ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Grado en Ingeniería de la Salud

Paquete de R para análisis de redes epistáticas

R Package for the analysis of epistatic networks

Realizado por

Samuel F. Benavides García

Tutorizado por

Oswaldo Trelles Salazar

Óscar Torreño Tirado

Departamento

Arquitectura de computadores

UNIVERSIDAD DE MÁLAGA

Málaga, septiembre de 2016

Fecha defensa:

El Secretario del Tribunal

Resumen

Las mutaciones o cambios en la dotación genética de los organismos, son uno de los mecanismos básicos de la evolución de las especies. Estas pueden encontrarse en zonas codificantes o zonas no codificantes del genoma, siendo las primeras -las encontradas en zonas codificantes- las que suscitan más interés ya que en muchos casos pueden tener efectos negativos asociados con la prevalencia de enfermedades en el organismo.

Entre estos efectos negativos, las mutaciones también pueden estar relacionadas con la aparición de enfermedades asumiendo que dichos cambios aumentan la propensión del individuo a padecer una enfermedad. Hoy en día se acepta que muchas de las enfermedades con origen en mutaciones son causadas por mecanismos epistáticos, esto es una interacción entre varias mutaciones que tienen efecto en su conjunto sobre una enfermedad. Estas relaciones epistáticas o de alto orden presentan dificultades para ser estudiadas debido a la complejidad computacional que presentan. Por lo tanto, es necesario diseñar nuevos métodos software de estudio de relaciones Genotipo-Fenotipo que permitan un análisis exhaustivo epistático, en este caso por parejas, complementando el análisis con teoría de grafos.

Este trabajo pretende contribuir a la investigación actual identificando relaciones Genotipo-Fenotipo, especialmente las que requieren análisis epistático, creando un paquete de software para R, uno de los lenguajes de programación más usados en biología computacional y en biomedicina, ofreciendo una interfaz de uso simplificada para completar el análisis de redes epistáticas creando un grafo de Polimorfismos de Nucleótidos Únicos o SNPs según sus siglas en inglés. A partir de dichos grafos el usuario puede obtener un grafo de genes y permitiendo al usuario obtener los genes o SNPs con mayor centralidad. La herramienta a desarrollar también permite al usuario almacenar sus resultados intermedios y finales para poderlos procesar posteriormente con otros programas externos. La herramienta a desarrollar permite su gestión con programas externos como Galaxy, un WMS (Workflow Management System) que permite la gestión de flujos de trabajo, y Cytoscape, un software especializado en la visualización y gestión de grafos. Ambas herramientas son comúnmente utilizada por el público objetivo.

El resultado de este trabajo pretende hacer posible la mejora en los estudios de esta área de investigación, permitiendo a más usuarios con menor conocimiento tecnológico realizar estudios GWAS expandiendo su uso y favoreciendo el incremento de resultados y conocimiento.

Palabras clave: epistasis, grafos, R, GWAS, Cytoscape, genotipo-fenotipo, Galaxy

Abstract

Mutations or changes in the genetic endowment are the engine of evolution. They can be found in coding or non-coding regions of the genome, those found on coding regions are usually the most interesting ones as in many cases they can lead to changes in the behaviour of the effects on the organism.

Between negative effects, mutations have been traditionally related to diseases assuming that these changes favor a certain disease. Nevertheless nowadays it is well known that many diseases are caused by epistasis which means that interactions between several mutations cause the disease. Due to the complexity it is problematic to study these epistatic relationships or high order relationships. As a result, new methods for studying genotype-phenotype need to be designed. These methods use, in this specific case, pairwise exhaustive epistatic analysis complement it with graph theory.

The main purpose of this study is to contribute to the current state of the art with a software capable to identify genotype-phenotype relationships, especially those that were described previously. A software package written in R, which is one of the most used programing languages among software developers in biology and biomedicine, is created and offers a simple interface to complete an epistatic network analysis. The package creates a graph of SNPs turning it into a genes allowing the user to choose the top genes or SNP and then gives the possibility to open the graph in an external software. The aim of the tool is to be able to allow its management with external programmes like they are Galaxy, for creation and management of workflows, and Cytoscape, a software for visualization and management of graphs. Both tools are commonly used by the target population.

The result of this project intends to provide all necessary tools to allow more users to perform GWAS expanding the use of it and favouring the increment of results and knowledge.

