subparameter values, influencing their performance. In other words, as with Figure 11.2b for the supervised phase, Figure 11.2d provides a simplified explanation of what occurred during the unsupervised phase. The analysis of Figure 11.2d

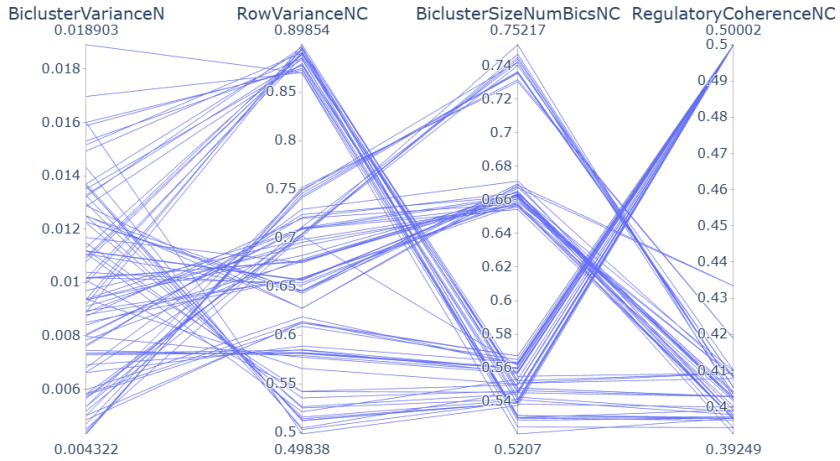


Figure 11.3: Parallel coordinate plot for the front obtained by MOEBA-BIO using complete representation and the best configuration found by the autoconfigurator for the first instance of the gene expression data simulator.

shows that the self-configurator converges decisively on the IBEA algorithm.

Finally, the winning configuration consists of the following fitness functions: BiclusterVarianceNorm (objectives summarized using the mean), RowVarianceNormComp (also summarized with the mean), BiclusterSizeNumBicsNormComp (coherence weight: 0.0802, objective summary: mean, row weight: 0.8746), and RegulatoryCoherenceNormComp. As for unsupervised parameters, a crossover probability of 63.79%, a mutation probability of 0.0287%, a population size of 100, and the IBEA-SingleThread algorithm (which has no subparameters) were determined. Regarding the operators, the only ones implemented so far were selected. For the crossover operator, PartiallyMappedCrossover, BicUniformCrossover, and CellUniformCrossover were used, while for the mutation operator, SwapMutation, BicUniformMutation and CellUniformMutation were selected.

In addition to the evolution of the internal and external wrapper populations, the self-configurator provides the results from the runs obtained by evaluating the winning solution vectors. That is, the execution for each instance of the winning configuration. This allows to represent in Figure 11.3 the parallel coordinates of the approximate Pareto front obtained by the self-configured MOEBA-BIO in the first data matrix generated by the FABIA R package. The aim is to highlight the trade-offs between the different self-determined objectives and discard any possible redundancy during the optimization process.

Table 11.1: Median of each front's medians and maximum of each front's maxima for the clustering error complementary metric after running with 5 replicates of the winning configuration and three other candidates on the gene expression dataset generated by FABIA.

Configuration	Instance 1		Instance 2		Instance 3		Instance 4	
	Median	Max	Median	Max	Median	Max	Median	Max
Winner	**0.0136**	**0.0268**	0.0164	**0.0262**	0.0094	0.0174	**0.0136**	**0.0220**
Winner - No Reg. Coherence	0.0099	0.0238	0.0146	0.0235	0.0089	**0.0200**	0.0097	0.0176
Winner - No Row Variance	0.0134	0.0237	**0.0192**	0.0257	0.0106	0.0194	0.0130	0.0204
Winner - Dist Between Bics	0.0130	0.0254	0.0182	0.0241	**0.0118**	0.0178	0.0135	0.0218

Table 11.2: Hypervolume values for different candidates of the unsupervised phase and the winning configuration. In particular, the median of 5 independent runs on the FABIA simulated dataset is presented.

Configuration	Parameters				Hypervolume			
	Algorithm	Population Size	Crossover Probability	Mutation Probability	Instance 1	Instance 2	Instance 3	Instance 4
winner	IBEA	100	0.64	2.87e-04	**0.1058**	**0.0863**	**0.1378**	**0.0845**
candidate-1	NSGAIII	100	0.64	2.87e-04	0.0924	0.0564	0.0826	0.0809
candidate-2	NSGAII	500	0.90	0.10	0.0676	0.0447	0.0649	0.0573
candidate-3	SPEA2	300	0.75	0.05	0.0699	0.0503	0.0705	0.0700
candidate-4	MOSA	400	0.60	0.25	0.0483	0.0394	0.0466	0.0457

11.3.2 Candidates comparison: autoconfigurator validation

In **the second phase** of experimentation, the effectiveness of the self-configurator was evaluated by comparing the winning configuration with other additional candidate configurations.

First, the winning configuration is compared with other candidate configurations from the supervised phase by measuring the clustering error complementary metric for each of them. These configurations exclude various key objective functions and introduce others initially discarded by the self-configurator. The results, presented in Table 11.1, show the median of the medians and the maximum of the maximums from each front for the complementary clustering error metric across the four instances of the gene expression dataset.

Overall, the winning configuration outperforms the candidates in most instances, achieving the best medians in instances 1 and 4, and the best maximum values in instances 1, 2, and 4. This indicates that including all self-determined objective functions (including regulatory coherence and row variance) is crucial for ensuring high performance in minimizing the complementary clustering error. Meanwhile, introducing other objectives discarded by the self-configurator also seems to reflect a decline in result accuracy.

Second, the winning configuration is compared with other candidates from the unsupervised phase. To do this, the hypervolume value is measured with

respect to the reference front, constructed as the set of non-dominated solutions from all executions, which differs from the reference point used during the self-configurator.

Among the candidates, one configuration results from modifying the winning one solely by replacing the IBEA algorithm with the extended NSGA-III. Another configuration features a larger population size and higher operator probabilities using NSGA-II. Additionally, an intermediate configuration is evaluated with the SPEA2 algorithm, and a final configuration is tested with a crossover probability similar to the winning one but with a higher mutation rate using MOSA.

Table 11.2 presents the median of these values for each candidate, clearly demonstrating the dominance of the winning configuration over the others.

Finally, the analysis of these results using 5 independent runs for each instance further justifies the decision to discard such repetitions in the self-configuration process, prioritizing computational feasibility without compromising the quality of the self-configuration.

11.3.3 Algorithmic comparison

The third phase of experimentation is oriented to compare the results of the self-configured `MOEBA-BIO` from the previous phase with various state-of-the-art techniques. Figure 11.4 shows the results obtained by the different techniques for the supervised metrics: Clustering Error [314], Recovery [315], Relevance [315], and Ayadi's Score [316]. As discussed in the experimentation section, clustering error is the most suitable and comprehensive metric to validate this proposal's contributions. However, two additional individual metrics have been implemented to demonstrate that the performance in terms of clustering error is not due to the specification of an inappropriate number of biclusters in specific state-of-the-art techniques that require this parameter.

As shown in Figure 11.4, both the best solution of `MOEBA-BIO` and the median of the medians of the five fronts generated for each instance are located at the top in all metrics, particularly excelling in Clustering Error, the primary metric of this study.

A key point is that the median of the medians, despite outperforming most state-of-the-art techniques, represents a random selection within the approximate Pareto front generated by the algorithm. This should not occur in an ideal context, as selecting the most appropriate solution within the front is a task for the domain expert, who, with a better understanding of the search space, can make more informed decisions than a random choice.

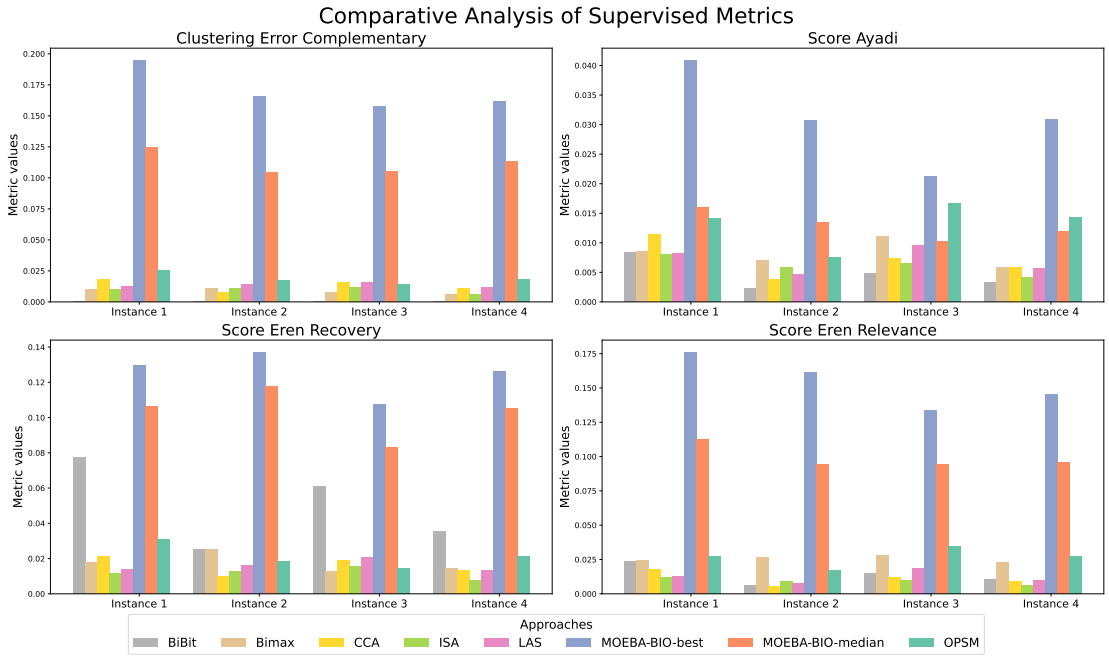


Figure 11.4: Comparison of the self-generated MOEBA-BIO algorithm for the gene co-expression domain concerning different recognised state-of-the-art techniques. For each instance, the values of the supervised metrics: Clustering Error, ScoreAyadi and ScoreEren (recovery and relevance) are compared.

According to a Friedman statistical rank, the best solution of MOEBA-BIO leads the ranking in all metrics, followed in all cases by its median. It is worth mentioning that in the ScoreAyadi metric, the median of MOEBA-BIO shares second place with a value of 2.75, alongside the OPSM technique, which does not show outstanding performance in any of the other metrics. OPSM is well-suited to the ScoreAyadi metric, which is why, in some instances, it achieves better values than the median of MOEBA-BIO, thus affirming the validity of the No Free Lunch theorem in this context.

Finally, Figure 11.5 presents the execution times required for each instance using each methodology on the same machine with eight cores and 64 GB of RAM. In the case of MOEBA-BIO, an additional, particularly costly internal step has been timed separately to assess its impact on the total execution time and provide a more detailed performance comparison of the algorithm. This step corresponds to the initial inference of the GRN required for the *regulatory coherence* objective, performed by GENIE3 [9] (gray fragment).

Although the execution times for MOEBA-BIO are higher than those of other

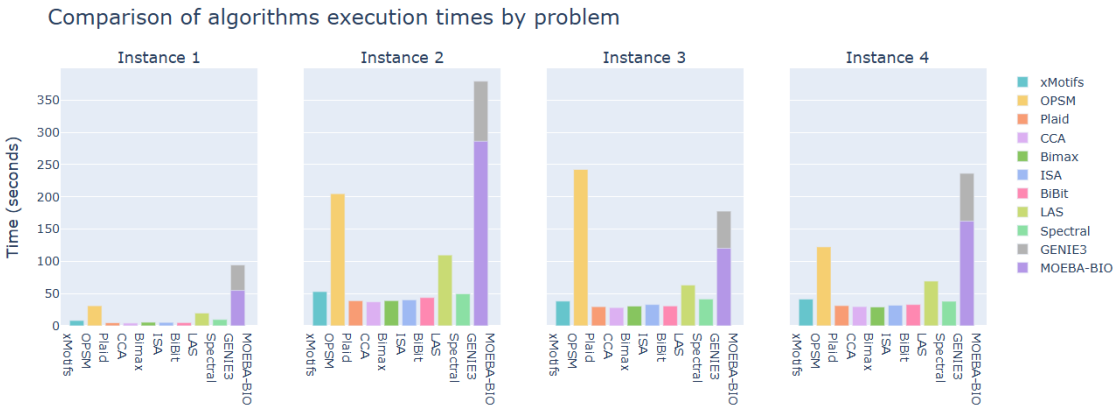


Figure 11.5: Comparison of the time required by each methodology for each instance of the synthetic dataset generated by the FABIA R package. The time required by `MOEBA-BIO` is divided into two fragments, detailing the time required for the initial GRN inference performed by GENIE3 for the regulatory coherence objective.

methodologies, three aspects should be considered: (1) a significant portion of the execution time is due to the GRN inference using GENIE3, a component that users can replace with a lighter technique if preferred; (2) previous results demonstrate that our algorithm achieves higher accuracy in identifying relevant biclusters; and (3) unlike the other compared methodologies, `MOEBA-BIO` incorporates the self-determination of the number of biclusters, which introduces additional computational complexity.

11.3.4 Functional enrichment comparison

The fourth phase compares the results obtained by different biclustering methodologies on real gene expression datasets by measuring the proportion of significantly enriched biclusters under various significance thresholds. This proportion ranges from 0 to 1, reaching its maximum value when all inferred biclusters exceed the significance threshold and its minimum when none do.

Figure 11.6 presents these proportions for α values of 0.05, 0.005, and 0.00001. As observed, both the best solution of `MOEBA-BIO` and the median of the front achieve relatively high proportions, surpassing a significant number of state-of-the-art methodologies.

However, to provide greater statistical rigor to this comparison, Tables 11.3, 11.4, and 11.5 present the results of applying the Friedman statistical ranking with non-parametric Holm tests for α values of 0.05, 0.005, and 0.00001, respectively. These tables show that, although its dominance does not reach absolute

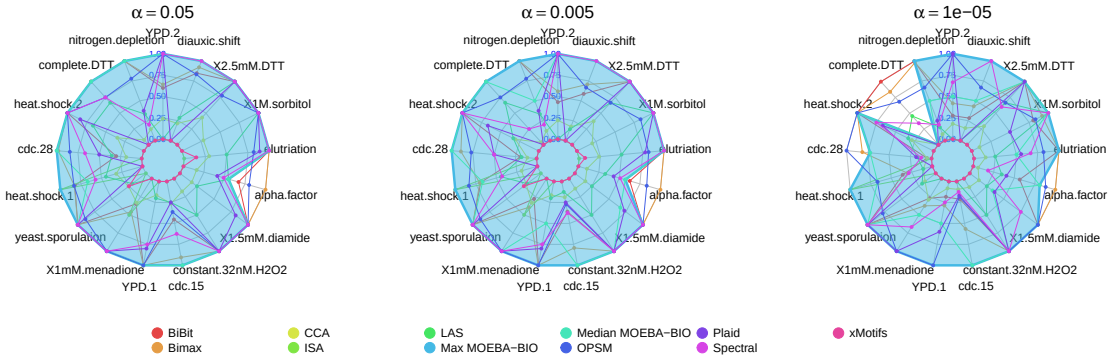


Figure 11.6: Comparison of the proportion of enriched biclusters obtained by each methodology on the real-world gene expression dataset, evaluated under different significance thresholds ($\alpha \in \{0.05, 0.005, 0.00001\}$). Higher proportions indicate a stronger functional coherence among the genes within the detected biclusters. The values obtained for the best solution of the `MOEBA-BIO` Pareto front plot a colored area that allows visualizing the dominance of the proposal.

Table 11.3: Friedman mean rank with Holm's adjusted p values (0.05) for the ratio of enriched biclusters with $alpha = 0.05$.

Enriched biclusters ratio ($alpha = 0.05$)		
Technique	$Friedman's\{Rank\}$	$Holm's\{Adj - p\}$
Max MOEBA-BIO	**3.03**	-
Median MOEBA-BIO	3.29	1.17
Bimax	3.65	1.17
BiBit	4.41	0.85
Spectral	4.56	0.85
OPSM	4.59	0.85
Plaid	6.29	2.46e-02
LAS	7.56	4.79e-04
CCA	8.65	6.31e-06
ISA	9.68	4.61e-08
xMotifs	10.29	1.70e-09

statistical significance overall proposals, the best solution of `MOEBA-BIO` consistently ranks first across all α thresholds. Additionally, the median of the front achieves promising positions, frequently appearing on the ranking podium.

Finally, highlight that functional enrichment analysis, widely used to validate the quality of biclustering algorithms on unlabeled expression data, does not consider the bi-dimensionality of the results. That is, while this approach validates the coexistence of certain genes within biclusters, it does not consider the

Table 11.4: Friedman mean rank with Holm's adjusted p values (0.05) for the ratio of enriched biclusters with $alpha = 0.005$.

Enriched biclusters ratio ($alpha = 0.005$)		
Technique	$Friedman's\{Rank\}$	$Holm's\{Adj - p\}$
Max MOEBA-BIO	**2.82**	-
Bimax	3.47	0.79
Median MOEBA-BIO	3.79	0.79
BiBit	4.41	0.67
OPSM	4.47	0.67
Spectral	4.53	0.67
Plaid	6.00	3.14e-02
LAS	7.94	4.79e-05
CCA	8.65	2.46e-06
ISA	9.68	1.53e-08
xMotifs	10.24	7.25e-10

Table 11.5: Friedman mean rank with Holm's adjusted p values (0.05) for the ratio of enriched biclusters with $alpha = 0.00001$.

Enriched biclusters ratio ($alpha = 0.00001$)		
Technique	$Friedman's\{Rank\}$	$Holm's\{Adj - p\}$
Max MOEBA-BIO	**2.76**	-
OPSM	3.53	0.88
Bimax	3.65	0.88
Median MOEBA-BIO	4.32	0.51
BiBit	4.59	0.44
Spectral	5.32	0.12
Plaid	5.65	6.77e-02
LAS	7.74	8.72e-05
CCA	8.56	2.81e-06
ISA	9.68	1.11e-08
xMotifs	10.21	6.10e-10

conditions under which they have been grouped. Therefore, the previous experimental phase with simulated data is essential, as it provides a more precise validation through metrics that consider both bicluster dimensions and evaluate `MOEBA-BIO`'s ability to self-determine the number of biclusters.

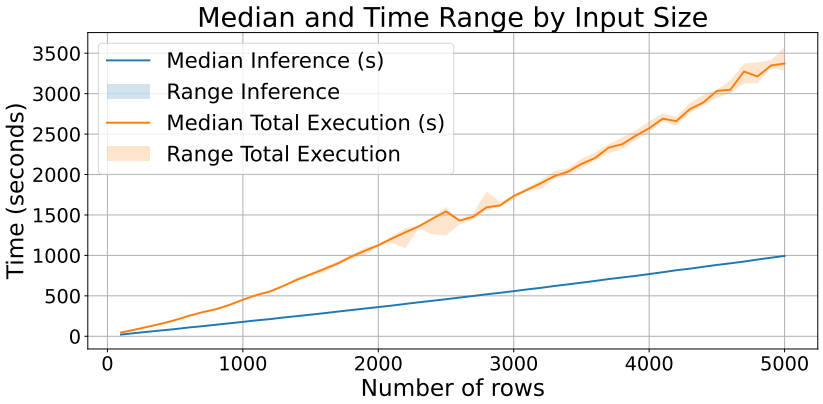


Figure 11.7: Median and range of execution times (in seconds) according to input matrix size (number of rows). The plot differentiates between the total execution time of the biclustering process and the inference time required by GENIE3 to instantiate the *Regulatory coherence* objective.

11.3.5 Scalability study

The fifth phase of the experimentation involved a total of 250 executions of the self-constructed algorithm for the gene co-expression domain. The analysis of execution times enabled the generation of the plot shown in Figure 11.7, which illustrates the evolution of total execution time and the time required by GENIE3 at the beginning of the *Regulatory Coherence* objective instantiation, as a function of the number of input rows.

The results show that both the total execution time and the time specifically required by the inference technique exhibit exponential growth as the matrix size increases. As observed, the time required by GENIE3 represents a significant portion of the total execution time. Consequently, the inference technique can be considered a relevant factor in terms of scalability.

This behavior suggests that, in scenarios where execution time is a priority, a viable alternative would be to replace the current inference technique with a more efficient one or one better suited to the scale of the available data. This is feasible thanks to the modularity of the constructed framework, which not only allows the configuration of algorithms tailored to the application domain, but also provides the flexibility to adapt the process according to computational constraints and the size of the input data.

Regarding the algorithm's asymptotic time complexity (Big-O), an implementation-informed approximation is provided below:

- *Preprocessing (once):* infer the regulatory network from the gene-expression matrix with GENIE3 (C_{base}).

- *Evolutionary optimization (over E evaluations):* in each evaluation, objectives decompose into:
  - *Global (whole individual):* **Regulatory coherence** on the precomputed network: nested scan of node pairs $\Rightarrow O(n^2)$.
  - *Per bicluster $b = 1, \ldots, B$:*
    - **Row variance**: two passes over entries $\Rightarrow O(r_b\, c_b)$.
    - **Bicluster variance**: mean + squared deviations $\Rightarrow O(r_b\, c_b)$.
    - **Adaptive bicluster size**: size-based term $\Rightarrow O(1)$ per bicluster.

  Summing the per-evaluation costs:

$$O(n^2) \; + \; \sum_{b=1}^{B} \Big(O(r_b\, c_b) + O(r_b\, c_b) + O(1) \Big).$$

- *Overall:*

$$\boxed{C_{\text{base}} \; + \; C_{\text{MOEBA-BIO-CoExp}}} \; \approx \; C_{\text{base}} \; + \; O\left(E \cdot \left(n^2 + \sum_{b=1}^{B} r_b\, c_b \right) \right).$$

- *Symbols:* n = number of genes (rows of the expression matrix); s = number of samples (columns); B = number of biclusters in an individual; r_b, c_b = rows/columns of bicluster b; P = population size; T = number of generations; $E \approx T \cdot P$; C_{base} = one-off cost of GENIE3 (or the chosen inference pipeline).

Part III
Final observations

Chapter 12

Conclusions

This chapter presents the final conclusions of the thesis, structured around the two main lines of work developed: the consensus inference of gene regulatory networks and biclustering in biomedical contexts. Both have contributed, from complementary approaches, to validating the central hypothesis of this thesis.

12.1 Gene regulatory networks consensus inference

Throughout this thesis, the problem of consensus inference of gene regulatory networks has been progressively addressed, through the development of a series of proposals that incrementally overcome the limitations of traditional approaches.

GENECI, the initial proposal presented in Chapter 5, laid the foundation for an evolutionary framework to combine multiple inference techniques through an optimized weighted voting system. Unlike previous strategies that required labeled data or direct supervision using gold standards, `GENECI` avoids any dependence on prior knowledge of the underlying network, evaluating the consensus networks based on internal properties such as the quality of inferred links and the emerging topology.

Its evaluation on standard academic benchmarks (DREAM3, DREAM4, DREAM5, and IRMA) demonstrated a remarkable generalization capability, delivering highly competitive results in all cases and frequently outperforming individual techniques in both AUROC and AUPR. Its performance was particularly notable in medium- and large-scale networks, where the consensus approach succeeded in leveraging the strengths of each technique without inheriting their weaknesses.

Furthermore, its applicability to clinical data was demonstrated through the analysis of gene expression in melanoma patients, where the prioritized interactions showed strong bibliographic support.

Stability analyses revealed a high consistency in the solutions obtained across independent runs, reinforcing the reliability of the approach despite its stochastic nature. Additionally, the analysis of the weights assigned by the algorithm to each technique revealed an intelligent adaptation to the context of the problem: consistent patterns were observed in assigning greater weight to techniques whose predictions were supported by others, while those with relatively poor performance were systematically penalized, without introducing noise into the final consensus. This ability to self-regulate without external supervision positions `GENECI` as a solid starting point for the further development of more complex approaches.

Building on this foundation, **Memetic Inference**, presented in Chapter 6, introduced for the first time the active integration of expert knowledge into the inference of regulatory networks through a guided local search phase. This memetic extension enabled the selective incorporation of known gene interactions during the evolutionary process, steering the optimization towards more accurate solutions. Despite using only 5% of the gold standard as reference, the results showed significant and statistically validated improvements over its predecessor, even in extensively studied networks from the DREAM benchmark.

Sensitivity analysis revealed that configurations with higher local search frequency and full inclusion of known interactions achieved the best performance. However, it also highlighted the need for a balanced use of this information to avoid biases or overfitting. In this regard, the proposal demonstrated remarkable robustness against poorly distributed knowledge samples, minimizing the impact of unfavorable cases without compromising convergence.

The benefits were particularly evident in larger networks, where the local search phase was able to reinforce valid patterns and correct deviations in the consensus when individual techniques produced divergent predictions. This ability to leverage partial knowledge and redirect the evolution of solutions without relying on direct supervision positions `Memetic Inference` as a versatile and adaptable tool with high potential for application in real clinical scenarios, where expert knowledge is often limited but highly valuable.

Subsequently, **MO-GENECI**, presented in Chapter 7, took a decisive step forward by overcoming one of the main limitations of its predecessors: the need to combine multiple criteria into a single aggregated function. To address this, it adopted a multi-objective evolutionary approach, enabling the simultaneous

optimization of several key properties of gene regulatory networks, such as technique coherence, topological structure, and the presence of regulatory motifs. Each of these objectives underwent an independent design and refinement process: several functional versions were proposed for each, and the most suitable was selected through rigorous statistical comparisons (Friedman and Wilcoxon tests) on a large set of simulated networks, assessing their ability to approximate reference networks.

The algorithm, based on the NSGA-II framework, incorporated evolutionary operators tailored to the problem, such as a simplex crossover that preserves feasibility and a guided mutation with intensity control. Its parameters were tuned by identifying the optimal configuration through multiple independent runs and six quality metrics of the Pareto front. The effectiveness of the proposal was evaluated on an extensive benchmark of 106 networks, ranging in size from 10 to over 2,000 genes, demonstrating statistically significant superiority over 26 individual inference techniques across all considered size ranges.

Furthermore, to validate its clinical applicability, the same real gene expression dataset from melanoma patients used in `GENECI` was reused. Thanks to its methodological enhancements and multi-objective integration, `MO-GENECI` was not only able to rediscover the most relevant interactions previously identified, but also uncovered new high-confidence interactions that were subsequently supported by biomedical publications.

PBEvoGen, presented in Chapter 8, introduced a new perspective in the consensus inference of gene regulatory networks by incorporating, for the first time in this context, a preference-based selection mechanism. This mechanism allows domain experts to guide the algorithm's evolution toward regions of the objective space with high biological interest through the definition of reference points. This proposal grants the expert an active role in the evolutionary search without requiring changes to the general architecture of the algorithm or the objectives being optimized.

First, the proposal demonstrated that the use of well-positioned preference points significantly improves the accuracy of the inferred networks, even outperforming `MO-GENECI`, which had already shown superiority over 26 state-of-the-art techniques. Second, spatial analyses confirmed that the solutions generated by `PBEvoGen` not only concentrate around the reference point but also inherit its biological relevance, providing evidence that experts can anticipate areas of interest based on topological properties or prior literature. Finally, in large-scale networks, it was observed that the new mechanism enables optimization levels equivalent to those of `MO-GENECI` with only half the number of evaluations,

substantially reducing computational cost without compromising result quality.

Finally, **BIO-INSIGHT**, detailed in Chapter 9, represents the most recent and comprehensive proposal of this thesis in the field of consensus inference of gene regulatory networks. Its development has addressed a broad and ambitious set of research questions, consolidating an evolutionary strategy that surpasses previous approaches in accuracy thanks to its extensive analytical framework, which incorporates a richer and more biologically coherent perspective.

First, `BIO-INSIGHT` has shown that networks inferred by different high-precision techniques exhibit substantial discrepancies, reinforcing the need for consensus strategies. Through a novel and extensive objective space composed of biologically grounded trade-offs, it has been demonstrated that it is possible to guide inference toward more robust and interpretable solutions, whose structure and topology align with known patterns of biological networks.

At the algorithmic level, it has been confirmed that the proposed strategy does not simply replicate the learning already performed by individual techniques, but rather provides complementary knowledge, thereby reinforcing the holistic and innovative nature of the approach. Moreover, the ablation study confirms that the joint optimization of the proposed objectives yields networks of higher quality than those obtained by optimizing each aspect independently, thus justifying the multifaceted design of the model.

Finally, the applicability of `BIO-INSIGHT` has been validated in a real clinical context through its application to gene expression data from patients with fibromyalgia, myalgic encephalomyelitis, and co-diagnosis. In this setting, the tool was able to detect pathology-specific differential interactions, some of which are supported by scientific literature or experimental evidence. These findings not only reinforce the robustness of the proposal but also demonstrate its utility as a support tool for understanding complex diseases without validated biomarkers, paving the way for future biomedical applications with clinical impact.

`BIO-INSIGHT` thus represents the culmination of this line of research: a mature, flexible, and clinically interpretable proposal that integrates previous advances and establishes new foundations for consensus inference of regulatory networks in real-world and complex scenarios.

12.2 Biclustering for biomedical domains

In addition to the main line of this thesis focused on the consensus inference of gene regulatory networks, a complementary research direction has been ex-

plored, oriented toward biclustering in biomedical contexts with the aim of extrapolating and reusing part of the previously developed knowledge. In this context, **MOEBA-BIO** was presented in Chapter 10, a flexible evolutionary framework that introduces a global representation in which each individual encodes a complete solution to the problem, enabling the joint evaluation of all proposed biclusters and thus overcoming the typical fragmentation of traditional representations. This representation has enabled the introduction of novel evaluation objectives in the literature, capable of leveraging relationships between biclusters and assessing global properties of the solution, such as differentiation and structural coherence. Among its main contributions is also the ability to self-determine the number of biclusters, eliminating the need to set this parameter externally or apply subjective postprocessing procedures.

In an initial validation phase, `MOEBA-BIO` outperformed traditional evolutionary architectures with partial encodings in generic scenarios through the use of general-purpose objective functions. Both the adaptive size function and the bicluster differentiation function showed significant improvements in solution quality. Furthermore, statistical analyses confirmed that the combination of these new objectives yields solutions that are more coherent and better aligned with the latent structure of the data than when used individually.

Beyond this general context, the self-configurator proposed in `MOEBA-BIO` has also demonstrated its adaptability to specific problems in the biomedical field, such as the detection of co-expressed gene groups. For this purpose, **MOEBA-BIO-CoExp** was presented in Chapter 11, a self-constructed specialization of the framework configured through an evolutionary process guided by supervised metrics. This specialized version integrates a new global objective based on regulatory coherence, which measures the modularity of the biclusters over a gene regulatory network inferred from the expression data itself. Thanks to this integration, the algorithm has demonstrated its ability to generate biclusters with high functional coherence, aligned with the regulators shared by the genes they comprise.

The advantages of this approach have been validated on both simulated and real data, enabling not only an improvement in the technical quality of the results but also a richer biological interpretation. The framework's ability to self-configure, determine the number of biclusters, and specialize according to the domain opens up new possibilities for the application of evolutionary biclustering to complex problems in the biomedical field, overcoming the limitations of previous approaches and providing more realistic, explainable, and context-aware solutions.

Chapter 13

Future works

Based on the results obtained and the methodologies developed throughout this thesis, multiple future research directions emerge aimed at consolidating and expanding their applicability. In particular, clear opportunities for advancement have been identified both in the consensus inference of gene regulatory networks and in biclustering adapted to biomedical domains, each with specific challenges and possibilities that are outlined below.

13.1 Gene regulatory network consensus inference

The research conducted throughout this thesis on the consensus inference of gene regulatory networks has laid the groundwork for a range of future developments. The proposed methodologies can be expanded and refined to further enhance their flexibility, adaptability to diverse data types, and usefulness in real-world biological contexts. The following research lines are proposed as potential extensions:

- **Design of new codifications for individuals:** Explore richer and more expressive representations that allow for selective weighting, conditional exclusions, or partial groupings of interactions, enabling a more flexible and contextual integration of the knowledge provided by individual inference techniques.
- **Preprocessing based on structural clustering of networks:** Introduce a hybrid optimization strategy that alternates between global and local search phases, leveraging network clustering to identify conflict regions and direct computational resources toward resolving inconsistencies among

techniques.

- **Supervised autoconfigurator guided by AUROC and AUPR:** Develop a parameter self-tuning mechanism driven by gold-standard metrics, capable of discarding irrelevant objectives or learning reference points depending on the structural properties of the network to be inferred.
- **Dynamic extension of `PBEvoGen`:** Implement interactive preference-based selection mechanisms that allow experts to modify their guidance during execution, enabling progressive feedback without restarting the optimization process.
- **Machine learning assistance for Pareto front selection:** Train predictive models using simulated networks to identify which regions of the Pareto front are most likely to yield high-accuracy solutions based on AUROC and AUPR. These models could then assist researchers in selecting promising solutions for real-world datasets without gold standards.
- **Explainable AI:** Integrate XAI mechanisms that enable each inferred regulatory link to be traced back to the individual inference methods that supported it. This would provide experts with transparent attributions, facilitating the biological validation of results and increasing confidence in the consensus network. Beyond improving interpretability, such mechanisms could also help to identify biases or recurring limitations in specific inference techniques, offering valuable insights for refining both the consensus strategy and the individual methods involved.

13.2 Biclustering for biomedical domains

The line of research on biclustering in this thesis has introduced a flexible and extensible framework that enables the construction of algorithms adapted to different biological contexts. However, numerous opportunities remain to expand the applicability and generality of the proposal. The following future directions are envisioned:

- **Application of the autoconfigurator to new domains:** Leverage the `MOEBA-BIO` self-configurator to design biclustering algorithms tailored to specific biomedical problems, such as epigenomic profiling, drug response prediction, or metabolomic pattern detection.
- **Integration of objectives for heterogeneous data:** Develop new fitness functions capable of handling mixed-type data matrices, including numerical, categorical, and ordinal attributes, paving the way for biclustering on

real clinical cohorts with multimodal patient information.

- **Alternative encodings with dual-dimension overlap:** Explore representations that allow overlapping in both rows and columns, enhancing the modeling of biologically complex patterns such as partial co-expression or hierarchical regulation.

List of Tables

3.1 Assessment of consensus methods based on five key questions . . 68

3.2 Comparison between the proposals with non-traditional encodings. 77

4.1 Overview of the academic benchmark assembled for the experimental framework of this Thesis. It includes over one hundred problem instances designed to maximize diversity and support robust conclusions. Each regulatory network has been subjected to all applicable types of perturbations, with each resulting dataset constituting a distinct instance. Legend: KO (Knock-Out), KD (Knock-Down), OE (Over-Expression). 81

4.2 Characteristics of the data generated by G-bic. 90

4.3 Characteristics of the matrices generated by the FABIA simulator. 91

4.4 Summary of the real-world gene expression data sets used in the experiments, adapted from [210]. 92

5.1 Example of input to the evaluation process. This table presents a gene network of 4 interactions (column 1) that has been inferred using three different individual techniques (columns 2, 3 and 4), the results of which are intended to be consensual. In this case, an individual is evaluated, proposing the following vector of weights: (0.5, 0.3, 0.2). The first significant value to be calculated is the consensus confidence (column 6), which consists of a simple weighted sum where the weight of each technique is multiplied by the level of individual confidence reported by the technique for the interaction in question. Second, a vector (column 7) is constructed, storing in each position the mean between the weight of the technique and the distance normalized to the median of the confidence levels of all techniques (column 5). This approach allows the calculation of the second significant value, the distance. This value consists of the difference between the maximum and the minimum of the vector constructed above, and the fitness function will try to minimize it. 99

5.2 GENECI input parameters. 107

5.3 Accuracy values for DREAM3 and size 10 networks. AUPR and AUROC values are provided for each technique and problem, highlighting in bold the results obtained by GENECI and the best obtained by any of the individual techniques. 110

5.4 Accuracy values for DREAM3 and size 50 networks. In this table, a gene network is contemplated for each pair of columns, where in each row the AUPR and AUROC values are provided for each inference technique. 112

5.5 Accuracy values for DREAM3 and size 100 networks. The AUPR and AUROC values are presented in two clearly distinguishable bands. The first band shows the precision values for the individual inference techniques, while the second band shows the values obtained by GENECI after the consensus of the techniques, distinguishing between the median of the runs and the best result obtained from them. 114

5.6 Accuracy values for DREAM4 and size 10 networks. For each gene regulatory network included in this dataset (columns), the AUPR and AUROC values are shown after comparing the networks inferred by the different techniques (rows) with the respective gold standards. 116

5.7 Accuracy values for DREAM4 and size 100 networks. This table shows the results of the evaluation scripts run on each of the individual (first band) and consensus (second band) results for each problem network (columns). The best value of the individual techniques and the values of the consensus networks per GENECI (best and median of all runs) are shown in bold. 118
5.8 Accuracy values for DREAM5 networks. AUPR and AUROC values are provided for each technique and problem, highlighting in bold the results obtained by GENECI and the best obtained by any of the individual techniques. 120
5.9 Accuracy values for IRMA networks. In this table, a gene network is contemplated for each pair of columns, where in each row the AUPR and AUROC values are provided for each inference technique.120
5.10 Friedman mean rank with Holm's adjusted p values (0.05) for AUROC. 123
5.11 Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.124

6.1 Friedman mean rank with Holm's adjusted p values (0.05) for AUPR. Several distance (D) and local search probability (P) configurations are compared based on the AUPR metric. For this purpose, 15 independent runs of each configuration were performed and the median of them (Median) was rescued. After running Friedman's statistical ranking (second column), the winner (highlighted in bold with $*$) is taken as a reference to measure statistical significance against the rest using Holm's nonparametric tests (third column). 133
6.2 Friedman mean rank with Holm's adjusted p values (0.05) for AUROC. The procedure and nomenclature are identical to those in Table 6.1. 133
6.3 Accuracy values for IRMA networks. In this table, a gene network is contemplated for each pair of columns, where in each row the AUPR and AUROC values are provided for each algorithm. 137

7.1 Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.150
7.2 Friedman mean rank with Holm's adjusted p values (0.05) for AUROC. 150
7.3 Friedman mean rank with Wilcoxon p values (0.05) for AUPR and AUROC. 152
7.4 Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.156
7.5 Friedman mean rank with Holm's adjusted p values (0.05) for AUROC. 157

7.6 The first 5 configurations from the statistical Friedman ranking for each metric, indicating significance compared to the winner based on non-parametric Holm tests (R = Rejected and A = Accepted). The configurations are presented using the following abbreviations: PS = Population Size, CP = Crossover Probability, NP = Number of Parents, MP = Mutation Probability y MS = Mutation Strength. 158
7.7 AUROC and AUPR values for networks of more than 2,000 genes 171
7.8 Friedman mean rank with Holm's adjusted p values (0.05) for the AUPR metric measured across all techniques in networks with 0 to 25 genes. 172
7.9 Friedman mean rank with Holm's adjusted p values (0.05) for the AUROC metric measured across all techniques in networks with 0 to 25 genes. 173
7.10 Friedman mean rank with Holm's adjusted p values (0.05) for the AUPR metric measured across all techniques in networks with 25 to 110 genes. 174
7.11 Friedman mean rank with Holm's adjusted p values (0.05) for the AUROC metric measured across all techniques in networks with 25 to 110 genes. 175
7.12 Friedman mean rank with Holm's adjusted p values (0.05) for the AUPR metric measured across all techniques in networks with 110 to 250 genes. 176
7.13 Friedman mean rank with Holm's adjusted p values (0.05) for the AUROC metric measured across all techniques in networks with 110 to 250 genes. 177
7.14 Friedman mean rank with Holm's adjusted p values (0.05) for the AUPR metric measured across all techniques in networks with 250 to 2,000 genes. 178
7.15 Friedman mean rank with Holm's adjusted p values (0.05) for the AUROC metric measured across all techniques in networks with 250 to 2,000 genes. 178

8.1 Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.192
8.2 Friedman mean rank with Holm's adjusted p values (0.05) for AUROC. 192

9.1 Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.222
9.2 Friedman mean rank with Holm's adjusted p values (0.05) for AUROC. 222
9.3 Friedman mean rank with Holm's adjusted p values (0.05) for AUPR.225

9.4 Friedman mean rank with Holm's adjusted p values (0.05) for AUROC. 225

10.1 Configurable parameters of MOEBA-BIO framework. 235
10.2 Methodological and implementation-level comparison of MOEBA-BIO with representative non-traditional encoding proposals. 247
10.3 Friedman mean rank with Holm's adjusted p values (0.05) for Clustering Error. 250

11.1 Median of each front's medians and maximum of each front's maxima for the clustering error complementary metric after running with 5 replicates of the winning configuration and three other candidates on the gene expression dataset generated by FABIA. . 267
11.2 Hypervolume values for different candidates of the unsupervised phase and the winning configuration. In particular, the median of 5 independent runs on the FABIA simulated dataset is presented. 267
11.3 Friedman mean rank with Holm's adjusted p values (0.05) for the ratio of enriched biclusters with $alpha = 0.05$. 271
11.4 Friedman mean rank with Holm's adjusted p values (0.05) for the ratio of enriched biclusters with $alpha = 0.005$. 272
11.5 Friedman mean rank with Holm's adjusted p values (0.05) for the ratio of enriched biclusters with $alpha = 0.00001$. 272

List of Figures

1 Representación gráfica de las contribuciones desarrolladas en esta tesis, organizadas en torno a tres áreas principales: inferencia de redes de regulación génica (verde), biclustering aplicado a datos biomédicos (azul) y diseño algorítmico (rojo). Las flechas de colores indican la vinculación de cada trabajo con los conceptos metodológicos o aplicados dentro de cada área, mientras que las flechas negras reflejan evolución directa o influencia entre propuestas. Las contribuciones se representan mediante recuadros grises ubicados en las intersecciones temáticas que abordan. En la esquina superior izquierda de cada recuadro se especifica su tipo de difusión: Q1 y Q2 indican publicación en revistas del primer o segundo cuartil en categorías de Ciencias Computacionales y Biología; IC señala difusión en congreso internacional; y RC, en congreso de ámbito regional. 11

1.1 Graphical representation of the contributions developed in this thesis, organized around three main areas: inference of gene regulatory networks (green), biclustering applied to biomedical data (blue), and algorithm design (red). The colored arrows indicate the connection of each work with methodological or applied concepts within each area, while the black arrows reflect direct evolution or influence between proposals. The contributions are represented by gray boxes placed at the thematic intersections they address. The top-left corner of each box specifies its type of dissemination: Q1 and Q2 indicate publication in first- or second-quartile journals in the fields of Computer Science and Biology; IC refers to dissemination at an international conference; and RC, at a regional-level conference. 30

2.1 Biological basis of a gene regulatory network. The left part shows the molecular mechanisms that give rise to a gene regulatory network. Each gene (colored boxes) is transcribed into a messenger RNA (mRNA) molecule, which is then translated into a protein. Some of these proteins act as transcription factors (TFs), capable of binding to specific DNA sequences to modulate (activate or repress) the expression of other genes. The right part represents the resulting gene regulatory network, a computational abstraction where the nodes (G1–G4) correspond to genes and the directed edges indicate regulatory relationships between them. 35

2.2 Conceptual representation of gene co-expression. On the left, the expression profiles of different genes are shown, captured under various experimental conditions or over time. When two or more genes exhibit similar patterns, such as a simultaneous increase or decrease in their activity, they are considered co-expressed. On the right, it is illustrated how these genes tend to be regulated by common factors, suggesting a possible functional coordination within the biological system. 37

2.3 Pareto front in a two-dimensional multi-objective optimization problem. The figure exemplifies a case with two objectives to minimize, $f_1(x)$ and $f_2(x)$. The green points represent non-dominated solutions that form the Pareto front: none of them can be improved in one objective without worsening in the other. Conversely, the blue solutions are dominated, as there are other solutions better in at least one objective and no worse in the others. 41

2.4 General schematic of an evolutionary ensemble learning approach. Starting from a dataset, multiple base models are trained and used as components to build the individuals of the population. Each individual represents a specific combination of models, combining them in different ways or even taking different configurations of the same models. These individuals are evaluated based on their performance and undergo an evolutionary process with selection, crossover, and mutation operators, which enables the generation of new combinations and progressively improves the quality of the ensemble. 45

2.5 Examples of structural patterns detectable through biclustering. Each matrix shows a different type of pattern that can appear in subsets of rows and columns within a data matrix. From left to right: constant bicluster (all values are equal), constant rows (each row has a constant value), constant columns (each column maintains its constant value), additive coherence (values follow an additive pattern between rows and columns), and multiplicative coherence (values vary following a proportional relationship). These patterns reflect different types of relationships that biclustering algorithms aim to identify in the data. 49

2.6 Possible combinations of coverage and overlap between biclusters. The figure shows the different scenarios that can arise in a biclustering solution according to the degree of coverage (partial or total) and the presence or absence of overlap between biclusters, whether in rows, columns, or both dimensions. These combinations determine the structural constraints that a biclustering algorithm may or may not allow, and directly affect the complexity of the problem and the interpretation of the results. 50

3.1 Interpretation of each coding of individuals in the approximate Pareto front obtained by biclustering evolutionary algorithms. . . 74

5.1 Architecture and workflow covered in GENECI. First, the execution of multiple individual inference techniques in parallel is enabled by encapsulating their implementations in Docker containers. After that, their results are normalized and collected by the evolutionary algorithm in order to optimize weight vectors that assign a value to each technique. The weight vectors are iteratively subjected to evaluation (depending on the quality and topology of the consensus networks they represent), selection, crossover, mutation and finally an additional repair step to keep the sum of values at unity. 96

5.2 For the first 10-gene yeast network of the DREAM3 challenge, the gene networks inferred by the individual techniques and the consensus gene network computed in the run whose AUROC corresponds to the median exposed in Table 5.3 are illustrated. Graphs attempt to represent gene regulatory networks by setting up genes in the form of nodes and interactions through links. In addition, it can be seen that the directionality and confidence of these interactions are represented in these networks. 111

5.3 Boxplots of the fitness values and weights over the 25 independent runs. The first 5 graphs represent the distribution of the weights assigned by GENECI across all the runs for each of the techniques. Finally, the sixth figure shows the distribution of the fitness values obtained in the different runs performed. 113

5.4 For the first 100-gene Ecoli network of the DREAM3 challenge, the consensus gene network calculated in the run whose AUROC corresponds to the median illustrated in Table 5.5 is plotted on the left. On the right, the evolution of the fitness values obtained during the 25 runs and a violin plot representing the distribution of their corresponding final values. 115

5.5 For Net-4 10-gene of the DREAM4 challenge, the gene networks inferred by the individual techniques and the consensus gene network computed in the run whose AUROC corresponds to the median exposed in Table 5.6 are illustrated. In these graphs we can see how each gene corresponds to a node and each edge to a specific gene interaction. The directionalities are expressed by arrows and the confidence values by the thickness of the links, even specifying their value when this is highly significant. 117

5.6 For the first 100-gene network of the DREAM4 challenge, the consensus gene network calculated in the run whose AUROC corresponds to the median illustrated in Table 5.7 is plotted on the left. On the right, the evolution of the fitness values obtained during the 25 runs and a violin plot representing the distribution of their corresponding final values. 119

5.7 For the "switch on" IRMA network, the gene networks inferred by the individual techniques and the consensus gene network calculated in the run whose AUROC corresponds to the median exposed in Table 5.9 are illustrated. The graphs shown in this figure try to represent the different inferred networks, arranging the genes in the same layout in order to facilitate visual comparison between them. 121

5.8 For Melanoma expression data, gene networks inferred by multiple individual techniques as well as the consensus one produced by GENECI are represented. Graphs attempt to represent gene regulatory networks by setting up genes through nodes and interactions through links. In this case, as they are captures of interactive representations, the directionality of the interactions is not visible to the naked eye. However, the equal arrangement of nodes allows the topology of the different networks to be easily compared. 125

6.1 Succession of phases within the evolutionary process. Individuals are crossed through simulated binary crossover and subsequently subjected to polynomial mutation. Following this, the local search begins where several variations of the individual (encoding a given solution) are compared to select the one whose consensus network is closest to the known interactions. Finally, the individuals are repaired to resume their representation in the form of a weight vector. 129

6.2 Examples of interactions involved in the distance calculation in different executions based on the proportion of the gold standard extracted as a set of "known by the expert" interactions (rows) and the type of distance (columns). The case of extracting 5%, 10%, and 15% of the gold standard for the distance types *all*, *some*, and *one* respectively, is shown. All executions take the same reference in the case of *all*, while for *some* and *one*, there is a certain random component that causes differences on each local search. 131

6.3 Comparison of the AUROC and AUPR performance metrics for the `GENECI` (in blue) and `Memetic Inference` (in orange) algorithmic proposals on each of the networks belonging to the third edition of the dream challenges (horizontal axis). For identification, the challenge prefix (D3) is followed by the size of the network (10, 50 or 100) and finally the initial of the organism on which it is based (Y: Yeast, E: E. coli). The bars indicate the medians of the AUPR values and the lines with markers represent the medians of the AUROC values for each network. The AUPR and AUROC values are displayed on separate vertical axes due to their different measurement scales, reserving the left axis for AUPR and the right axis for AUROC. 136

6.4 Comparison of the AUROC and AUPR performance metrics for the `GENECI` (in blue) and `Memetic Inference` (in orange) algorithmic proposals on each of the networks belonging to the fourth edition of the dream challenges (horizontal axis). The nomenclature and interpretation of the graph are identical to those in Figure 6.3. . 137

7.1 Workflow Implemented in MO-GENECI. After providing the input expression data, the inference techniques are executed in parallel thanks to their encapsulation in Docker containers. Once the results of all techniques are obtained, the lists containing confidence values for each interaction are handed over to the multi-objective evolutionary algorithm. This algorithm generates an initial random population (weight vectors) and undergoes the iterative process of evaluation (*Quality*, *Degree distribution*, *Motifs*), selection, crossover, and mutation until the maximum number of iterations is reached. Upon completion of the multi-objective evolutionary algorithm, the result consists of the set of non-dominated solutions from the last generation [249]. 141

7.2 Flowchart of the multi-objective evolutionary algorithm developed in this proposal. The algorithm starts with the generation of an initial population, whose representation is in the form of a vector of weights (simplex). The individuals are evaluated for each of the fitness functions, previously applying the conversion of the individual to its corresponding consensus network. This is followed by a selection process guided by a binary tournament. Finally, the individuals are subjected to the crossover and mutation operators selected for the representation of this problem, giving rise to the next generation of individuals. 142

7.3 Example of mutation of an individual belonging to an execution trying to agree on 9 inference techniques. In red are highlighted the members of team 1 (intended to reduce their values), in green the members of team 2 (intended to increase their values), in blue the value of the mutation strength factor (set in this case to 0.2) and finally the values finally modified in yellow. 147

7.4 Visualization of a degree distribution following a power-law [256]. In this type of distribution, most nodes have only a few connections, while a small number of highly connected hubs concentrate a large fraction of the links. This characteristic 'long tail' pattern is frequently observed in biological networks, where scale-free organization enables robustness and modularity. 151

7.5 Motifs considered in the study of the third objective function. These motifs represent fundamental regulatory patterns commonly found in gene networks: (i) Regulatory Route, a sequential path of interactions with feedback to the initial node, reflecting ordered information flow; (ii) Differentiation, a branching pathway without cycles, essential for controlling cell specialization; (iii) Bifurcation, a node splitting into multiple successors, allowing diverse responses to signals; (iv) Coupling, reciprocal regulation between two nodes, contributing to stability and homeostasis; (v) Feed-forward loop, where a regulator influences a target both directly and indirectly via an intermediate factor, conferring robustness against fluctuations; (vi) Feedback loop with co-regulation, which adds reciprocal interactions between regulators and targets, enhancing coordination and precision; (vii) Biparallel, representing indirect multi-path regulation that, taken together, reflects strong functional relationships; and (viii) Co-regulation, where multiple regulators converge on the same gene, enabling fine-tuned and context-specific expression control. These motifs are used as structural building blocks to evaluate the proposed motif-based objective function. 154

7.6 Evolution of fitness functions for the 200-node network constructed from scratch with a scale-free degree distribution and subjected to overexpression perturbation. 161

7.7 Set of non-dominated solutions in Pareto front approximation from the final population. 163

7.8 Extended parallel coordinate plots showing Pareto front individuals. The first columns are optimization objectives, the last ones quality metrics, and color intensity encodes the mean accuracy (acc_mean). 165

7.9 AUROC and AUPR values for networks up to 25 genes. Each column represents a gene network, and each row represents a different metric. The upper row shows AUROC values, while the lower row displays AUPR values. Each cell contains two sets of bars stacked from lowest to highest height. In the first set of bars, the values obtained by BEST_MO-GENECI and MEDIAN_MO-GENECI are represented, while the second set contains the values from the other individual techniques. The number of individual techniques surpassed by BEST_MO-GENECI is indicated on the first set of bars, which can be visualized thanks to the dashed horizontal line that sets the threshold. The name of the winning individual technique that achieved the best results for the given network and metric is displayed on the second set of bars. 167

7.10 AUROC and AUPR values for networks between 25 and 110 genes in size. Each plot displays the results for a specific metric. The upper plot shows the AUROC values, while the lower one shows the AUPR values. In both plots, the vertical axis represents the values of the corresponding metric, and the horizontal axis represents the different networks considered in the figure. Each technique is assigned a color and represented by a series of points connected by dashed lines. To facilitate the comparison of BEST_MO-GENECI with the other techniques, a continuous representation of its curve and shading over its area have been used. 168

7.11 AUROC and AUPR values for networks between 110 and 250 genes in size. In this case, radar charts have been used, which clearly represent the specializations of different techniques. Once again, shading has been applied to BEST_MO-GENECI (yellow), highlighting its ability to cover all domains of expertise. 169

7.12 AUROC and AUPR values for networks between 250 and 2,000 genes in size. La explicación de esta representación es la misma que en la Figura 7.9 169

7.13 Scatter plot illustrating the relationship between the number of genes and the execution time of the algorithm. Each point represents a specific execution instance. The plot provides insight into the performance characteristics of the algorithm across different gene counts. 180

8.1 Conceptual diagram of the proposed multi-objective evolutionary algorithm. Individuals, represented as weight vectors, are converted into their respective consensus regulatory networks for subsequent evaluation. Three objectives are considered: Quality, Degree Distribution, and Motifs. Once the performance of each individual is obtained for these objectives, selection is carried out based on the reference point established by the domain expert. The expert draws from various sources and personal experience to guide the population of individuals toward a region of interest in the problem. Afterward, the selected individuals undergo crossover and mutation to form a new generation of individuals. . 186

8.2 Phases carried out during the experimentation of this study, illustrating the workflow from multiple independent runs and filtering of non-dominated individuals, through validation with AUROC and AUPR, to the generation and comparison of refined populations.190

8.3 Graphical representation for different gene regulatory networks of the three most relevant populations in the study. Specifically, the population of non-dominated solutions from the 15 independent runs of: (1) the original algorithm, colored based on the average of the AUPR and AUROC metrics; (2) the algorithm with preference-based selection guided by the point corresponding to the top 10 individuals with the best AUPR (green diamond), colored based on AUPR values; and (3) the algorithm with preference-based selection guided by the point corresponding to the top 10 individuals with the best AUROC (orange diamond), colored based on AUROC values. 193

8.4 Median of the 5 independent runs of the percentage of dominated solutions from the original front (trimmed by the reference point) in each generation for large networks ($\geq$ 100 nodes) using the winning configurations (one for each metric). 195

9.1 Standard workflow of BIO-INSIGHT. Starting from gene expression data, the 26 available inference techniques are executed in parallel, with computational load distributed based on their expected cost (left). Each technique infers a different gene regulatory network, producing a set of individual solutions (center-left). These networks are integrated through an asynchronous and parallel many-objective evolutionary algorithm, which optimizes a weighted voting system using six biologically driven objectives: interaction quality (O1), presence of biological motifs (O2), eigenvector centrality distribution (O3), reduction of non-essential interactions (O4), node degree distribution (O5), and dynamic stability (O6). Each individual in the population represents a set of weights for the inference techniques and is evaluated according to how well its resulting consensus network satisfies the biological objectives (center). The evolutionary process generates a Pareto front of optimal trade-offs, where each solution corresponds to a consensus network that balances different biological properties. These final networks (right) are supported both by the inference techniques and by known structural characteristics of real-world gene regulatory systems. 200

9.2 Comparison of the networks inferred by the two most accurate techniques for a given problem. Each technique is represented by a different color, arrows indicate the direction of regulation, and their thickness reflects the confidence level of the interaction. . . 213

9.3 Chord diagram for the solution front obtained by BIO-INSIGHT after optimizing the consensus of the techniques applied to the first network from the DREAM4 challenge, with a size of 10 nodes. In this diagram, each objective is represented as a circular trapezoid. Inside the trapezoid, a histogram illustrates the solutions' distribution across the corresponding objective's normalized values. Dashed lines indicate the maximum and minimum values within the histogram. Small boxes at the base of the circular trapezoid represent subsets of individuals grouped based on their proximity in the objective score. These boxes are initially interactive and allow the selection of individuals to be displayed in the core of the diagram. Since this document is static, a snapshot has been taken, selecting the worst-performing individuals for each fitness function to highlight the opposition between the objectives of the algorithm appropriately. 215

9.4 Polar plot locating each solution from the front obtained by BIO-INSIGHT for the second yeast network from the DREAM3 challenge, with a size of 10 nodes. In these plots, individuals are represented by points positioned according to the weights assigned to different techniques and are coloured based on their accuracy level. Additionally, larger markers represent simulated solutions (not part of the front), corresponding to assigning all the weight to a single technique. To the right of the radar, a sidebar displays the colour gradient associated with the accuracy metric in question. In this bar, black markers indicate the accuracy values of each technique, while white markers highlight the accuracy of BIO-INSIGHT's best solution as well as the median of the front. 216

9.5 Representation of the best solution obtained by BIO-INSIGHT for the *Emericella nidulans FGSC A4* network extracted from the BioGRID repository. Genes are coloured based on their neighbourhood, arrows indicate the direction of regulation, and their thickness represents the confidence level of the interaction. 218

9.6 In these plots, the moving medians of the normalized objective values for the individuals forming the front obtained by BIO-INSIGHT are represented and sorted for each accuracy metric. Additionally, each moving median is shaded by the interquartile range, allowing an outline of the diversity of objective values at each accuracy level. 219

9.7 In these plots, the moving medians of the weights assigned to each technique in the front obtained by BIO-INSIGHT for the *Glycine max* network from BioGRID are represented and sorted for each objective of the algorithm. Additionally, as in Figure 9.6, each moving median is shaded by the interquartile range, which in this case outlines the diversity of the weights assigned to each technique at each objective function value. 220

9.8 Comparison of the AUPR and AUROC metrics obtained by BIO-INSIGHT (blue box) for a diverse set of networks from different sources, in relation to MO-GENECI (orange box), other consensus strategies (diamonds), and individual techniques (circles). 221

9.9 Comparison of execution time depending on the network size for MO-GENECI (blue) and BIO-INSIGHT (orange). 223

9.10 Differential gene enrichment and unique interactions across conditions. (A) Pathway enrichment analysis comparing gene expression between control and ME/CFS, FM, and co-diagnosis study groups. Dot size represents the number of genes involved while colour indicates the -log10(FDR) significance. (B) Venn diagrams displaying the overlap of predicted gene interactions across study groups. (C) Unique pathway interactions for the control and ME/CFS groups, highlighting significantly enriched biological processes. . 227

10.1 An example of the COMPLETE and PARTIAL encodings, as well as their common intermediate state, on a simplified 8x8 matrix. A solution consisting of 4 biclusters with overlapping columns is shown. 237

10.2 Structure of the specific self-configurator in the MOEBA-BIO framework. It consists of two wrapper genetic algorithms. The outer one handles the self-configuration of objectives through a supervised evaluation that depends on the gold standards of the input data. The inner wrapper handles the self-configuration of the remaining parameters, based on unsupervised metrics such as hypervolume. 244

10.3 Visual representation of the MOEBA-BIO self-configuration mechanism. The process is structured in two nested evolutionary phases: a supervised outer layer (left) and an unsupervised inner layer (right). Each individual in the outer population encodes a specific combination of objective functions and subparameters. For each one, the inner layer determines the best technical configuration (algorithm, operator probabilities, population size, etc.) based on hypervolume (HV). The final evaluation of each outer individual is computed by averaging the clustering error (CE) of the top 5 Pareto-optimal solutions obtained for each dataset. The best outer individual (i.e., configuration) is selected as the winner. 246

10.4 Biclustering solution obtained by MOEBA-BIO using the partial representation for the simulated dataset with 3 biclusters and seed 102. The green nodes represent the biclusters from the gold standard. Each purple node in this encoding represents a partial solution from the front obtained by the algorithm since, in this case, the combination of all biclusters in the front constitutes the real biclustering solution to the problem. The graph's edges refer to the intersection between biclusters, with their thickness increasing in proportion to the number of shared rows and columns. To avoid adding noise to the graph, intersections between nodes of the same color caused by possible overlaps between biclusters have been ignored. 251

10.5 Examples of solutions obtained by MOEBA-BIO during the first phase of experimentation using the complete representation and the new objective functions of adaptive bicluster size and bicluster differentiation. The interpretation of the graphs is the same as discussed in Figure 10.4, except that in this case, all purple nodes belong to a single solution from the front obtained by the algorithm, as in the complete encoding, a solution from the front is equivalent to a real solution to the problem. On the left side of the figure, solutions that are balanced across the three objectives are presented. On the right side, extreme solutions from the front are shown, specifically one for the case of extreme optimization of Adaptive Bicluster Size and another for the case of extreme optimization of MSR. 252

10.6 Evolution of the minimum fitness values of each objective during the MOEBA-BIO run using complete representation and the new objective functions on the simulated dataset with an overlap level of 20 and seed 101. 254

10.7 Distribution of the number of biclusters contained in each individual per generation in the MOEBA-BIO run using complete representation and the new objective functions on the simulated dataset with an overlap level of 20 and seed 101. 255

10.8 Set of non-dominated solutions in Pareto front approximation from the final population obtained by MOEBA-BIO using complete representation and the new objective functions on the simulated dataset. 255

11.1 The extended framework of MOEBA-BIO outlines its main phases and contributions. It starts with the input of the data matrix and the specification of the type of each column (1). Based on the selected encoding strategy (in this case, COMPLETE), MOEBA-BIO initializes the population for the evolutionary algorithm. The execution then proceeds through an iterative process until the stopping condition, defined as reaching a maximum number of evaluations, is met. Each individual is translated into an intermediate representation common to all encodings (2), enhancing interpretability and facilitating evaluation. Individuals are assessed according to the selected objective functions, with normalized fitness scores ranging from 0 (best case) to 1 (worst case). These scores are stored in the original individual, after which the standard evolutionary steps (selection, crossover, and mutation) are applied (3), depending on the chosen algorithm and the operators defined during parameter configuration. 258

11.2 Tracking the execution of the autoconfigurator for gene co-expression data. The evolution of the outer loop (supervised phase) is shown on the left and the inner loop (unsupervised phase) on the right. 264

11.3 Parallel coordinate plot for the front obtained by MOEBA-BIO using complete representation and the best configuration found by the autoconfigurator for the first instance of the gene expression data simulator. 266

11.4 Comparison of the self-generated MOEBA-BIO algorithm for the gene co-expression domain concerning different recognised state-of-the-art techniques. For each instance, the values of the supervised metrics: Clustering Error, ScoreAyadi and ScoreEren (recovery and relevance) are compared. 269

11.5 Comparison of the time required by each methodology for each instance of the synthetic dataset generated by the FABIA R package. The time required by MOEBA-BIO is divided into two fragments, detailing the time required for the initial GRN inference performed by GENIE3 for the regulatory coherence objective. 270

11.6 Comparison of the proportion of enriched biclusters obtained by each methodology on the real-world gene expression dataset, evaluated under different significance thresholds ($\alpha \in \{0.05, 0.005, 0.00001\}$). Higher proportions indicate a stronger functional coherence among the genes within the detected biclusters. The values obtained for the best solution of the MOEBA-BIO Pareto front plot a colored area that allows visualizing the dominance of the proposal. 271

11.7 Median and range of execution times (in seconds) according to input matrix size (number of rows). The plot differentiates between the total execution time of the biclustering process and the inference time required by GENIE3 to instantiate the *Regulatory coherence* objective. 273

A.1 Top-level commands available in `GENECI`, as shown by the `geneci -help` command. 318

List of Algorithms

1 Main code of the generalized EA 98
2 First term of the fitness function: Quality 102
3 Second term of the fitness function: Topology 103
4 Main code of the local search phase 130
5 MO-GENECI Algorithm. 143
6 Main code of Simplex Mutation operator. 145
7 First fitness function: Quality. 148
8 Second fitness function: Degree Distribution. 151
9 Third fitness function: Motifs. 153
10 PBEvoGen Algorithm. 187
11 Evolutionary optimization of the consensus in BIO-INSIGHT . . . 201
12 Third objective: Eigenvector Distribution. 204
13 Fourth objective: Reduce Non-Essential Interactions. 206
14 Sixth objective: Dynamicity 209
15 Self-configuring scheme of evolutionary metaheuristics offered by MOEBA-BIO. 233
16 Adaptive bSize fitness function. 240
17 Bicluster Differentiation fitness function. 242
18 Regulatory Coherence fitness function. 260

Appendix A

Gene Regulatory Network Consensus Inference User Guide

This appendix provides a practical user guide for the `GENECI` package, a Python-based tool developed in the context of this thesis to facilitate gene regulatory network inference through ensemble learning and evolutionary optimization. The guide includes installation instructions, a step-by-step example of a typical execution workflow, and an overview of the output files generated during the process. It is intended to support researchers and practitioners in reproducing the experiments or applying the tool to new gene expression datasets.

Prerequisites

- Python $\geq$ 3.9
- Docker

Installation

```
pip install GENECI
```

Output

To execute `GENECI`, the `run` command is provided with the file containing the expression levels of the genes that make up the network, the list of techniques to

be agreed upon and the values of the different algorithm parameters in the event of not wishing to use those established by default. If more than one proposed objectives are used, the following files are obtained after execution:

- `FUN.csv`: List with fitness values for each individual in the final population for each of the objective functions.
- `VAR.csv`: List of winning weight vectors, i.e., individuals from the last generation.
- `fitness_evolution.txt` and `fitness_evolution.html`: In each generation, the most optimal value found for each objective function is recorded. These values are stored in the text file and subsequently represented in plots in HTML format.
- `parallel_coordinates.html`: File containing a graphical representation of parallel coordinates. Each column refers to a specific objective function, and each horizontal line represents an individual from the final population. This plot is very useful to observe conflicts between different fitness functions in a multi-objective evolutionary algorithm.
- `pareto_front.html`: Pareto front represented in a 3D or 2D plot, where each axis refers to a different fitness function.

Example procedure

Step 1: Obtain simulated expression data and their respective gold standards

To do this, we have two options:

Extraction: Use the `extract-data` command to download expression data from known challenges and benchmarks.

```
# Expression data
geneci extract-data expression-data \
       --database DREAM4 \
       --output-dir input_data

# Gold standard
geneci extract-data gold-standard \
       --database DREAM4 \
       --output-dir input_data
```

Simulation: Use the `generate-data` command to generate expression data through the SysGenSIM simulator. In this case, data can be generated from scratch by choosing a particular node size and distribution, or from real biological networks stored in multiple databases. In both cases, the type of perturbation to be simulated must be specified.

```
# From scratch
geneci generate-data generate-from-scratch \
       --topology eipo-modular \
       --network-size 20 \
       --perturbation knockout \
       --output-dir input_data

# From real network
geneci generate-data download-real-network \
       --database BioGrid \
       --id Oryza_sativa_Japonica \
       --output-dir input_data
geneci generate-data generate-from-real-network \
       --real-list-of-links ../BioGrid_Oryza_sativa_Japonica.tsv \
       --perturbation overexpression \
       --output-dir input_data
```

Step 2: Inference and consensus of networks for the selected expression data

To perform this task, you can make use of the `run` command or proceed to an equivalent execution consisting of the `infer-network` and `optimize-ensemble` commands. This can be very useful when you need to incorporate external trust lists or run the evolutionary algorithm with different configurations on the same files, without the need to infer them several times.

Form 1: Procedure prefixed by the command `run`.

Minimal example:

```
geneci run \
    --expression-data ../dream4_100_01_exp.csv \
    --technique ARACNE --technique BC3NET \
    --function Quality \
    --algorithm GA
```


Detailed example with full customization:

```
geneci run \
    --expression-data ../dream4_100_01_exp.csv \
    --technique ARACNE --technique BC3NET \
    --technique C3NET --technique GENIE3_RF \
    --crossover-probability 0.9 \
    --num-parents 3 \
    --mutation-probability 0.05 \
    --mutation-strength 0.1 \
    --population-size 100 \
    --num-evaluations 50000 \
    --cut-off-criteria PercLinksWithBestConf \
    --cut-off-value 0.4 \
    --function Quality --function DegreeDistribution --function Motifs \
    --algorithm NSGAII \
    --plot-results \
    --output-dir inferred_networks
```

Optional enhancements:

- `-time-series` for dynamic objectives like `Loyalty`.
- `-known-interactions`, `-memetic-distance-type`, `-memetic-probability` to enable local search during optimization (see Chapter 6).
- `-reference-point` to guide the multi-objective selection towards a specific zone of the objective space. The format must be `f1;f2;f3`, where each value represents the target score for each objective (see Chapter 8).
- `-compare-performance` to compare the performance of the current run with a reference front. A plot will be produced showing the percentage of reference solutions that have already been dominated by the current population.
- `-threads` or `-str-threads` for parallel execution.
- `-no-plot-results` to skip graphical representations.

Form 2: Division of the procedure into several commands

```
# 1. Inference using individual techniques
geneci infer-network \
       --expression-data .../dream4_100_01_exp.csv \
       --technique ARACNE --technique BC3NET \
       --technique C3NET --technique CLR \
```

```
       --technique ... \
       --output-dir inferred_networks

# 2. Optimize the assembly of the trust lists resulting from the above
    command
geneci optimize-ensemble \
       --confidence-list inferred_networks/GRN_LOCPCACMI.csv \
       --confidence-list inferred_networks/GRN_BC3NET.csv \
       --confidence-list ... \
       --crossover-probability 0.9 \
       --mutation-probability 0.05 \
       --population-size 100 \
       --num-parents 3 \
       --mutation-strength 0.1 \
       --num-evaluations 50000 \
       --cut-off-criteria PercLinksWithBestConf \
       --cut-off-value 0.4 \
       --function Quality --function DegreeDistribution \
       --function Motifs --function ... \
       --algorithm NSGAII \
       --plot-results \
       --output-dir inferred_networks
```

Consensus under own criteria: Assign specific weights to each of the files resulting from each technique. In case the researcher has some experience in this domain, he can determine for himself the weights he wants to assign to each inferred network to build his own consensus network.

```
geneci weighted-confidence \
       --weight-file-summand 0.5*inferred_networks/GRN_GENIE3_ET.csv \
       --weight-file-summand 0.25*inferred_networks/GRN_CMI2NI.csv \
       --weight-file-summand 0.25*inferred_networks/GRN_PIDC.csv \
       --output-file inferred_networks/weighted_confidence.csv
```

Step 3: Representation of inferred networks

Representation of the inferred networks is performed using the `draw-network` command.

```
geneci draw-network \
       --confidence-list inferred_networks/GRN_LOCPCACMI.csv \
       --confidence-list inferred_networks/GRN_BC3NET.csv \
       --confidence-list ... \
```


```
       --mode Both \
       --nodes-distribution Spring \
       --output-folder inferred_networks/network_graphics
```

Step 4: Validation of the inferred gene network

Validation of the quality of the inferred gene network with respect to the gold standard. Two procedures have been implemented: one specific to networks extracted from DREAM challenges, and another generic one that approaches the problem as a binary classification exercise. In both cases, the evaluation procedure can be applied to a list of interactions with their respective confidence levels, a certain weight distribution referring to the consensus, or even a Pareto front generated by our multi-objective algorithm mode that allows the representation of a parallel coordinate plot including both fitness functions and AUROC and AUPR metrics (which is quite useful for identifying high-quality regions).

DREAM: For the evaluation of networks from DREAM challenges, the evaluation data must be previously downloaded using the `extract-data` command and the `evaluation-data` subcommand, which requires providing the database and credentials of an account on the Synapse platform. After that, the `evaluate` command is used, followed by the `dream-prediction` subcommand to access the three input options mentioned above. In any case, the challenge identifier, network identifier, evaluation files, and input files need to be specified. The input files will depend on the chosen option: `dream-list-of-links`, `dream-weight--distribution` or `dream-pareto-front`.

```
# 1. Download evaluation data
geneci extract-data evaluation-data \
       --database DREAM4 \
       --username TFM-SynapseAccount \
       --password TFM-SynapsePassword

# 2. Evaluate the accuracy of the inferred consensus network.
geneci evaluate dream-prediction dream-list-of-links \
       --challenge D4C2 \
       --network-id 100_1 \
       --synapse-file input_data/DREAM4/EVAL/pdf_size100_1.mat \
       --confidence-list inferred_networks/dream4_100_01/consensus.csv
```

Generic: For network validation using a generic procedure, we directly use the `evaluate` command followed by the `generic-prediction` subcommand. This gives us access to the three types of input mentioned earlier, to which we must

provide the gold standard of the problem and the relevant input files: `generic--list-of-links`, `generic-weight-distribution`, and `generic-pareto-front`.

```
geneci evaluate generic-prediction generic-list-of-links \
       --confidence-list inferred_networks/consensus.csv \
       --gs-binary-matrix input_data/simulated/GS/sim_BioGrid_gs.csv
```

Step 5: Binarization of the inferred gene network

In many cases, it is useful to apply a cutoff criterion to convert a list of confidence values into a real network that asserts the specific interaction between genes. For this purpose, the `apply-cut` command is used, which is provided with the list of confidence values, the cutoff criterion and its corresponding threshold value.

```
geneci apply-cut \
       --confidence-list inferred_networks/dream4_100_01/consensus.csv \
       --cut-off-criteria PercLinksWithBestConf \
       --cut-off-value 0.4 \
       --output-file inferred_networks/dream4_100_01/binarized.csv
```

Additional commands and resources

Beyond the commands explained throughout the example procedure, the `GENECI` package includes many other commands and subcommands that can be explored for more advanced or specific use cases. Full documentation, source code, and updates are available at:

`https://github.com/AdrianSeguraOrtiz/GENECI`

Figure A.1 shows a screenshot of the help command output, where all top-level commands currently available in `GENECI` can be seen.

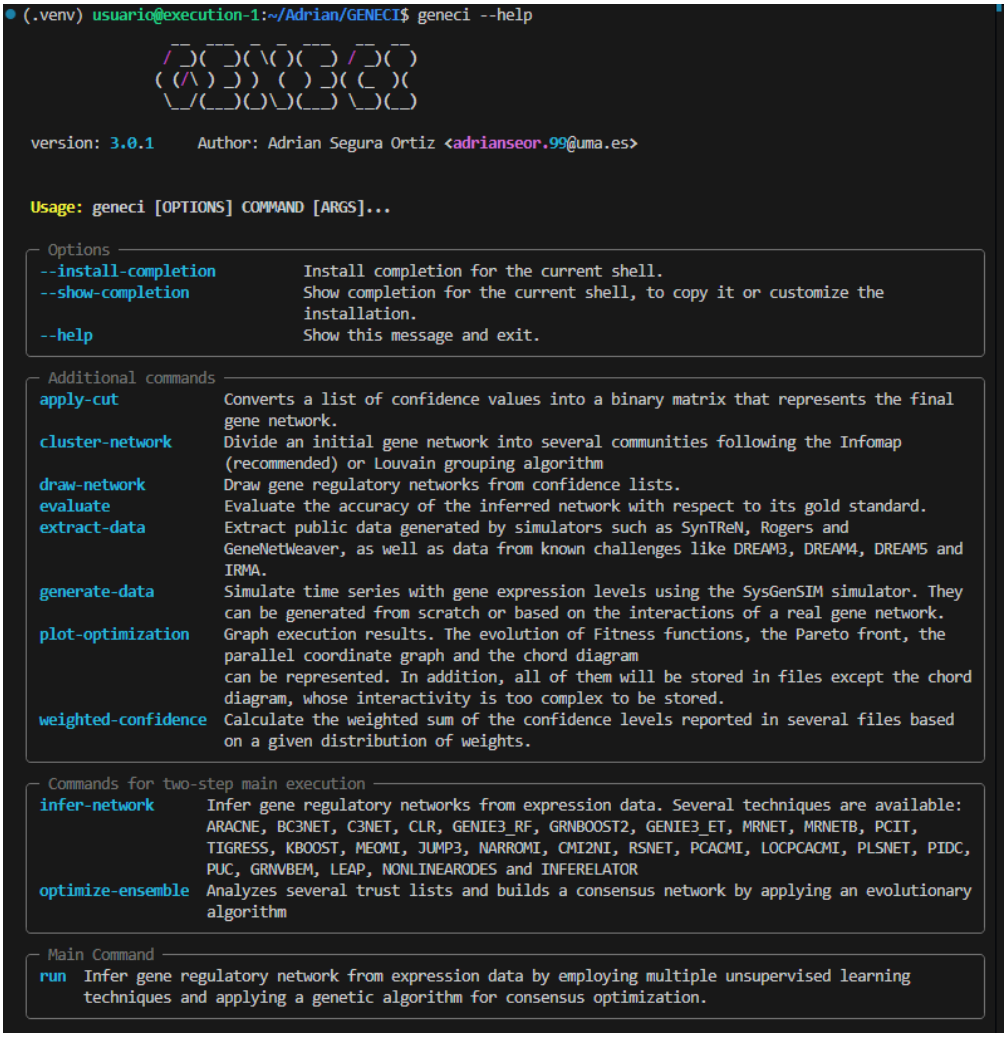


Figure A.1: Top-level commands available in `GENECI`, as shown by the `geneci -help` command.

Appendix B

Evolutionary Biclustering Algorithm for Expression Data User Guide

This appendix provides a detailed guide for executing the `MOEBA-BIO` framework from the command line. `MOEBA-BIO` (Multi-Objective Evolutionary Biclustering Algorithm for BIOmedical applications) is designed to discover coherent and relevant biclusters from complex biomedical datasets. It allows users to fully customize every component of the evolutionary process, including the representation of individuals, the optimization objectives, the search algorithm, and the variation operators.

The tool is implemented in Java and can be executed through the `RunnerMOEBA` class, which offers a comprehensive set of options via a command-line interface built with Picocli. This guide describes how to configure a specific run of `MOEBA-BIO` according to user preferences or suggestions provided by an autoconfigurator.

Prerequisites

To run `MOEBA-BIO`, the following requirements must be met:

- Java $\geq$ 17
- Maven

Basic Execution Command

Once compiled, MOEBA-BIO can be executed from the command line using the following format:

```
java -cp target/moeba.jar moeba.Runner [OPTIONS]
```

Each [OPTION] corresponds to a component of the evolutionary framework that can be configured by the user.

Input Files

To run an execution of MOEBA-BIO, two input files must be provided: a dataset file in CSV format and a JSON descriptor specifying the data type of each column. These files are essential for correctly interpreting the structure and content of the data matrix.

Dataset File (-input-dataset)

The dataset must be a comma-separated values (CSV) file, where each row typically corresponds to a biological entity (such as a gene), and each column to a condition, sample, or patient. The first row should contain the header with the column names, and each subsequent row must contain numerical values corresponding to each column.

Example:

```
sample1,sample2,sample3,sample4
3.51,4.72,5.39,6.01
2.84,3.17,3.85,4.10
...
```

This structure allows MOEBA-BIO to work on matrices where rows are grouped into biclusters based on shared patterns across columns (samples).

Column Type File (-input-column-types)

This is a JSON file in which each key corresponds to a column name from the dataset, and its value indicates the data type for that column. This information is required to convert the input data into a numerical matrix suitable for evolutionary evaluation.

Example:

```
{
    "sample1": "float64",
    "sample2": "float64",
    "sample3": "float64",
    "sample4": "float64"
}
```

In the example above, all four columns are interpreted as continuous numerical variables. The JSON must cover all columns from the dataset, even if they are of the same type.

Supported Data Types

Type label	Java type
`string`	`String.class`
`int`	`Integer.class`
`double`	`Double.class`
`float`	`Float.class`
`float64`	`Float.class`
`boolean`	`Boolean.class`

Important note: Although `MOEBA-BIO` supports the declaration of multiple data types in the input, all currently implemented fitness functions are designed to operate exclusively on **numeric data**. Therefore, columns declared as `string` or `boolean` will be ignored or lead to errors unless future objectives explicitly handle such data types. New objectives tailored to mixed-type or categorical data are planned for future development.

Representation Strategies

One of the core elements of `MOEBA-BIO` is the way in which candidate solutions (i.e., sets of biclusters) are encoded. Users can select the representation using the `-representation` flag. The available options are `GENERIC`, `SPECIFIC`, `INDIVIDUAL`, and `DYNAMIC`.

Important notice:

1. The **PARTIAL** encoding described in this thesis corresponds to the `INDIVIDUAL` option.

2. The **COMPLETE** encoding corresponds to the `GENERIC` option.
3. The `DYNAMIC` representation is currently under development and not yet available.

The `GENERIC` representation offers the most flexibility, as the number of biclusters is not predefined. Each solution consists of a variable-length list of biclusters, where each bicluster is encoded by an ordered list of rows and a binary encoding of column activations. Additionally, a vector of internal cell values is evaluated using a voting threshold to determine column inclusion. The user may optionally define initial limits for the number of biclusters using `-generic-initial-min-num-bics` and `-generic-initial-max-num-bics`.

In contrast, the `SPECIFIC` representation fixes the number of biclusters in advance via the `-specific-num-biclusters` option. In this case, the encoding becomes more structured: each row is assigned to one of the n biclusters, and each bicluster has an associated binary vector indicating the active columns.

The `INDIVIDUAL` representation, on the other hand, encodes a single bicluster per solution. Each individual is defined by two binary vectors: one for rows and another for columns. The global solution emerges from the entire population, where each individual contributes one bicluster. This approach matches the "partial encoding" previously introduced.

Lastly, the `DYNAMIC` representation was conceived as a hybrid between `GENERIC` and `SPECIFIC`, where a central population uses a generic encoding and subpopulations (or islands) progressively refine specific configurations when dominant structures emerge. However, this option is currently not available in the public release.

Extending MOEBA-BIO with New Encoding

`MOEBA-BIO` is built with extensibility in mind. If your problem requires a new encoding or you want to experiment with alternative representations, you can implement a new wrapper by extending the interface `RepresentationWrapper`.

To integrate a new representation, you must provide a concrete implementation that defines:

- **Crossover operator resolution** (`getCrossoverFromString`)
- **Mutation operator resolution** (`getMutationFromString`)
- **Solution decoding logic** (`decodeVAR`)

- **Variable labeling** for output (`getVarLabels`)
- **Population initialization** and **individual creation**
- Optional: **cache key generation** and internal structures

This modular design allows full control over how your encoding behaves in the evolutionary process.

Operator Support

When creating a new representation, you can either:

- **Use general-purpose operators** already included in `MOEBA-BIO` or inherited from the jMetal library.
- **Define custom operators** adapted to your encoding and inject them through the `-crossover-operator` and `-mutation-operator` options.

Defining Optimization Objectives

The optimization process in `MOEBA-BIO` is guided by one or more objective functions, which evaluate the quality of each candidate solution. These objectives are specified using the `-str-fitness-functions` parameter. Multiple objectives can be used simultaneously by separating them with semicolons. For example:

```
--str-fitness-functions BiclusterSizeNormComp;MeanSquaredResidueNorm
```

Each objective function returns a numerical score for a solution or for each bicluster it contains. These scores are then used by the evolutionary algorithm to explore the space of possible biclustering configurations.

Categories of Objective Functions

`MOEBA-BIO` supports three types of fitness functions, depending on the granularity of the evaluation:

(1) Individual Bicluster Objectives

These functions evaluate **each bicluster independently**, based solely on its internal structure (rows and columns). They are defined by extending the class `IndividualBiclusterFitnessFunction`. These are the most common objectives and can be applied across all representations.

Examples include:

- `BiclusterSizeNormComp`: rewards larger biclusters.
- `MeanSquaredResidueNorm`: evaluates internal coherence using MSR.
- `RowVarianceNormComp`: rewards variance across rows, useful for co-expression detection.
- `BiclusterVarianceNorm`: penalizes high variance, emphasizing homogeneity.

These objectives are automatically applied to each bicluster. When using `GENERIC` or `SPECIFIC` representations (i.e., multiple biclusters per individual), the scores are aggregated into a global fitness score using one of the summarization strategies:

```
--summarise-individual-objectives HarmonicMean
```

Available aggregation strategies are `Mean`, `HarmonicMean`, and `GeometricMean`.

(2) Global-Aware Bicluster Objectives

These functions evaluate **each bicluster in the context of the full solution**, allowing the fitness of one bicluster to depend on the presence and structure of others. They are implemented by extending `GenericBiclusterFitnessFunction`.

Examples:

- `BiclusterSizeNumBicsNormComp`: balances bicluster size with the number of biclusters.
- `DistanceBetweenBiclustersNormComp`: promotes dissimilarity between biclusters.

These objectives require access to the full solution (list of biclusters) and are internally aware of other components during evaluation.

(3) Whole-Solution Objectives

Some objectives do not evaluate biclusters individually, but instead assign a score to the entire **solution as a whole**. These are implemented by extending `GlobalFitnessFunction`.

An example is:

- `RegulatoryCoherenceNormComp`: designed for gene co-expression applications, this objective evaluates the modularity of the inferred biclustering solution over a gene regulatory network.

These global functions bypass aggregation and directly return a single score per individual, enabling more complex evaluation strategies based on external biological knowledge or structural constraints.

Parameterized Objectives

Many fitness functions in `MOEBA-BIO` support optional parameters. These are defined by appending them in parentheses after the objective name:

```
--str-fitness-functions BiclusterSizeNumBicsNormComp(rowsweight=0.6,
   coherenceweight=0.5)
```

The parser automatically maps the provided arguments to the corresponding attributes inside the fitness function class.

Implementing a Custom Objective

To extend `MOEBA-BIO` with your own fitness function, you must implement one of the following abstract base classes, depending on the evaluation scope:

1. Evaluate a single bicluster independently → Extend **IndividualBiclusterFitnessFunction** (e.g., `MeanSquaredResidueNorm`).
2. Evaluate a bicluster in the context of others in the same solution → Extend **GenericBiclusterFitnessFunction** (e.g., `BiclusterSizeNumBicsNormComp`).
3. Evaluate the whole solution as a unit (no aggregation) → Extend **GlobalFitnessFunction** (e.g., `RegulatoryCoherenceNormComp`).

Your class must override the `evaluate(...)` method accordingly. If you wish to support parameter configuration from the CLI, override the `configure(Map<String, Object>)` method as well.

Making Your Objective Available from the CLI

In order for your custom fitness function to be recognized and instantiated by the framework when specified via `-str-fitness-functions`, you **must register it** in the `StaticUtils.OBJETIVES_MAP` static map using the following structure:

```
OBJETIVES_MAP.put('myobjective', (str, op) -> {
    Map<String, String> subParams = getSubParams('myobjective', str);
    String sumIndObjs = StaticUtils.getOne('myobjective', subParams, '
        summariseindividualobjectives', op.summariseIndividualObjectives)
    return new MyObjectiveClass(op.data, op.types, op.cache, sumIndObjs);
});
```

Make sure:

- The map key (e.g., `"myobjective"`) matches the lowercase version of your objective name.
- The logic inside the lambda handles optional parameters and instantiates your class correctly.

Once registered, your new function can be directly invoked from the command line like any built-in objective.

Choosing the Optimization Algorithm

`MOEBA-BIO` provides a wide selection of single-objective, multi-objective, and many-objective evolutionary algorithms, which can be selected using the `-str-algorithm` option.

Available algorithms include:

- **Single-objective**: `GA-AsyncParallel`, `GA-SingleThread`
- **Multi-objective**: `NSGAII-AsyncParallel`, `NSGAII-SingleThread`, `MOEAD-SingleThread`, `SMS-EMOA-SingleThread`, `MOCell-SingleThread`, `SPEA2-SingleThread`, `IBEA-SingleThread`, `NSGAIII-SingleThread`, `MOSA-SingleThread`
- **Many-objective**: `NSGAII-ExternalFile-AsyncParallel`

Algorithms may also accept internal parameters:

```
--str-algorithm MOSA-SingleThread(InitialTemperature=0.9)
```

Additional global parameters that control the evolution process include:

- `-population-size`: sets the size of the population (e.g., 100).
- `-max-evaluations`: total number of evaluations to perform.

- `-crossover-probability` and `-mutation-probability`: probabilities of applying crossover and mutation respectively. Progressive mutation rates can also be specified using the format `0.3->0.05`.

Customizing Crossover and Mutation Operators

The variation operators in `MOEBA-BIO` are specified using the `-crossover-operator` and `-mutation-operator` flags. Each flag can include multiple operators separated by semicolons, as in:

```
--crossover-operator RowPermutationCrossover;CellBinaryCrossover
--mutation-operator RowPermutationMutation;CellBinaryMutation
```

Note: The use of multiple operators does *not* mean that they are applied sequentially to the entire solution. Instead, each operator is applied to a different part of the encoding, as defined by the representation.

For example, in the `INDIVIDUAL` representation, each individual is composed of two binary vectors, one for rows and one for columns. In this case, different operators may be applied to each component independently.

The specific mapping between parts of the encoding and operators is handled internally by the corresponding representation wrapper. Each wrapper defines:

- Which operators are supported.
- How many components of the genotype need variation.
- How to distribute the configured operators among those components.

To develop or use a new operator, simply declare it within the corresponding wrapper and expose it via the command line interface. Operators can also be parameterized using a parenthesis-based syntax:

```
RowBiclusterMixedCrossover(shuffleRate=0.5)
```

This flexible design allows precise customization and targeted variation for each structural component of a solution.

Observers and Cache Management

`MOEBA-BIO` incorporates internal mechanisms to avoid redundant computations and to track the evolutionary progress over time.

Caching

Two types of caches are supported:

- `-have-external-cache`: Enables an external cache shared across all objectives.
- `-have-internal-cache`: Activates one internal cache per objective, allowing fine-grained reuse of intermediate computations.

These options are especially useful for complex or expensive objective functions, reducing redundant evaluations and improving efficiency.

Observers

Observers allow real-time monitoring of the optimization process. They can be activated using the `-observers` parameter, listing one or more observer names separated by semicolons. For example:

```
--observers BiclusterCountObserver;FitnessEvolutionAvgObserver
```

The available observers are:

- `BiclusterCountObserver`: Tracks the number of biclusters found in each solution over time.
- `FitnessEvolutionMinObserver`: Records the best (minimum) fitness values for each objective across generations.
- `FitnessEvolutionAvgObserver`: Logs the average fitness values of the population at each generation.
- `FitnessEvolutionMaxObserver`: Tracks the worst (maximum) fitness values for each objective.
- `NumEvaluationsObserver`: Keeps a count of the number of evaluations performed.
- `ExternalCacheObserver`: Monitors the usage of the external cache (number of hits and misses).
- `InternalCacheObserver`: Logs internal cache activity for each objective.

Each observer writes their data to a CSV file in the output directory, allowing post-analysis and visualization of the algorithm's behavior.

Parallel Execution and Output Files

The number of threads used during the execution can be controlled via the following option:

```
--num-threads 8
```

Output files are written to the directory specified with the `-output-folder` parameter. The output includes:

- `FUN.csv` and `VAR.csv`: Fitness values and solution vectors for the final population.
- `VAR-translated.csv`: Human-readable representation of the biclusters contained in each solution.
- CSV logs from each observer used during the execution.

Example Execution

Here is a complete example of how to launch `MOEBA-BIO` with a generic representation and two standard objectives:

```
java -cp target/moeba.jar moeba.RunnerMOEBA \
  --input-dataset data/expression.csv \
  --input-column-types data/column_types.json \
  --representation GENERIC \
  --generic-initial-min-num-bics 5 \
  --generic-initial-max-num-bics 20 \
  --str-fitness-functions BiclusterSizeNormComp;MeanSquaredResidueNorm \
  --summarise-individual-objectives HarmonicMean \
  --str-algorithm NSGAII-AsyncParallel \
  --crossover-operator RowBiclusterMixedCrossover;CellBinaryCrossover \
  --mutation-operator RowPermutationMutation;BiclusterBinaryMutation \
  --population-size 100 \
  --max-evaluations 50000 \
  --crossover-probability 0.9 \
  --mutation-probability 0.3->0.05 \
  --have-external-cache \
  --have-internal-cache \
  --observers BiclusterCountObserver;FitnessEvolutionMinObserver;
     NumEvaluationsObserver \
  --num-threads 8 \
  --output-folder results/example_run
```


Advanced Usage and Customization

The MOEBA-BIO framework supports advanced use cases such as:

- Integration of domain-specific objectives like RegulatoryCoherenceNormComp, designed for gene co-expression analysis.
- Fine-tuning of operator parameters to adapt the variation strategy to specific data characteristics.
- Exploration of future extensions such as the DYNAMIC representation, which combines the flexibility of the generic approach with the structure of specific encoding via co-evolving subpopulations.

Users are encouraged to combine these strategies creatively and to exploit cache-aware distributed optimization for scaling to larger biomedical datasets.

For updates, examples, and documentation, visit the official GitHub repository:

https://github.com/AdrianSeguraOrtiz/MOEBA-BIO

Bibliography

[1] Pau Badia-i-Mompel, Lorna Wessels, Sophia Müller-Dott, Rémi Trimbour, Ricardo O Ramirez Flores, Ricard Argelaguet, and Julio Saez-Rodriguez. "Gene regulatory network inference in the era of single-cell multi-omics". In: *Nature Reviews Genetics* 24.11 (2023), pp. 739–754.

[2] Zihao He, Kai Gao, Lei Dong, Liu Liu, Xinchi Qu, Zhengkai Zou, Yang Wu, Dechao Bu, Jin-Cheng Guo, and Yi Zhao. "Drug screening and biomarker gene investigation in cancer therapy through the human transcriptional regulatory network". In: *Computational and Structural Biotechnology Journal* 21 (2023), pp. 1557–1572.

[3] A. N. Burska, K. Roget, M. Blits, L. Soto Gomez, F. Van De Loo, L. D. Hazelwood, C. L. Verweij, A. Rowe, G. N. Goulielmos, L. G.M. Van Baarsen, and F. Ponchel. "Gene expression analysis in RA: towards personalized medicine". In: *The Pharmacogenomics Journal 2014 14:2* 14 (2 Mar. 2014), pp. 93–106. ISSN: 1473-1150. DOI: 10.1038/tpj.2013.48. URL: https://www.nature.com/articles/tpj201348.

[4] Monique G.P. Van Der Wijst, Dylan H. De Vries, Harm Brugge, Harm Jan Westra, and Lude Franke. "An integrative approach for building personalized gene regulatory networks for precision medicine". In: *Genome Medicine 2018 10:1* 10 (1 Dec. 2018), pp. 1–15. ISSN: 1756-994X. DOI: 10.1186/S13073-018-0608-4. URL: https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-018-0608-4.

[5] Yijuan Wang and Zhi-Ping Liu. "Identifying biomarkers for breast cancer by gene regulatory network rewiring". In: *BMC bioinformatics* 22.Suppl 12 (2022), p. 308.

[6] Geng Chen, Baitang Ning, and Tieliu Shi. "Single-cell RNA-seq technologies and related computational data analysis". In: *Frontiers in genetics* 10 (2019), p. 317.

[7] Yusuke Imoto, Tomonori Nakamura, Emerson G Escolar, Michio Yoshiwaki, Yoji Kojima, Yukihiro Yabuta, Yoshitaka Katou, Takuya Yamamoto, Yasuaki Hiraoka, and Mitinori Saitou. "Resolution of the curse of dimen-

sionality in single-cell RNA sequencing data analysis". In: *Life Science Alliance* 5.12 (2022).

[8] Chongzhi Zang, Tao Wang, Ke Deng, Bo Li, Sheng'en Hu, Qian Qin, Tengfei Xiao, Shihua Zhang, Clifford A Meyer, Housheng Hansen He, et al. "High-dimensional genomic data bias correction and data integration using MANCIE". In: *Nature communications* 7.1 (2016), p. 11305.

[9] Vân Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts. "Inferring regulatory networks from expression data using tree-based methods". In: *PloS one* 5.9 (2010), e12776.

[10] Richard Bonneau, David J Reiss, Paul Shannon, Marc Facciotti, Leroy Hood, Nitin S Baliga, and Vesteinn Thorsson. "The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo". In: *Genome biology* 7 (2006), pp. 1–16.

[11] Xiujun Zhang, Xing-Ming Zhao, Kun He, Le Lu, Yongwei Cao, Jingdong Liu, Jin-Kao Hao, Zhi-Ping Liu, and Luonan Chen. "Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information". In: *Bioinformatics* 28.1 (2012), pp. 98–104.

[12] TM Murali and Simon Kasif. "Extracting conserved gene expression motifs from gene expression data". In: *Biocomputing 2003*. World Scientific, 2002, pp. 77–88.

[13] Sven Bergmann, Jan Ihmels, and Naama Barkai. "Iterative signature algorithm for the analysis of large-scale gene expression data". In: *Physical review E* 67.3 (2003), p. 031902.

[14] Domingo S Rodriguez-Baena, Antonio J Perez-Pulido, and Jesus S Aguilar-Ruiz. "A biclustering algorithm for extracting bit-patterns from binary datasets". In: *Bioinformatics* 27.19 (2011), pp. 2738–2745.

[15] Daniel Marbach, James C. Costello, Robert Küffner, Nicole M. Vega, Robert J. Prill, and Diogo M. Camacho et al. "Wisdom of crowds for robust gene network inference". In: *Nature Methods 2012 9:8* 9 (8 July 2012), pp. 796–804. ISSN: 1548-7105. DOI: 10.1038/nmeth.2016. URL: https://www.nature.com/articles/nmeth.2016.

[16] Bingran Shen, Gloria Coruzzi, and Dennis Shasha. "EnsInfer: a simple ensemble approach to network inference outperforms any single method". In: *BMC bioinformatics* 24.1 (2023), p. 114.

[17] Mengyuan Zhao, Wenying He, Jijun Tang, Quan Zou, and Fei Guo. "A comprehensive overview and critical evaluation of gene regulatory network inference technologies". In: *Briefings in Bioinformatics* 22.5 (2021), bbab009.

[18] Y Tu, G Stolovitzky, and Ulf Klein. “Quantitative noise analysis for gene expression microarray experiments”. In: *Proceedings of the National Academy of Sciences* 99.22 (2002), pp. 14031–14036.

[19] Adán José-García, Julie Jacques, Vincent Sobanski, and Clarisse Dhaenens. “Metaheuristic Biclustering Algorithms: From State-of-the-Art to Future Opportunities”. In: *ACM Computing Surveys* 56.3 (2023), pp. 1–38.

[20] Adrián Segura-Ortiz, José García-Nieto, José F Aldana-Montes, and Ismael Navas-Delgado. “GENECI: a novel evolutionary machine learning consensus-based approach for the inference of gene regulatory networks”. In: *Computers in Biology and Medicine* 155 (2023), p. 106653.

[21] Pablo Meyer and Julio Saez-Rodriguez. “Advances in systems biology modeling: 10 years of crowdsourcing DREAM challenges”. In: *Cell Systems* 12.6 (2021), pp. 636–653.

[22] Irene Cantone, Lucia Marucci, Francesco Iorio, Maria Aurelia Ricci, Vincenzo Belcastro, Mukesh Bansal, Stefania Santini, Mario Di Bernardo, Diego Di Bernardo, and Maria Pia Cosma. “A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches”. In: *Cell* 137.1 (2009), pp. 172–181.

[23] Adrián Segura-Ortiz, José García-Nieto, and José F Aldana-Montes. “Exploiting medical-expert knowledge via a novel memetic algorithm for the inference of gene regulatory networks”. In: *International Conference on Computational Science*. Springer. 2024, pp. 3–17.

[24] Adrián Segura-Ortiz, José García-Nieto, José F Aldana-Montes, and Ismael Navas-Delgado. “Multi-objective context-guided consensus of a massive array of techniques for the inference of Gene Regulatory Networks”. In: *Computers in Biology and Medicine* 179 (2024), p. 108850.

[25] Adrián Segura-Ortiz, Karen Giménez-Orenga, José García-Nieto, Elisa Oltra, and José F. Aldana-Montes. “Multifaceted evolution focused on maximal exploitation of domain knowledge for the consensus inference of Gene Regulatory Networks”. In: *Computers in Biology and Medicine* 196 (2025), p. 110632. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2025.110632. URL: https://www.sciencedirect.com/science/article/pii/S0010482525009837.

[26] Adrián Segura-Ortiz, Adán José-García, Laetitia Jourdan, and José García-Nieto. “Exhaustive biclustering driven by self-learning evolutionary approach for biomedical data”. In: *Computer Methods and Programs in Biomedicine* (2025), p. 108846.

[27] George Orphanides and Danny Reinberg. “A unified theory of gene expression”. In: *Cell* 108.4 (2002), pp. 439–451.

[28] Eric Davidson and Michael Levin. “Gene regulatory networks”. In: *Proceedings of the National Academy of Sciences* 102.14 (2005), pp. 4935–4935. DOI: 10.1073/pnas.0502024102. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.0502024102. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0502024102.

[29] Joshua M Stuart, Eran Segal, Daphne Koller, and Stuart K Kim. “A gene-coexpression network for global discovery of conserved genetic modules”. In: *science* 302.5643 (2003), pp. 249–255.

[30] Shai S Shen-Orr, Robert Tibshirani, Purvesh Khatri, Dale L Bodian, Frank Staedtler, Nicholas M Perry, Trevor Hastie, Minnie M Sarwal, Mark M Davis, and Atul J Butte. “Cell type–specific gene expression differences in complex tissues”. In: *Nature methods* 7.4 (2010), pp. 287–289.

[31] Helen C Causton, Bing Ren, Sang Seok Koh, Christopher T Harbison, Elenita Kanin, Ezra G Jennings, Tong Ihn Lee, Heather L True, Eric S Lander, and Richard A Young. “Remodeling of yeast genome expression in response to environmental changes”. In: *Molecular biology of the cell* 12.2 (2001), pp. 323–337.

[32] Martin Fischer, Amy E Schade, Timothy B Branigan, Gerd A Müller, and James A DeCaprio. “Coordinating gene expression during the cell cycle”. In: *Trends in biochemical sciences* 47.12 (2022), pp. 1009–1022.

[33] Francis Crick. “Central dogma of molecular biology”. In: *Nature* 227.5258 (1970), pp. 561–563.

[34] David Whitford. *Proteins: structure and function*. John Wiley & Sons, 2013.

[35] Harley H McAdams and Adam Arkin. “Stochastic mechanisms in gene expression”. In: *Proceedings of the National Academy of Sciences* 94.3 (1997), pp. 814–819.

[36] David S Latchman. “Transcription factors: an overview”. In: *The international journal of biochemistry & cell biology* 29.12 (1997), pp. 1305–1312.

[37] Rossella Tupler, Giovanni Perini, and Michael R Green. “Expressing the human genome”. In: *Nature* 409.6822 (2001), pp. 832–833.

[38] ER Gibney and CM Nolan. “Epigenetics and gene expression”. In: *Heredity* 105.1 (2010), pp. 4–13.

[39] Marilena V Iorio, Manuela Ferracin, Chang-Gong Liu, Angelo Veronese, Riccardo Spizzo, Silvia Sabbioni, Eros Magri, Massimo Pedriali, Muller Fabbri, Manuela Campiglio, et al. “MicroRNA gene expression deregulation in human breast cancer”. In: *Cancer research* 65.16 (2005), pp. 7065–7070.

[40] Dinesh Singh, Phillip G Febbo, Kenneth Ross, Donald G Jackson, Judith Manola, Christine Ladd, Pablo Tamayo, Andrew A Renshaw, Anthony V D'Amico, Jerome P Richie, et al. "Gene expression correlates of clinical prostate cancer behavior". In: *Cancer cell* 1.2 (2002), pp. 203–209.

[41] Soroush Tahmasebi, Arkady Khoutorsky, Michael B Mathews, and Nahum Sonenberg. "Translation deregulation in human disease". In: *Nature Reviews Molecular Cell Biology* 19.12 (2018), pp. 791–807.

[42] Eric H Davidson. *The regulatory genome: gene regulatory networks in development and evolution*. Elsevier, 2010.

[43] Thomas Schlitt and Alvis Brazma. "Current approaches to gene regulatory network modelling". In: *BMC bioinformatics* 8 (2007), pp. 1–22.

[44] Réka Albert. "Scale-free networks in cell biology". In: *Journal of cell science* 118.21 (2005), pp. 4947–4957.

[45] Stefan Wuchty. "Scale-free behavior in protein domain networks". In: *Molecular biology and evolution* 18.9 (2001), pp. 1694–1702.

[46] Uri Alon. "Network motifs: theory and experimental approaches". In: *Nature Reviews Genetics* 8.6 (2007), pp. 450–461.

[47] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. "Network motifs: simple building blocks of complex networks". In: *Science* 298.5594 (2002), pp. 824–827.

[48] Robert J Prill, Pablo A Iglesias, and Andre Levchenko. "Dynamic properties of network motifs contribute to biological network organization". In: *PLoS biology* 3.11 (2005), e343.

[49] Michael Stock and Florian Otto. "Gene deregulation in gastric cancer". In: *Gene* 360.1 (2005), pp. 1–19.

[50] Vân Anh Huynh-Thu and Guido Sanguinetti. "Combining tree-based and dynamical systems for the inference of gene regulatory networks". In: *Bioinformatics* 31.10 (2015), pp. 1614–1622.

[51] Pawel Michalak. "Coexpression, coregulation, and cofunctionality of neighboring genes in eukaryotic genomes". In: *Genomics* 91.3 (2008), pp. 243–248.

[52] Steve Horvath and Jun Dong. "Geometric interpretation of gene coexpression network analysis". In: *PLoS computational biology* 4.8 (2008), e1000117.

[53] Liis Kolberg, Nurlan Kerimov, Hedi Peterson, and Kaur Alasoo. "Co-expression analysis reveals interpretable gene modules controlled by trans-acting genetic variants". In: *Elife* 9 (2020), e58705.

[54] Qi Liao, Changning Liu, Xiongying Yuan, Shuli Kang, Ruoyu Miao, Hui Xiao, Guoguang Zhao, Haitao Luo, Dechao Bu, Haitao Zhao, et al. "Large-scale prediction of long non-coding RNA functions in a coding–non-

coding gene co-expression network". In: *Nucleic acids research* 39.9 (2011), pp. 3864–3878.

[55] Feng Luo, Yunfeng Yang, Jianxin Zhong, Haichun Gao, Latifur Khan, Dorothea K Thompson, and Jizhong Zhou. "Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory". In: *BMC bioinformatics* 8 (2007), pp. 1–17.

[56] Sipko Van Dam, Urmo Vosa, Adriaan van der Graaf, Lude Franke, and Joao Pedro de Magalhaes. "Gene co-expression analysis for functional classification and gene–disease predictions". In: *Briefings in bioinformatics* 19.4 (2018), pp. 575–592.

[57] Beatriz Pontes, Raúl Giráldez, and Jesús S Aguilar-Ruiz. "Biclustering on expression data: A review". In: *Journal of biomedical informatics* 57 (2015), pp. 163–180.

[58] Sushmita Mitra and Haider Banka. "Multi-objective evolutionary biclustering of gene expression data". In: *Pattern Recognition* 39.12 (2006), pp. 2464–2477.

[59] Yanyun Zhang, Li Cheng, Guanyu Chen, and Daniyal Alghazzawi. "Evolutionary computation in bioinformatics: A survey". In: *Neurocomputing* 591 (2024), p. 127758.

[60] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[61] Aimo Törn and Antanas Žilinskas. *Global optimization*. Vol. 350. Springer, 1989.

[62] Peter Adby. *Introduction to optimization methods*. Springer Science & Business Media, 2013.

[63] Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*. John Wiley & Sons, 1999.

[64] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*. Vol. 6. Athena scientific Belmont, MA, 1997.

[65] Richard Bellman. "Dynamic programming". In: *science* 153.3731 (1966), pp. 34–37.

[66] Ailsa H Land and Alison G Doig. *An automatic method for solving discrete programming problems*. Springer, 2010.

[67] Zbigniew Michalewicz and David B Fogel. *How to solve it: modern heuristics*. Springer Science & Business Media, 2013.

[68] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

[69] Shen Lin and Brian W Kernighan. "An effective heuristic algorithm for the traveling-salesman problem". In: *Operations research* 21.2 (1973), pp. 498–516.

[70] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.
[71] Agoston E Eiben and James E Smith. *Introduction to evolutionary computing*. Springer, 2015.
[72] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. "Optimization by simulated annealing". In: *science* 220.4598 (1983), pp. 671–680.
[73] James Kennedy and Russell Eberhart. "Particle swarm optimization". In: *Proceedings of ICNN'95-international conference on neural networks*. Vol. 4. ieee. 1995, pp. 1942–1948.
[74] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization". In: *IEEE computational intelligence magazine* 1.4 (2007), pp. 28–39.
[75] Kalyanmoy Deb. "Multi-objective optimisation using evolutionary algorithms: an introduction". In: *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, 2011, pp. 3–34.
[76] Vilfredo Pareto. *Cours d'économie politique*. Vol. 1. Librairie Droz, 1964.
[77] Tobias Wagner, Nicola Beume, and Boris Naujoks. "Pareto-, aggregation-, and indicator-based methods in many-objective optimization". In: *International conference on evolutionary multi-criterion optimization*. Springer. 2007, pp. 742–756.
[78] Eckart Zitzler and Simon Künzli. "Indicator-based selection in multiobjective search". In: *International conference on parallel problem solving from nature*. Springer. 2004, pp. 832–842.
[79] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE transactions on evolutionary computation* 6.2 (2002), pp. 182–197.
[80] Kalyanmoy Deb and Himanshu Jain. "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints". In: *IEEE transactions on evolutionary computation* 18.4 (2013), pp. 577–601.
[81] Qingfu Zhang and Hui Li. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition". In: *IEEE Transactions on evolutionary computation* 11.6 (2007), pp. 712–731.
[82] Antonio J Nebro, Juan J Durillo, Francisco Luna, Bernabé Dorronsoro, and Enrique Alba. "Mocell: A cellular genetic algorithm for multiobjective optimization". In: *International Journal of Intelligent Systems* 24.7 (2009), pp. 726–746.
[83] Marco Laumanns. "SPEA2: Improving the strength Pareto evolutionary algorithm". In: *Technical Report, Gloriastrasse 35* (2001).

[84] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. "Performance assessment of multiobjective optimizers: An analysis and review". In: *IEEE Transactions on evolutionary computation* 7.2 (2003), pp. 117–132.

[85] David A Van Veldhuizen and Gary B Lamont. *Multiobjective evolutionary algorithm research: A history and analysis*. Tech. rep. Citeseer, 1998.

[86] Eckart Zitzler and Lothar Thiele. "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach". In: *IEEE transactions on Evolutionary Computation* 3.4 (1999), pp. 257–271.

[87] Hisao Ishibuchi, Hiroyuki Masuda, and Yusuke Nojima. "A study on performance evaluation ability of a modified inverted generational distance indicator". In: *Proceedings of the 2015 annual conference on genetic and evolutionary computation*. 2015, pp. 695–702.

[88] Thomas G Dietterich. "Ensemble methods in machine learning". In: *International workshop on multiple classifier systems*. Springer. 2000, pp. 1–15.

[89] Arjun Chandra and Xin Yao. "Ensemble learning using multi-objective evolutionary algorithms". In: *Journal of Mathematical Modelling and Algorithms* 5 (2006), pp. 417–445.

[90] Michael Hecker, Sandro Lambeck, Susanne Toepfer, Eugene Van Someren, and Reinhard Guthke. "Gene regulatory network inference: data integration in dynamic models—a review". In: *Biosystems* 96.1 (2009), pp. 86–103.

[91] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. "Using Bayesian networks to analyze expression data". In: *Proceedings of the fourth annual international conference on Computational molecular biology*. 2000, pp. 127–135.

[92] John Quackenbush. "Microarray data normalization and transformation". In: *Nature genetics* 32.4 (2002), pp. 496–501.

[93] Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, Daniel J Gaffney, Laura L Elo, Xuegong Zhang, et al. "A survey of best practices for RNA-seq data analysis". In: *Genome biology* 17 (2016), pp. 1–19.

[94] Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner. "Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles". In: *PLoS biology* 5.1 (2007), e8.

[95] Xiujun Zhang, Juan Zhao, Jin-Kao Hao, Xing-Ming Zhao, and Luonan Chen. "Conditional mutual inclusive information enables accurate quan-

tification of associations in gene regulatory networks". In: *Nucleic Acids Research* 43.5 (Dec. 2014), e31–e31. ISSN: 0305-1048. DOI: 10.1093/nar/gku1315. eprint: https://academic.oup.com/nar/article-pdf/43/5/e31/16864438/gku1315.pdf. URL: https://doi.org/10.1093/nar/gku1315.

[96] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. "ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context". In: *BMC bioinformatics*. Vol. 7. Springer. 2006, pp. 1–15.

[97] Anne-Claire Haury, Fantine Mordelet, Paola Vera-Licona, and Jean-Philippe Vert. "TIGRESS: trustful inference of gene regulation using stability selection". In: *BMC systems biology* 6.1 (2012), pp. 1–17.

[98] Baoshan Ma, Mingkun Fang, and Xiangtian Jiao. "Inference of gene regulatory networks based on nonlinear ordinary differential equations". In: *Bioinformatics* 36.19 (2020), pp. 4885–4893.

[99] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[100] John A Hartigan. "Direct clustering of a data matrix". In: *Journal of the american statistical association* 67.337 (1972), pp. 123–129.

[101] Yizong Cheng and George M Church. "Biclustering of expression data." In: *Ismb*. Vol. 8. 2000, pp. 93–103.

[102] Sara C Madeira and Arlindo L Oliveira. "Biclustering algorithms for biological data analysis: a survey". In: *IEEE/ACM transactions on computational biology and bioinformatics* 1.1 (2004), pp. 24–45.

[103] Anirban Mukhopadhyay, Ujjwal Maulik, and Sanghamitra Bandyopadhyay. "A novel coherence measure for discovering scaling biclusters from gene expression data". In: *Journal of bioinformatics and computational biology* 7.05 (2009), pp. 853–868.

[104] Xiaowen Liu and Lusheng Wang. "Computing the maximum similarity biclusters of gene expression data". In: *Bioinformatics* 23.1 (2007), pp. 50–56.

[105] Jiong Yang, Haixun Wang, Wei Wang, and Philip S Yu. "An improved biclustering method for analyzing gene expression profiles". In: *International Journal on Artificial Intelligence Tools* 14.05 (2005), pp. 771–789.

[106] Fabrizio Angiulli, Eugenio Cesario, and Clara Pizzuti. "Random walk biclustering for microarray data". In: *Information Sciences* 178.6 (2008), pp. 1479–1497.

[107] Smitha Dharan and Achuthsankar S Nair. “Biclustering of gene expression data using reactive greedy randomized adaptive search procedure”. In: *BMC bioinformatics* 10 (2009), pp. 1–10.

[108] Kenneth Bryan, Pádraig Cunningham, and Nadia Bolshakova. “Application of simulated annealing to the biclustering of gene expression data”. In: *IEEE transactions on information technology in biomedicine* 10.3 (2006), pp. 519–525.

[109] Junwan Liu, Zhoujun Li, Xiaohua Hu, and Yiming Chen. “Biclustering of microarray data with MOSPO based on crowding distance”. In: *BMC bioinformatics*. Vol. 10. Springer. 2009, pp. 1–10.

[110] Guilherme Palermo Coelho, Fabrício Olivetti de França, and Fernando J Von Zuben. “Multi-objective biclustering: When non-dominated solutions are not enough”. In: *Journal of Mathematical Modelling and Algorithms* 8 (2009), pp. 175–202.

[111] Carlos Cano, L Adarve, J López, and Armando Blanco. “Possibilistic approach for biclustering microarray data”. In: *Computers in biology and medicine* 37.10 (2007), pp. 1426–1436.

[112] Wen-Hui Yang, Dao-Qing Dai, and Hong Yan. “Finding correlated biclusters from gene expression data”. In: *IEEE Transactions on Knowledge and Data Engineering* 23.4 (2010), pp. 568–584.

[113] Heather Turner, Trevor Bailey, and Wojtek Krzanowski. “Improved biclustering of microarray data demonstrated through systematic performance tests”. In: *Computational statistics & data analysis* 48.2 (2005), pp. 235–254.

[114] Eran Segal, Ben Taskar, Audrey Gasch, Nir Friedman, and Daphne Koller. “Rich probabilistic models for gene expression”. In: *BIOINFORMATICS-OXFORD-* 17 (2001), S243–S252.

[115] Amos Tanay, Roded Sharan, and Ron Shamir. “Discovering statistically significant biclusters in gene expression data”. In: *Bioinformatics* 18.suppl_1 (2002), S136–S144.

[116] Guojun Li, Qin Ma, Haibao Tang, Andrew H Paterson, and Ying Xu. “QUBIC: a qualitative biclustering algorithm for analyses of gene expression data”. In: *Nucleic acids research* 37.15 (2009), e101–e101.

[117] Lizhuang Zhao and Mohammed J Zaki. “MicroCluster: efficient deterministic biclustering of microarray data”. In: *IEEE intelligent systems* 20.6 (2005), pp. 40–49.

[118] Yuval Kluger, Ronen Basri, Joseph T Chang, and Mark Gerstein. “Spectral biclustering of microarray data: coclustering genes and conditions”. In: *Genome research* 13.4 (2003), pp. 703–716.

[119] Pedro Carmona-Saez, Roberto D Pascual-Marqui, Francisco Tirado, Jose M Carazo, and Alberto Pascual-Montano. “Biclustering of gene expression data by non-smooth non-negative matrix factorization”. In: *BMC bioinformatics* 7 (2006), pp. 1–18.

[120] Amir Ben-Dor, Benny Chor, Richard Karp, and Zohar Yakhini. “Discovering local structure in gene expression data: the order-preserving submatrix problem”. In: *Proceedings of the sixth annual international conference on Computational biology*. 2002, pp. 49–57.

[121] Peter A DiMaggio, Scott R McAllister, Christodoulos A Floudas, Xiao-Jiang Feng, Joshua D Rabinowitz, and Herschel A Rabitz. “Biclustering via optimal re-ordering of data matrices in systems biology: rigorous methods and comparative studies”. In: *BMC bioinformatics* 9 (2008), pp. 1–16.

[122] Rui Henriques, Claudia Antunes, and Sara C Madeira. “A structured view on pattern mining-based biclustering”. In: *Pattern Recognition* 48.12 (2015), pp. 3941–3958.

[123] Victor A Padilha and Ricardo J G B Campello. “A systematic comparative evaluation of biclustering techniques”. In: *BMC Bioinformatics* 18.1 (2017), p. 55.

[124] Amela Prelić, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele, and Eckart Zitzler. “A systematic comparison and evaluation of biclustering methods for gene expression data”. In: *Bioinformatics* 22.9 (2006), pp. 1122–1129.

[125] Andrey A Shabalin, Victor J Weigman, Charles M Perou, and Andrew B Nobel. “Finding large average submatrices in high dimensional data”. In: *The Annals of Applied Statistics* (2009), pp. 985–1012.

[126] Ricardo de Matos Simoes and Frank Emmert-Streib. “Bagging statistical network inference from large-scale gene expression data”. In: *PloS one* 7.3 (2012), e33624.

[127] Gökmen Altay and Frank Emmert-Streib. “Inferring the conservative causal core of gene regulatory networks”. In: *BMC systems biology* 4.1 (2010), pp. 1–13.

[128] Solomon Kullback and Richard A Leibler. “On information and sufficiency”. In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[129] Thomas Moerman, Sara Aibar Santos, Carmen Bravo González-Blas, Jaak Simm, Yves Moreau, Jan Aerts, and Stein Aerts. “GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks”. In: *Bioinformatics* 35.12 (Nov. 2018), pp. 2159–2161. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty916. eprint: https://academic.

oup.com/bioinformatics/article-pdf/35/12/2159/48934650/bioinformatics_35_12_2159.pdf. URL: https://doi.org/10.1093/bioinformatics/bty916.

[130] Matthew Rocklin et al. "Dask: Parallel computation with blocked algorithms and task scheduling." In: *SciPy*. 2015, pp. 126–132.

[131] M Sanchez-Castillo, D Blanco, I M Tienda-Luna, M C Carrion, and Yufei Huang. "A Bayesian framework for the inference of gene regulatory networks from time and pseudo-time series data". In: *Bioinformatics* 34.6 (Sept. 2017), pp. 964–970. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx605. eprint: https://academic.oup.com/bioinformatics/article-pdf/34/6/964/48914307/bioinformatics_34_6_964.pdf. URL: https://doi.org/10.1093/bioinformatics/btx605.

[132] Luis F Iglesias-Martinez, Barbara De Kegel, and Walter Kolch. "KBoost: a new method to infer gene regulatory networks from gene expression data". In: *Scientific Reports* 11.1 (2021), p. 15461.

[133] Alicia T Specht and Jun Li. "LEAP: constructing gene co-expression networks for single-cell RNA-sequencing data using pseudotime ordering". In: *Bioinformatics* 33.5 (2017), pp. 764–766.

[134] Xiang Chen, Min Li, Ruiqing Zheng, Siyu Zhao, Fang-Xiang Wu, Yaohang Li, and Jianxin Wang. "A novel method of gene regulatory network structure inference from gene knock-out expression data". In: *Tsinghua Science and Technology* 24.4 (2019), pp. 446–455. DOI: 10.26599/TST.2018.9010097.

[135] Jimeng Lei, Zongheng Cai, Xinyi He, Wanting Zheng, and Jianxiao Liu. "An approach of gene regulatory network construction using mixed entropy optimizing context-related likelihood mutual information". In: *Bioinformatics* 39.1 (2023), btac717.

[136] Jean Hausser and Korbinian Strimmer. "Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks." In: *Journal of Machine Learning Research* 10.7 (2009).

[137] Thomas Minka. *Estimating a Dirichlet distribution*. 2000.

[138] Patrick E Meyer, Kevin Kontos, Frederic Lafitte, and Gianluca Bontempi. "Information-theoretic inference of large transcriptional regulatory networks". In: *EURASIP journal on bioinformatics and systems biology* 2007 (2007), pp. 1–9.

[139] Patrick Meyer, Daniel Marbach, Sushmita Roy, and Manolis Kellis. "Information-Theoretic Inference of Gene Networks Using Backward Elimination." In: *BioComp*. 2010, pp. 700–705.

[140] Xiujun Zhang, Keqin Liu, Zhi-Ping Liu, Béatrice Duval, Jean-Michel Richer, Xing-Ming Zhao, Jin-Kao Hao, and Luonan Chen. "NARROMI: a noise

and redundancy reduction technique improves accuracy of gene regulatory network inference". In: *Bioinformatics* 29.1 (2013), pp. 106–113.

[141] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[142] Pablo Meyer and Julio Saez-Rodriguez. "Advances in systems biology modeling: 10 years of crowdsourcing DREAM challenges". In: *Cell Systems* 12.6 (2021), pp. 636–653. ISSN: 2405-4712. DOI: https://doi.org/10.1016/j.cels.2021.05.015. URL: https://www.sciencedirect.com/science/article/pii/S2405471221002015.

[143] Antonio Reverter and Eva KF Chan. "Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks". In: *Bioinformatics* 24.21 (2008), pp. 2491–2497.

[144] Thalia E. Chan, Michael P.H. Stumpf, and Ann C. Babtie. "Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures". In: *Cell Systems* 5.3 (2017), 251–267.e3. ISSN: 2405-4712. DOI: https://doi.org/10.1016/j.cels.2017.08.014. URL: https://www.sciencedirect.com/science/article/pii/S2405471217303861.

[145] Shun Guo, Qingshan Jiang, Lifei Chen, and Donghui Guo. "Gene regulatory network inference using PLS-based methods". In: *BMC bioinformatics* 17.1 (2016), pp. 1–10.

[146] Xiaohan Jiang and Xiujun Zhang. "RSNET: inferring gene regulatory networks by a redundancy silencing and network enhancement technique". In: *BMC bioinformatics* 23.1 (2022), pp. 1–18.

[147] Duaa Mohammad Alawad, Ataur Katebi, Md Wasi Ul Kabir, and Md Tamjidul Hoque. "AGRN: accurate gene regulatory network inference using ensemble machine learning methods". In: *Bioinformatics Advances* 3.1 (2023), vbad032.

[148] Maneesha Aluru, Harsh Shrivastava, Sriram P Chockalingam, Shruti Shivakumar, and Srinivas Aluru. "EnGRaiN: a supervised ensemble learning method for recovery of large-scale gene regulatory networks". In: *Bioinformatics* 38.5 (2022), pp. 1312–1319.

[149] Chisato Fujii, Hiroyuki Kuwahara, Ge Yu, Lili Guo, and Xin Gao. "Learning gene regulatory networks from gene expression data using weighted consensus". In: *Neurocomputing* 220 (2017), pp. 23–33.

[150] Sergio Peignier, Baptiste Sorin, and Federica Calevro. "Ensemble Learning Based Gene Regulatory Network Inference". In: *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. 2021, pp. 113–120. DOI: 10.1109/ICTAI52525.2021.00024.

[151] Pauline Schmitt, Baptiste Sorin, Timothée Frouté, Nicolas Parisot, Federica Calevro, and Sergio Peignier. "GReNaDIne: A Data-Driven Python Library to Infer Gene Regulatory Networks from Gene Expression Data". In: *Genes* 14.2 (2023). ISSN: 2073-4425. DOI: 10.3390/genes14020269. URL: https://www.mdpi.com/2073-4425/14/2/269.

[152] Carlo Bonferroni. "Teoria statistica delle classi e calcolo delle probabilita". In: *Pubblicazioni del R istituto superiore di scienze economiche e commericiali di firenze* 8 (1936), pp. 3–62.

[153] Eduardo N Castanho, Helena Aidos, and Sara C Madeira. "Biclustering data analysis: a comprehensive survey". In: *Briefings in Bioinformatics* 25.4 (2024).

[154] Adán José-García, Julie Jacques, Vincent Sobanski, and Clarisse Dhaenens. "Biclustering algorithms based on metaheuristics: a review". In: *Metaheuristics for machine learning: new advances and tools* (2022), pp. 39–71.

[155] Zhoufan Kong, Qinghua Huang, and Xuelong Li. "Bi-Phase evolutionary biclustering algorithm with the NSGA-II algorithm". In: *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE. 2019, pp. 146–149.

[156] Sudipta Acharya, Sriparna Saha, and Pracheta Sahoo. "Bi-clustering of microarray data using a symmetry-based multi-objective optimization framework". In: *Soft Computing* 23 (2019), pp. 5693–5714.

[157] Marta DM Noronha, Rui Henriques, Sara C Madeira, and Luis E Zárate. "Impact of metrics on biclustering solution and quality: A review". In: *Pattern Recognition* 127 (2022), p. 108612.

[158] Federico Divina, Francisco A Goméz Vela, and Miguel García Torres. "Biclustering of smart building electric energy consumption data". In: *Applied Sciences* 9.2 (2019), p. 222.

[159] Maryam Golchin, Seyed Hashem Davarpanah, and Alan Wee-Chung Liew. "Biclustering analysis of gene expression data using multi-objective evolutionary algorithms". In: *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*. Vol. 2. IEEE. 2015, pp. 505–510.

[160] Ons Maatouk, Emna Ayari, Hend Bouziri, and Wassim Ayadi. "Bobea: a bi-objective biclustering evolutionary algorithm for genome-wide association analysis". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 2022, pp. 344–347.

[161] Pracheta Sahoo, Sudipta Acharya, and Sriparna Saha. "Automatic generation of biclusters from gene expression data using multi-objective simulated annealing approach". In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE. 2016, pp. 2174–2179.

[162] Ons Maâtouk, Wassim Ayadi, Hend Bouziri, and Béatrice Duval. "Evolutionary biclustering algorithms: an experimental study on microarray data". In: *Soft Computing* 23 (2019), pp. 7671–7697.

[163] Khedidja Seridi, Laetitia Jourdan, and El-Ghazali Talbi. "Multi-objective evolutionary algorithm for biclustering in microarrays data". In: *2011 IEEE congress of evolutionary computation (CEC)*. IEEE. 2011, pp. 2593–2599.

[164] Beatriz Pontes, Raúl Giráldez, and Jesús S Aguilar-Ruiz. "Configurable pattern-based evolutionary biclustering of gene expression data". In: *Algorithms for Molecular Biology* 8 (2013), pp. 1–22.

[165] Anirban Mukhopadhyay, Ujjwal Maulik, and Sanghamitra Bandyopadhyay. "A novel coherence measure for discovering scaling biclusters from gene expression data". In: *Journal of bioinformatics and computational biology* 7.05 (2009), pp. 853–868.

[166] Juan A Nepomuceno, Alicia Troncos, and Jesús S Aguilar-Ruiz. "Evolutionary metaheuristic for biclustering based on linear correlations among genes". In: *Proceedings of the 2010 ACM Symposium on Applied Computing*. 2010, pp. 1143–1147.

[167] Li Teng and Laiwan Chan. "Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data". In: *Journal of Signal Processing Systems* 50 (2008), pp. 267–280.

[168] Federico Divina, Beatriz Pontes, Raúl Giráldez, and Jesús S Aguilar-Ruiz. "An effective measure for assessing the quality of biclusters". In: *Computers in biology and medicine* 42.2 (2012), pp. 245–256.

[169] Ons Maâtouk, Wassim Ayadi, Hend Bouziri, and Beatrice Duval. "Evolutionary algorithm based on new crossover for the biclustering of gene expression data". In: *Pattern Recognition in Bioinformatics: 9th IAPR International Conference, PRIB 2014, Stockholm, Sweden, August 21-23, 2014. Proceedings 9*. Springer. 2014, pp. 48–59.

[170] Meriem Bousselmi, Slim Bechikh, Chih-Cheng Hung, and Lamjed Ben Said. "Bi-MOCK: A multi-objective evolutionary algorithm for bi-clustering with automatic determination of the number of bi-clusters". In: *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14–18, 2017, Proceedings, Part IV 24*. Springer. 2017, pp. 366–376.

[171] Maryam Golchin and Alan Wee Chung Liew. "Parallel biclustering detection using strength Pareto front evolutionary algorithm". In: *Information Sciences* 415 (2017), pp. 283–297.

[172] Naveen Saini, Sriparna Saha, Chirag Soni, and Pushpak Bhattacharyya. "Automatic evolution of bi-clusters from microarray data using self-organized

multi-objective evolutionary algorithm". In: *Applied Intelligence* 50 (2020), pp. 1027–1044.

[173] Irene Cantone, Lucia Marucci, Francesco Iorio, Maria Aurelia Ricci, Vincenzo Belcastro, Mukesh Bansal, Stefania Santini, Mario di Bernardo, Diego di Bernardo, and Maria Pia Cosma. "A Yeast Synthetic Network for In Vivo Assessment of Reverse-Engineering and Modeling Approaches". In: *Cell* 137.1 (2009), pp. 172–181. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2009.01.055. URL: https://www.sciencedirect.com/science/article/pii/S0092867409001561.

[174] Orsolya Liska, Balázs Bohár, András Hidas, Tamás Korcsmáros, Balázs Papp, Dávid Fazekas, and Eszter Ari. "TFLink: an integrated gateway to access transcription factor–target gene interactions for multiple species". In: *Database* 2022 (Sept. 2022). baac083. ISSN: 1758-0463. DOI: 10.1093/database/baac083. eprint: https://academic.oup.com/database/article-pdf/doi/10.1093/database/baac083/45888472/baac083.pdf. URL: https://doi.org/10.1093/database/baac083.

[175] Víctor H Tierrafría, Claire Rioualen, Heladia Salgado, Paloma Lara, Socorro Gama-Castro, Patrick Lally, Laura Gómez-Romero, Pablo Peña-Loredo, Andrés G López-Almazo, Gabriel Alarcón-Carranza, et al. "RegulonDB 11.0: Comprehensive high-throughput datasets on transcriptional regulation in Escherichia coli K-12". In: *Microbial Genomics* 8.5 (2022), p. 000833.

[176] Zhi-Ping Liu, Canglin Wu, Hongyu Miao, and Hulin Wu. "RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse". In: *Database* 2015 (2015).

[177] Rose Oughtred, Jennifer Rust, Christie Chang, Bobby-Joe Breitkreutz, Chris Stark, Andrew Willems, Lorrie Boucher, Genie Leung, Nadine Kolas, Frederick Zhang, et al. "The BioGRID database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions". In: *Protein Science* 30.1 (2021), pp. 187–200.

[178] Li Fang, Yunjin Li, Lu Ma, Qiyue Xu, Fei Tan, and Geng Chen. "GRNdb: decoding the gene regulatory networks in diverse human and mouse conditions". In: *Nucleic acids research* 49.D1 (2021), pp. D97–D103.

[179] Andrea Pinna, Nicola Soranzo, Ina Hoeschele, and Alberto de la Fuente. "Simulating systems genetics data with SysGenSIM". In: *Bioinformatics* 27.17 (2011), pp. 2459–2462.

[180] Tim Van den Bulcke, Koenraad Van Leemput, Bart Naudts, Piet van Remortel, Hongwu Ma, Alain Verschoren, Bart De Moor, and Kathleen Marchal. "SynTReN: a generator of synthetic gene expression data for design

and analysis of structure learning algorithms". In: *BMC bioinformatics* 7 (2006), pp. 1–12.

[181] Simon Rogers and Mark Girolami. "A Bayesian regression approach to the inference of regulatory networks from gene expression data". In: *Bioinformatics* 21.14 (May 2005), pp. 3131–3137. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti487. eprint: https://academic.oup.com/bioinformatics/article-pdf/21/14/3131/48971535/bioinformatics_21_14_3131.pdf. URL: https://doi.org/10.1093/bioinformatics/bti487.

[182] Thomas Schaffter, Daniel Marbach, and Dario Floreano. "GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods". In: *Bioinformatics* 27.16 (June 2011), pp. 2263–2270. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btr373. eprint: https://academic.oup.com/bioinformatics/article-pdf/27/16/2263/48863257/bioinformatics_27_16_2263.pdf. URL: https://doi.org/10.1093/bioinformatics/btr373.

[183] Robert J Prill, Daniel Marbach, Julio Saez-Rodriguez, Peter K Sorger, Leonidas G Alexopoulos, Xiaowei Xue, Neil D Clarke, Gregoire Altan-Bonnet, and Gustavo Stolovitzky. "Towards a rigorous assessment of systems biology models: the DREAM3 challenges". In: *PloS one* 5.2 (2010), e9202.

[184] P Bellot, C Olsen, and PE Meyer. "Grndata: synthetic expression data for gene regulatory network inference". In: *R package version* 1.0 (2018).

[185] Luz Garcia-Alonso, Christian H. Holland, Mahmoud M. Ibrahim, Denes Turei, and Julio Saez-Rodriguez. "Benchmark and integration of resources for the estimation of human transcription factor activities". In: *Genome Research* (2019). DOI: 10.1101/gr.240663.118.

[186] Semyon Kolmykov, Ivan Yevshin, Mikhail Kulyashov, Ruslan Sharipov, Yury Kondrakhin, Vsevolod J Makeev, Ivan V Kulakovskiy, Alexander Kel, and Fedor Kolpakov. "GTRD: an integrated view of transcription regulation". In: *Nucleic Acids Research* 49.D1 (Nov. 2020), pp. D104–D111. ISSN: 0305-1048. DOI: 10.1093/nar/gkaa1057. eprint: https://academic.oup.com/nar/article-pdf/49/D1/D104/35364856/gkaa1057.pdf. URL: https://doi.org/10.1093/nar/gkaa1057.

[187] Luiz Bovolenta, Marcio Acencio, and Ney Lemke. "HTRIdb: an open-access database for experimentally verified human transcriptional regulation interactions". In: *Nature Precedings* (2012), pp. 1–1.

[188] Jaime A Castro-Mondragon, Rafael Riudavets-Puig, Ieva Rauluseviciute, Roza Berhanu Lemma, Laura Turchi, Romain Blanc-Mathieu, Jeremy Lucas, Paul Boddie, Aziz Khan, Nicolás Manosalva Pérez, Oriol Fornes,

Tiffany Y Leung, Alejandro Aguirre, Fayrouz Hammal, Daniel Schmelter, Damir Baranasic, Benoit Ballester, Albin Sandelin, Boris Lenhard, Klaas Vandepoele, Wyeth W Wasserman, François Parcy, and Anthony Mathelier. "JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles". In: *Nucleic Acids Research* 50.D1 (Nov. 2021), pp. D165–D173. ISSN: 0305-1048. DOI: 10.1093/nar/gkab1113. eprint: https://academic.oup.com/nar/article-pdf/50/D1/D165/42058061/gkab1113.pdf. URL: https://doi.org/10.1093/nar/gkab1113.

[189] Robert Lesurf, Kelsy C. Cotto, Grace Wang, Malachi Griffith, Katayoon Kasaian, Steven J. M. Jones, Stephen B. Montgomery, Obi L. Griffith, and The Open Regulatory Annotation Consortium. "ORegAnno 3.0: a community-driven resource for curated regulatory annotation". In: *Nucleic Acids Research* 44.D1 (Nov. 2015), pp. D126–D132. ISSN: 0305-1048. DOI: 10.1093/nar/gkv1203. eprint: https://academic.oup.com/nar/article-pdf/44/D1/D126/16661622/gkv1203.pdf. URL: https://doi.org/10.1093/nar/gkv1203.

[190] Soile VE Keränen, Angel Villahoz-Baleta, Andrew E Bruno, and Marc S Halfon. "REDfly: An Integrated Knowledgebase for Insect Regulatory Genomics". In: *Insects* 13.7 (2022), p. 618.

[191] Fayrouz Hammal, Pierre de Langen, Aurélie Bergon, Fabrice Lopez, and Benoit Ballester. "ReMap 2022: a database of human, mouse, drosophila and arabidopsis regulatory regions from an integrative analysis of DNA-binding sequencing experiments". In: *Nucleic Acids Research* 50.D1 (2022), pp. D316–D325.

[192] C Jiang, Zhenyu Xuan, Fang Zhao, and Michael Q Zhang. "TRED: a transcriptional regulatory element database, new entries and other development". In: *Nucleic acids research* 35.suppl_1 (2007), pp. D137–D140.

[193] Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al. "TRRUST v2: an expanded reference database of human and mouse transcriptional regulatory interactions". In: *Nucleic acids research* 46.D1 (2018), pp. D380–D386.

[194] Miguel Cacho Teixeira, Romeu Viana, Margarida Palma, Jorge Oliveira, Mónica Galocha, Marta Neves Mota, Diogo Couceiro, Maria Galhardas Pereira, Miguel Antunes, Inês V Costa, Pedro Pais, Carolina Parada, Claudine Chaouiya, Isabel Sá-Correia, and Pedro Tiago Monteiro. "YEASTRACT+: a portal for the exploitation of global transcription regulation and metabolic model data in yeast biotechnology and pathogenesis". In: *Nucleic Acids Research* 51.D1 (Nov. 2022), pp. D785–D791. ISSN: 0305-

1048. DOI: 10.1093/nar/gkac1041. eprint: https://academic.oup.com/nar/article-pdf/51/D1/D785/48440926/gkac1041.pdf. URL: https://doi.org/10.1093/nar/gkac1041.

[195] Nabil Guelzim, Samuele Bottani, Paul Bourgine, and François Képès. "Topological and causal structure of the yeast transcriptional regulatory network". In: *Nature genetics* 31.1 (2002), pp. 60–63.

[196] Ismael Navas-Delgado, José García-Nieto, Esteban López-Camacho, Maciej Rybinski, Rocio Lavado, Miguel Ángel Berciano Guerrero, and José F Aldana-Montes. "VIGLA-M: visual gene expression data analytics". In: *BMC bioinformatics* 20 (2019), pp. 1–11.

[197] Gary K Geiss, Roger E Bumgarner, Brian Birditt, Timothy Dahl, Naeem Dowidar, Dwayne L Dunaway, H Perry Fell, Sean Ferree, Renee D George, Tammy Grogan, et al. "Direct multiplexed measurement of gene expression with color-coded probe pairs". In: *Nature biotechnology* 26.3 (2008), pp. 317–325.

[198] Hong Wang, Craig Horbinski, Hao Wu, Yinxing Liu, Shaoyi Sheng, Jinpeng Liu, Heidi Weiss, Arnold J Stromberg, and Chi Wang. "NanoStringDiff: a novel statistical method for differential expression analysis based on NanoString nCounter data". In: *Nucleic acids research* 44.20 (2016), e151–e151.

[199] Daryl Waggott, Kenneth Chu, Shaoming Yin, Bradly G Wouters, Fei-Fei Liu, and Paul C Boutros. "NanoStringNorm: an extensible R package for the pre-processing of NanoString mRNA and miRNA data". In: *Bioinformatics* 28.11 (2012), pp. 1546–1548.

[200] Sandro Hurtado, Jose Garcia-Nieto, Ismael Navas-Delgado, Antonio J Nebro, and Jose F Aldana-Montes. "Reconstruction of gene regulatory networks with multi-objective particle swarm optimisers". In: *Applied Intelligence* 51.4 (2021), pp. 1972–1991.

[201] Bruce M Carruthers, Anil Kumar Jain, Kenny L De Meirleir, Daniel L Peterson, Nancy G Klimas, A Martin Lerner, Alison C Bested, Pierre Flor-Henry, Pradip Joshi, AC Peter Powles, et al. "Myalgic encephalomyelitis/chronic fatigue syndrome: clinical working case definition, diagnostic and treatment protocols". In: *Journal of chronic fatigue syndrome* 11.1 (2003), pp. 7–115.

[202] Bruce M Carruthers, Marjorie I van de Sande, Kenny L De Meirleir, Nancy G Klimas, Gordon Broderick, Terry Mitchell, Don Staines, AC Peter Powles, Nigel Speight, Rosamund Vallings, et al. "Myalgic encephalomyelitis: international consensus criteria". In: *Journal of internal medicine* 270.4 (2011), pp. 327–338.

[203] Frederick Wolfe, Hugh A Smythe, Muhammad B Yunus, Robert M Bennett, Claire Bombardier, Don L Goldenberg, Peter Tugwell, Stephen M Campbell, Micha Abeles, Patricia Clark, et al. "The American College of Rheumatology 1990 criteria for the classification of fibromyalgia". In: *Arthritis & Rheumatism: Official Journal of the American College of Rheumatology* 33.2 (1990), pp. 160–172.

[204] Frederick Wolfe, Daniel J Clauw, Mary-Ann Fitzcharles, Don L Goldenberg, Robert S Katz, Philip Mease, Anthony S Russell, I Jon Russell, John B Winfield, and Muhammad B Yunus. "The American College of Rheumatology preliminary diagnostic criteria for fibromyalgia and measurement of symptom severity". In: *Arthritis care & research* 62.5 (2010), pp. 600–610.

[205] Jérémie Becker, Philippe Pérot, Valérie Cheynet, Guy Oriol, Nathalie Mugnier, Marine Mommert, Olivier Tabone, Julien Textoris, Jean-Baptiste Veyrieras, and François Mallet. "A comprehensive hybridization model allows whole HERV transcriptome profiling using high density microarray". In: *Bmc Genomics* 18 (2017), pp. 1–14.

[206] Benilton S Carvalho and Rafael A Irizarry. "A framework for oligonucleotide microarray preprocessing". In: *Bioinformatics* 26.19 (2010), pp. 2363–2367.

[207] Matthew E Ritchie, Belinda Phipson, DI Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. "limma powers differential expression analyses for RNA-sequencing and microarray studies". In: *Nucleic acids research* 43.7 (2015), e47–e47.

[208] Eduardo N Castanho, João P Lobo, Rui Henriques, and Sara C Madeira. "G-bic: generating synthetic benchmarks for biclustering". In: *BMC bioinformatics* 24.1 (2023), p. 457.

[209] Sepp Hochreiter, Ulrich Bodenhofer, Martin Heusel, Andreas Mayr, Andreas Mitterecker, Adetayo Kasim, Tatsiana Khamiakova, Suzy Van Sanden, Dan Lin, Willem Talloen, et al. "FABIA: factor analysis for bicluster acquisition". In: *Bioinformatics* 26.12 (2010), pp. 1520–1527.

[210] Pablo A Jaskowiak, Ricardo JGB Campello, and Ivan G Costa. "Proximity measures for clustering gene expression microarray data: a validation methodology and a comparative analysis". In: *IEEE/ACM transactions on computational biology and bioinformatics* 10.4 (2013), pp. 845–857.

[211] Paul T Spellman, Gavin Sherlock, Michael Q Zhang, Vishwanath R Iyer, Kirk Anders, Michael B Eisen, Patrick O Brown, David Botstein, and Bruce Futcher. "Comprehensive identification of cell cycle–regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization". In: *Molecular biology of the cell* 9.12 (1998), pp. 3273–3297.

[212] Audrey P Gasch, Paul T Spellman, Camilla M Kao, Orna Carmel-Harel, Michael B Eisen, Gisela Storz, David Botstein, and Patrick O Brown. "Genomic expression programs in the response of yeast cells to environmental changes". In: *Molecular biology of the cell* 11.12 (2000), pp. 4241–4257.

[213] Shelley Chu, Joe DeRisi, Michael Eisen, Jon Mulholland, David Botstein, Patrick O Brown, and Ira Herskowitz. "The transcriptional program of sporulation in budding yeast". In: *Science* 282.5389 (1998), pp. 699–705.

[214] Yee Hwa Yang, Sandrine Dudoit, Percy Luu, and Terence P Speed. "Normalization for cDNA microarry data". In: *Microarrays: optical technologies and informatics*. Vol. 4266. SPIE. 2001, pp. 141–152.

[215] Mohammad Nazmul Haque, Nasimul Noman, Regina Berretta, and Pablo Moscato. "Heterogeneous ensemble combination search using genetic algorithm for class imbalanced data classification". In: *PloS one* 11.1 (2016), e0146116.

[216] Harith Al-Sahaf, Ying Bi, Qi Chen, Andrew Lensen, Yi Mei, Yanan Sun, Binh Tran, Bing Xue, and Mengjie Zhang. "A survey on evolutionary machine learning". In: *Journal of the Royal Society of New Zealand* 49.2 (2019), pp. 205–228.

[217] Antonio J Nebro, Juan J Durillo, and Matthieu Vergne. "Redesigning the jMetal Multi-Objective Optimization Framework". In: *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation* (2015). DOI: 10.1145/2739482. URL: http://dx.doi.org/10.1145/2739482.2768462.

[218] Kalyanmoy Deb, Ram Bhushan Agrawal, et al. "Simulated binary crossover for continuous search space". In: *Complex systems* 9.2 (1995), pp. 115–148.

[219] Larry J. Eshelman and J. David Schaffer. "Real-Coded Genetic Algorithms and Interval-Schemata". In: *Foundations of Genetic Algorithms*. Ed. by L. DARRELL WHITLEY. Vol. 2. Foundations of Genetic Algorithms. Elsevier, 1993, pp. 187–202. DOI: https://doi.org/10.1016/B978-0-08-094832-4.50018-0. URL: https://www.sciencedirect.com/science/article/pii/B9780080948324500180.

[220] Rainer Storn and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces". In: *Journal of global optimization* 11.4 (1997), pp. 341–359.

[221] M. Srinivas and L.M. Patnaik. "Adaptive probabilities of crossover and mutation in genetic algorithms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 24.4 (1994), pp. 656–667. DOI: 10.1109/21.286385.

[222] Haruna Chiroma, Sameem Abdulkareem, Adamu Abubakar, Akram Zeki, Abdulsalam Ya'u Gital, and Mohammed Joda Usman. "Correlation study of genetic algorithm operators: crossover and mutation probabilities". In: *Proceedings of the International Symposium on Mathematical Sciences and Computing Research*. 2013, pp. 6–7.

[223] Earl Cox. *Fuzzy modeling and genetic algorithms for data mining and exploration*. Elsevier, 2005.

[224] Anne Claire Haury, Fantine Mordelet, Paola Vera-Licona, and Jean Philippe Vert. "TIGRESS: Trustful Inference of Gene REgulation using Stability Selection". In: *BMC Systems Biology* 6 (1 Nov. 2012), pp. 1–17. ISSN: 17520509. DOI: 10.1186/1752-0509-6-145/TABLES/5. URL: https://bmcsystbiol.biomedcentral.com/articles/10.1186/1752-0509-6-145.

[225] Rosa Aghdam, Mojtaba Ganjali, Xiujun Zhang, and Changiz Eslahchi. "CN: a consensus algorithm for inferring gene regulatory networks using the SORDER algorithm and conditional mutual information test". In: *Molecular BioSystems* 11.3 (2015), pp. 942–949.

[226] Sandro Hurtado, José García-Nieto, Ismael Navas-Delgado, Antonio J. Nebro, and José F. Aldana-Montes. "Reconstruction of gene regulatory networks with multi-objective particle swarm optimisers". In: *Applied Intelligence* 51 (4 Apr. 2021), pp. 1972–1991. ISSN: 15737497. DOI: 10.1007/S10489-020-01891-1.

[227] Jamshid Pirgazi and Ali Reza Khanteymoori. "A robust gene regulatory network inference method base on Kalman filter and linear regression". In: *PloS one* 13.7 (2018), e0200094.

[228] José García-Nieto, Antonio J. Nebro, and José F. Aldana-Montes. "Inference of gene regulatory networks with multi-objective cellular genetic algorithm". In: *Computational Biology and Chemistry* 80 (June 2019), pp. 409–418. ISSN: 1476-9271. DOI: 10.1016/J.COMPBIOLCHEM.2019.05.003.

[229] Luis F Iglesias-Martinez, Walter Kolch, and Tapesh Santra. "BGRMI: A method for inferring gene regulatory networks from time-course gene expression data and its application in breast cancer research". In: *Scientific Reports* 6.1 (2016), pp. 1–12.

[230] Takaya Saito and Marc Rehmsmeier. "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets". In: *PloS one* 10.3 (2015), e0118432.

[231] Luis F. Iglesias-Martinez, Barbara De Kegel, and Walter Kolch. "KBoost: a new method to infer gene regulatory networks from gene expression data". In: *Scientific Reports 2021 11:1* 11 (1 July 2021), pp. 1–13. ISSN:

2045-2322. DOI: 10.1038/s41598-021-94919-6. URL: https://www.nature.com/articles/s41598-021-94919-6.

[232] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms". In: *Swarm and Evolutionary Computation* 1.1 (2011), pp. 3–18.

[233] Patrik Andersson, Yunlong Yang, Kayoko Hosaka, Yin Zhang, Carina Fischer, Harald Braun, Shuzhen Liu, Guohua Yu, Shihai Liu, Rudi Beyaert, et al. "Molecular mechanisms of IL-33–mediated stromal interactions in cancer metastasis". In: *JCI insight* 3.20 (2018).

[234] Shaniya Ahmad, Prithvi Singh, Archana Sharma, Shweta Arora, Nitesh Shriwash, Arshad Husain Rahmani, Saleh A. Almatroodi, Kailash Manda, Ravins Dohare, and Mansoor Ali Syed. "Transcriptome Meta-Analysis Deciphers a Dysregulation in Immune Response-Associated Gene Signatures during Sepsis". In: *Genes* 10.12 (2019). ISSN: 2073-4425. DOI: 10.3390/genes10121005. URL: https://www.mdpi.com/2073-4425/10/12/1005.

[235] Renjian Xie, Bifei Li, Lee Jia, and Yumei Li. "Identification of core genes and pathways in melanoma metastasis via bioinformatics analysis". In: *International journal of molecular sciences* 23.2 (2022), p. 794.

[236] Naomi E Reijmerink, Dirkje S Postma, Marcel Bruinenberg, Ilja M Nolte, Deborah A Meyers, Eugene R Bleecker, and Gerard H Koppelman. "Association of IL1RL1, IL18R1, and IL18RAP gene cluster polymorphisms with asthma and atopy". In: *Journal of Allergy and Clinical Immunology* 122.3 (2008), pp. 651–654.

[237] Olga E Savenije, Jestinah M Mahachie John, Raquel Granell, Marjan Kerkhof, F Nicole Dijk, Johan C de Jongste, Henriëtte A Smit, Bert Brunekreef, Dirkje S Postma, Kristel Van Steen, et al. "Association of IL33–IL-1 receptor–like 1 (IL1RL1) pathway polymorphisms with wheezing phenotypes and asthma in childhood". In: *Journal of Allergy and Clinical Immunology* 134.1 (2014), pp. 170–177.

[238] Tomomitsu Hirota, Atsushi Takahashi, Michiaki Kubo, Tatsuhiko Tsunoda, Kaori Tomita, Masafumi Sakashita, Takechiyo Yamada, Shigeharu Fujieda, Shota Tanaka, Satoru Doi, et al. "Genome-wide association study identifies eight new susceptibility loci for atopic dermatitis in the Japanese population". In: *Nature genetics* 44.11 (2012), pp. 1222–1226.

[239] David Ellinghaus, Hansjörg Baurecht, Jorge Esparza-Gordillo, Elke Rodríguez, Anja Matanovic, Ingo Marenholz, Norbert Hübner, Heidi Schaarschmidt, Natalija Novak, Sven Michel, et al. "High-density genotyping study iden-

tifies four new susceptibility loci for atopic dermatitis". In: *Nature genetics* 45.7 (2013), pp. 808–812.

[240] Pierre Larrieu, Laure-Hélène Ouisse, Yannick Guilloux, Francine Jotereau, and Jean-François Fonteneau. "A HLA-DQ5 restricted Melan-A/MART-1 epitope presented by melanoma tumor cells to CD4+ T lymphocytes". In: *Cancer Immunology, Immunotherapy* 56.10 (2007), pp. 1565–1575.

[241] Biao Huang, Wei Han, Zu-Feng Sheng, and Guo-Liang Shen. "Identification of immune-related biomarkers associated with tumorigenesis and prognosis in cutaneous melanoma patients". In: *Cancer cell international* 20.1 (2020), pp. 1–15.

[242] Xu Wang, Francisco Almazan, Yoel Genaro Montoyo-Pujol, Antonia Martin-Casares, Aurelio Martin, Teresa Cabrera, and Miguel Angel López-Nevot. "HLA-DRB116: 01 and HLA-DQB105: 02 Alleles Influence the Susceptibility and Progression of Cutaneous Malignant Melanoma". In: *Journal of Oncology* 2021 (2021).

[243] Santos Kumar Baliarsingh, Khan Muhammad, and Sambit Bakshi. "SARA: a memetic algorithm for high-dimensional biomedical data". In: *Applied Soft Computing* 101 (2021), p. 107009.

[244] Leonardo Correa, Bruno Borguesan, Camilo Farfan, Mario Inostroza-Ponta, and Marcio Dorn. "A memetic algorithm for 3D protein structure prediction problem". In: *IEEE/ACM transactions on computational biology and bioinformatics* 15.3 (2016), pp. 690–704.

[245] Maoguo Gong, Zhenglin Peng, Lijia Ma, and Jiaxiang Huang. "Global biological network alignment by using efficient memetic algorithm". In: *IEEE/ACM transactions on computational biology and bioinformatics* 13.6 (2015), pp. 1117–1129.

[246] Leonardo de Lima Corrêa and Márcio Dorn. "A multi-population memetic algorithm for the 3-D protein structure prediction problem". In: *Swarm and Evolutionary Computation* 55 (2020), p. 100677.

[247] Luowen Liu et al. "Reconstructing gene regulatory networks via memetic algorithm and LASSO based on recurrent neural networks". In: *Soft Computing* 24 (2020), pp. 4205–4221.

[248] Fu Yin, Jiarui Zhou, Weixin Xie, and Zexuan Zhu. "Inferring sparse genetic regulatory networks based on maximum-entropy probability model and multi-objective memetic algorithm". In: *Memetic Computing* 15.1 (2023), pp. 117–137.

[249] Nobuo Namura. "Surrogate-assisted Reference Vector Adaptation to Various Pareto Front Shapes for Many-objective Bayesian Optimization". In: *2021 IEEE Congress on Evolutionary Computation (CEC)*. 2021, pp. 901–908. DOI: 10.1109/CEC45853.2021.9504917.

[250] Carlos A Coello Coello, Gregorio Toscano Pulido, and Maximino Salazar Lechuga. “Handling multiple objectives with particle swarm optimization”. In: *IEEE Transactions on evolutionary computation* 8.3 (2004), pp. 256–279.

[251] Saku Kukkonen and Jouni Lampinen. “GDE3: The third evolution step of generalized differential evolution”. In: *2005 IEEE congress on evolutionary computation*. Vol. 1. IEEE. 2005, pp. 443–450.

[252] Antonio J Nebro, Juan José Durillo, Jose Garcia-Nieto, CA Coello Coello, Francisco Luna, and Enrique Alba. “SMPSO: A new PSO-based metaheuristic for multi-objective optimization”. In: *2009 IEEE Symposium on computational intelligence in multi-criteria decision-making (MCDM)*. IEEE. 2009, pp. 66–73.

[253] Shigeyoshi Tsutsui, Masayuki Yamamura, and Takahide Higuchi. “Multi-parent recombination with simplex crossover in real coded genetic algorithms”. In: *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*. 1999, pp. 657–664.

[254] David Hadka. *MOEA framework-a free and open source Java framework for multiobjective optimization*. 2012.

[255] Rob Eisinga, Tom Heskes, Ben Pelzer, and Manfred Te Grotenhuis. “Exact p-values for pairwise comparison of Friedman rank sums, with application to comparing classifiers”. In: *BMC bioinformatics* 18.1 (2017), pp. 1–18.

[256] Albert-László Barabási. *Linked: The new science of networks*. 2003.

[257] Maryam Nazarieh, Andreas Wiese, Thorsten Will, Mohamed Hamed, and Volkhard Helms. “Identification of key player genes in gene regulatory networks”. In: *BMC Systems Biology* 10 (2016), pp. 1–12.

[258] Julia Åkesson, Zelmina Lubovac-Pilav, Rasmus Magnusson, and Mika Gustafsson. “ComHub: Community predictions of hubs in gene regulatory networks”. In: *BMC bioinformatics* 22 (2021), pp. 1–12.

[259] Sneh Gulati and Samuel Shapiro. “Goodness-of-fit tests for Pareto distribution”. In: *Statistical models and methods for biomedical and technical systems* (2008), pp. 259–274.

[260] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. “Superfamilies of evolved and designed networks”. In: *Science* 303.5663 (2004), pp. 1538–1542.

[261] Dimitrios Michail, Joris Kinable, Barak Naveh, and John V Sichi. “JGraphT—A Java library for graph data structures and algorithms”. In: *ACM Transactions on Mathematical Software (TOMS)* 46.2 (2020), pp. 1–29.

[262] Ricardo Pinho, Victor Garcia, Manuel Irimia, and Marcus W Feldman. "Stability depends on positive autoregulation in Boolean gene regulatory networks". In: *PLoS computational biology* 10.11 (2014), e1003916.

[263] Eun-young Cho. "Jprofiler: Code coverage analysis tool for omp project". In: *Technical Report: CMU 17–654 & 17, Tech. Rep.* (2006).

[264] Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L Gable, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, et al. "The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest". In: *Nucleic acids research* 51.D1 (2023), pp. D638–D646.

[265] Sara Principe, Enrique Zapater-Latorre, Leo Arribas, Enrique Garcia-Miragall, and Jose Bagan. "Salivary IL-8 as a putative predictive biomarker of radiotherapy response in head and neck cancer patients". In: *Clinical Oral Investigations* (2022), pp. 1–12.

[266] Nofar Erlichman, Tamir Baram, Tsipi Meshel, Dina Morein, Benny Da'adoosh, and Adit Ben-Baruch. "Tumor cell-autonomous pro-metastatic activities of PD-L1 in human breast cancer are mediated by PD-L1-S283 and chemokine axes". In: *Cancers* 14.4 (2022), p. 1042.

[267] Marie Simonneau, Eric Frouin, Vincent Huguier, Cynthia Jermidi, Jean François Jégou, Julie Godet, Anne Barra, Isabelle Paris, Pierre Levillain, Sevda Cordier-Dirikoc, et al. "Oncostatin M is overexpressed in skin squamous-cell carcinoma and promotes tumor progression". In: *Oncotarget* 9.92 (2018), p. 36457.

[268] Ranka Kanda, Yuko Miyagawa, Osamu Wada-Hiraike, Haruko Hiraike, Kazunori Nagasaka, Eiji Ryo, Tomoyuki Fujii, Yutaka Osuga, and Takuya Ayabe. "Ulipristal acetate simultaneously provokes antiproliferative and proinflammatory responses in endometrial cancer cells". In: *Heliyon* 8.1 (2022).

[269] AP Masilamani, R Ferrarese, E Kling, NK Thudi, Hoon Kim, DM Scholtens, F Dai, M Hadler, T Unterkircher, L Platania, et al. "KLF6 depletion promotes NF-κB signaling in glioblastoma". In: *Oncogene* 36.25 (2017), pp. 3562–3575.

[270] Hsiang-Chi Huang, Bi-He Cai, Ching-Shu Suen, Hsueh-Yi Lee, Ming-Jing Hwang, Fu-Tong Liu, and Reiji Kannagi. "BGN/TLR4/NF-κB Mediates Epigenetic Silencing of Immunosuppressive Siglec Ligands in Colon Cancer Cells". In: *Cells* 9.2 (2020), p. 397.

[271] Shuk-Ling Chau, Joanna Hung-Man Tong, Chit Chow, Johnny Sheung-Him Kwan, Raymond Wai-Ming Lung, Lau-Ying Chung, Edith Ka-Yee Tin, Shela Shu-Yan Wong, Alvin Ho-Kwan Cheung, Rainbow Wing-Hung

Lau, et al. “Distinct molecular landscape of Epstein–Barr virus associated pulmonary lymphoepithelioma-like carcinoma revealed by genomic sequencing”. In: *Cancers* 12.8 (2020), p. 2065.

[272] Andrzej P Wierzbicki. “The use of reference objectives in multiobjective optimization”. In: *Multiple criteria decision making theory and application: Proceedings of the third conference Hagen/Königswinter, West Germany, August 20–24, 1979*. Springer. 1980, pp. 468–486.

[273] Yali Wang, Steffen Limmer, Markus Olhofer, Michael Emmerich, and Thomas Bäck. “Automatic preference based multi-objective evolutionary algorithm on vehicle fleet maintenance scheduling optimization”. In: *Swarm and Evolutionary Computation* 65 (2021), p. 100933.

[274] Hao Li, Dezhong Li, Maoguo Gong, Jianzhao Li, A Kai Qin, Lining Xing, and Fei Xie. “Sparse Hyperspectral Unmixing With Preference-Based Evolutionary Multiobjective Multitasking Optimization”. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* (2024).

[275] A.P. Wierzbicki. “Reference Point Approaches”. In: *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*. Ed. by Tomas Gal, Theodor J. Stewart, and Thomas Hanne. Boston, MA: Springer US, 1999, pp. 237–275. ISBN: 978-1-4615-5025-9. DOI: 10.1007/978-1-4615-5025-9_9. URL: https://doi.org/10.1007/978-1-4615-5025-9_9.

[276] Juan J Durillo and Antonio J Nebro. “jMetal: A Java framework for multi-objective optimization”. In: *Advances in engineering software* 42.10 (2011), pp. 760–771.

[277] J. Molina, L.V. Santana, A.G. Hernández-Díaz, C.A. Coello Coello, and R. Caballero. “g-dominance: Reference point based dominance for multiobjective metaheuristics”. In: *European Journal of Operational Research* 197.2 (Sept. 2009), pp. 685–692. URL: https://ideas.repec.org/a/eee/ejores/v197y2009i2p685-692.html.

[278] Hisao Ishibuchi, Lie Meng Pang, and Ke Shang. “A new framework of evolutionary multi-objective algorithms with an unbounded external archive”. In: *ECAI 2020*. IOS Press, 2020, pp. 283–290.

[279] Phillip Bonacich. “Some unique properties of eigenvector centrality”. In: *Social networks* 29.4 (2007), pp. 555–564.

[280] Michelle Girvan and Mark EJ Newman. “Community structure in social and biological networks”. In: *Proceedings of the national academy of sciences* 99.12 (2002), pp. 7821–7826.

[281] Joanna E Handzlik, Yen Lee Loh, and Manu. “Dynamic modeling of transcriptional gene regulatory networks”. In: *Modeling Transcriptional Regulation: Methods and Protocols* (2021), pp. 67–97.

[282] Archibald Vivian Hill. “The possible effects of the aggregation of the molecules of hemoglobin on its dissociation curves”. In: *j. physiol.* 40 (1910), pp. iv–vii.

[283] Moises Santillán. “On the use of the Hill functions in mathematical models of gene regulatory networks”. In: *Mathematical Modelling of Natural Phenomena* 3.2 (2008), pp. 85–97.

[284] Silpa Bhaskaran, Umesh P, and Achuthsankar S Nair. “Hill equation in modeling transcriptional regulation”. In: *Systems and synthetic biology* (2015), pp. 77–92.

[285] John R Dormand and Peter J Prince. “A family of embedded Runge-Kutta formulae”. In: *Journal of computational and applied mathematics* 6.1 (1980), pp. 19–26.

[286] Wei Liu, Yu Yang, Xu Lu, Xiangzheng Fu, Ruiqing Sun, Li Yang, and Li Peng. “NSRGRN: a network structure refinement method for gene regulatory network inference”. In: *Briefings in Bioinformatics* 24.3 (2023), bbad129.

[287] Karen Giménez-Orenga, Eva Martín-Martínez, Lubov Nathanson, and Elisa Oltra. “HERV activation segregates ME/CFS from fibromyalgia while defining a novel nosologic entity”. In: (Feb. 2025). DOI: 10.7554/elife.104441.1. URL: http://dx.doi.org/10.7554/eLife.104441.1.

[288] Brian Walitt, Komudi Singh, Samuel R LaMunion, Mark Hallett, Steve Jacobson, Kong Chen, Yoshimi Enose-Akahata, Richard Apps, Jennifer J Barb, Patrick Bedard, et al. “Deep phenotyping of post-infectious myalgic encephalomyelitis/chronic fatigue syndrome”. In: *Nature Communications* 15.1 (2024), p. 907.

[289] Shigekazu Nagata and Masato Tanaka. “Programmed cell death and the immune system”. In: *Nature Reviews Immunology* 17.5 (2017), pp. 333–340.

[290] Joseph T Opferman and Stanley J Korsmeyer. “Apoptosis in the development and maintenance of the immune system”. In: *Nature immunology* 4.5 (2003), pp. 410–415.

[291] Julia M Marchingo and Doreen A Cantrell. “Protein synthesis, degradation, and energy metabolism in T cell immunity”. In: *Cellular & Molecular Immunology* 19.3 (2022), pp. 303–315.

[292] Philippe Pierre. “Immunity and the regulation of protein synthesis: surprising connections”. In: *Current opinion in immunology* 21.1 (2009), pp. 70–77.

[293] Laura A Solt. “Emerging insights and challenges for understanding T cell function through the proteome”. In: *Frontiers in Immunology* 13 (2022), p. 1028366.

[294] Federica Borghese and Felix IL Clanchy. “CD74: an emerging opportunity as a therapeutic target in cancer and autoimmune disease”. In: *Expert opinion on therapeutic targets* 15.3 (2011), pp. 237–251.

[295] Yujing Sun, Zhenhua Zhang, Qincheng Qiao, Ying Zou, Lina Wang, Tixiao Wang, Bo Lou, Guosheng Li, Miao Xu, Yanxiang Wang, et al. “Immunometabolic changes and potential biomarkers in CFS peripheral immune cells revealed by single-cell RNA sequencing”. In: *Journal of Translational Medicine* 22.1 (2024), p. 925.

[296] Philipp Hubel, Christian Urban, Valter Bergant, William M Schneider, Barbara Knauer, Alexey Stukalov, Pietro Scaturro, Angelika Mann, Linda Brunotte, Heinrich H Hoffmann, et al. “A protein-interaction network of interferon-stimulated genes extends the innate immune system landscape”. In: *Nature immunology* 20.4 (2019), pp. 493–502.

[297] Ke Shang, Hisao Ishibuchi, Linjun He, and Lie Meng Pang. “A survey on the hypervolume indicator in evolutionary multiobjective optimization”. In: *IEEE Transactions on Evolutionary Computation* 25.1 (2020), pp. 1–20.

[298] Maxence Vandromme, Julie Jacques, Julien Taillard, Laetitia Jourdan, and Clarisse Dhaenens. “A biclustering method for heterogeneous and temporal medical data”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.2 (2020), pp. 506–518.

[299] A Suppapitnarm, Keith A Seffen, Geoff T Parks, and PJ Clarkson. “A simulated annealing algorithm for multiobjective optimization”. In: *Engineering optimization* 33.1 (2000), pp. 59–85.

[300] José F Aldana-Martín, Juan J Durillo, and Antonio J Nebro. “Evolver: Meta-optimizing multi-objective metaheuristics”. In: *SoftwareX* 24 (2023), p. 101551.

[301] Jianjun Sun and Qinghua Huang. “Two stages biclustering with three populations”. In: *Biomedical Signal Processing and Control* 79 (2023), p. 104182.

[302] Laizhong Cui, Sudipta Acharya, Sumit Mishra, Yi Pan, and Joshua Zhexue Huang. “MMCO-Clus–an evolutionary co-clustering algorithm for gene selection”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.9 (2020), pp. 4371–4384.

[303] Khedidja Seridi, Laetitia Jourdan, and El-Ghazali Talbi. “Using multiobjective optimization for biclustering microarray data”. In: *Applied Soft Computing* 33 (2015), pp. 239–249.

[304] Cristian Andrés Gallo, Jessica Andrea Carballido, and Ignacio Ponzoni. “Microarray biclustering: A novel memetic approach based on the pisa platform”. In: *Evolutionary Computation, Machine Learning and Data Min-*

ing in Bioinformatics: 7th European Conference, EvoBIO 2009 Tübingen, Germany, April 15-17, 2009 Proceedings 7. Springer. 2009, pp. 44–55.

[305] Ujjwal Maulik, Anirban Mukhopadhyay, and Sanghamitra Bandyopadhyay. "Finding multiple coherent biclusters in microarray data using variable string length multiobjective genetic algorithm". In: *IEEE Transactions on Information Technology in Biomedicine* 13.6 (2009), pp. 969–975.

[306] Guilherme Palermo Coelho, Fabrício Olivetti de França, and Fernando J Von Zuben. "Multi-objective biclustering: When non-dominated solutions are not enough". In: *Journal of Mathematical Modelling and Algorithms* 8 (2009), pp. 175–202.

[307] Junwan Liu, Zhoujun Li, Xiaohua Hu, and Yiming Chen. "Multi-objective ant colony optimization biclustering of microarray data". In: *2009 IEEE international conference on granular computing*. IEEE. 2009, pp. 424–429.

[308] Mohsen Lashkargir, S Amirhassan Monadjemi, and Ahmad Baraani Dastjerdi. "A new biclustering method for gene expersion data based on adaptive multi objective particle swarm optimization". In: *2009 Second International Conference on Computer and Electrical Engineering*. Vol. 1. IEEE. 2009, pp. 559–563.

[309] Junwan Liu, Zhoujun Li, Xiaohua Hu, and Yiming Chen. "Biclustering of microarray data with MOSPO based on crowding distance". In: *BMC bioinformatics*. Vol. 10. Springer. 2009, pp. 1–10.

[310] Junwan Liu, Zhoujun Li, Feifei Liu, and Yiming Chen. "Multi-objective particle swarm optimization biclustering of microarray data". In: *2008 IEEE International Conference on Bioinformatics and Biomedicine*. IEEE. 2008, pp. 363–366.

[311] Ujjwal Maulik, Anirban Mukhopadhyay, Sanghamitra Bandyopadhyay, Michael Q Zhang, and Xuegong Zhang. "Multiobjective fuzzy biclustering in microarray data: method and a new performance measure". In: *2008 IEEE congress on evolutionary computation (IEEE World Congress on Computational Intelligence)*. IEEE. 2008, pp. 1536–1543.

[312] Federico Divina and Jesús S Aguilar-Ruiz. "A multi-objective approach to discover biclusters in microarray data". In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. 2007, pp. 385–392.

[313] David J Sheskin. *Handbook of parametric and nonparametric statistical procedures*. Chapman and hall/CRC, 2003.

[314] Anne Patrikainen and Marina Meila. "Comparing subspace clusterings". In: *IEEE Transactions on Knowledge and Data Engineering* 18.7 (2006), pp. 902–916.

[315] Kemal Eren, Mehmet Deveci, Onur Küçüktunç, and Ümit V Çatalyürek. "A comparative analysis of biclustering algorithms for gene expression data". In: *Briefings in bioinformatics* 14.3 (2013), pp. 279–292.

[316] Wassim Ayadi, Ons Maatouk, and Hend Bouziri. "Evolutionary biclustering algorithm of gene expression data". In: *2012 23rd International Workshop on Database and Expert Systems Applications*. IEEE. 2012, pp. 206–210.

[317] Wencheng Yin, Luis Mendoza, Jimena Monzon-Sandoval, Araxi O Urrutia, and Humberto Gutierrez. "Emergence of co-expression in gene regulatory networks". In: *PloS one* 16.4 (2021), e0247671.

[318] Victor A Padilha and Ricardo JGB Campello. "A systematic comparative evaluation of biclustering techniques". In: *BMC bioinformatics* 18 (2017), pp. 1–25.

[319] Drew V Klopfenstein, Liangsheng Zhang, Brent S Pedersen, Fidel Ramírez, Alex Warwick Vesztrocy, Aurélien Naldi, Christopher J Mungall, Jeffrey M Yunes, Olga Botvinnik, Mark Weigel, et al. "GOATOOLS: A Python library for Gene Ontology analyses". In: *Scientific reports* 8.1 (2018), p. 10872.

[320] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. "Gene ontology: tool for the unification of biology". In: *Nature genetics* 25.1 (2000), pp. 25–29.

[321] Ronald A Fisher. "On the interpretation of χ 2 from contingency tables, and the calculation of P". In: *Journal of the royal statistical society* 85.1 (1922), pp. 87–94.

[322] Yoav Benjamini and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the Royal statistical society: series B (Methodological)* 57.1 (1995), pp. 289–300.