Keywords: epistasis, graph, R, GWAS, Cytoscape, geno-pheno, Galaxy

Index

Section	Page
1. Introduction	9
1.1. Motivation	9
1.2. Objectives	10
1.3. Methodology	11
2. State of the art	12
2.1. General review	12
2.2. Postgwas	14
2.3. Postgwas	16
2.4. Other software	20
3. Analysis and design	24
4. Development	30
4.1. Introduction	30
4.2. createGraphFromTable	30
4.3. fillInfo	30
4.4. replaceSNPwithGene	31
4.5. getTopDegree	32
4.6. openGraph	33
4.7. workflow implementation in Galaxy	35
5. Use case and validation	37
5.1. Dataset	37
5.2. Replication	37
5.3. Use case	41
6.1 Conclusions	45
6.2 Conclusiones (versión en español)	46
7. References	47

1. Introduction

1.1. Motivation

As consequence of new sequencing technological breakdown, commonly referred as 'next generation sequencing'[1] the price of sequencing nucleotide chains has decreased leading to an increase of the popularity of these technologies on the sequencing market. The price decreasement is related also to the fact of the frequent usage of these methods.

Nowadays one of the main applications of sequencing data (genes, genomes) is biomedical research, and it is also starting to be used for clinical purposes. Furthermore in last years, as a result of the decrease of its price, the technology is starting to be used often in hospitals in order to identify pathologies.

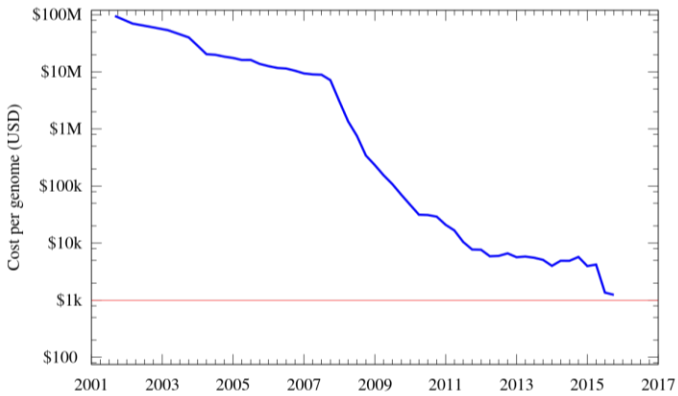

Figure 1 Cost to sequence a human genome expressed in thousand of USD[1]

Once the genome is sequenced or alternative methods such as SNP-microarray [2] are used to detect mutations, these mutations are taken into account to diagnose a patient or predict a disease. In order to make that possible it is necessary to perform a GWAS (Genome Wide Association Study) which is used principally to relate mutations to diseases, in other words to infer relationships genotype-phenotype. This type of studies has identified several mutations related to diseases such as asthma [3], breast cancer [4] or hypertension [5] among many others.

Traditionally a single isolated SNP (Single Nucleotide Polymorphism) [6] has been associated to a phenotypical expression. However this one-to-one correlations only cover a small part of the full spectrum of diseases, as many of the phenotype instances may be caused by epistasis, which means interactions between two or

[1] https://www.genome.gov/27541954/dna-sequencing-costs-data/

more mutations. One-to-one correlations are no longer of interest as most of the diseases caused by a single mutation have already been studied.

In order to perform an epistatic study it is necessary to perform an epistatic analysis in a specialized software like PLINK [7] which requires high computational effort only to detect pairwise epistasis increasing exponentially in three-wise or higher.

While the main computational bottleneck is the epistatic analysis there are other limitations in further steps. One of these limitations is that final users dealing with complex computational workflows lack the specific knowledge of programing, even to compose the sequence of programs to be executed as a unit.

All previously described leads us to the need of a software which easier to use by biologists and clinicians, allowing them to perform more complex functions with simple commands. There is a programing language that stands out among any other in the area of data analysis in bioinformatics and that is R [8], mostly together with Rstudio [9] which is a software developed to simplify the interaction with R. The software is used by professionals from the sector who require somehow the use of computer to perform analysis of their data. That allows them to program at high level using a simple interface to run a complex function which would require several code lines using a lower programing language as C. Besides there is a large active community developing specific software with many purposes and releasing them as packages so anyone can access them.

Workflows are becoming more important everyday in the area of bioinformatics, allowing to perform complex analysis so they can be reproduced and shared. A great number of WMSs (Workflow Management System) [10] have become more popular with Taverna [11], Chipster [12], GenePattern [13] and Galaxy [14] between them. As they offer a big increase of the usability and they -in general- make available a graphical interface to make it easier for the user to interact with them, allowing users lacking computational skills to launch a proper workflow which normally would require to use command line. As users are able to publish and share the results through internet, WMSs offer transparency, reusability and reproducibility.

1.2. Objectives

The aim of this study is to implement a software written in R to export as a package for further utilization on R. This package is able to create a SNP network from an input list of SNP pairs and then turn it into a genes network according to a flanking distance and consequently obtain the top genes according to a measure of the importance of the gene. Furthermore it will allow the user to visualize the graph either in an third party software as it is Cytoscape [15] or plot the graph on R.

It is necessary to create a method which, receiving as input a set of SNP ids and a flanking distance, will return the closest gene for every given SNP id. As previous methods such as those from Postgwas [16] are now out of service as stated in the state of the art, this method should query the NCBI (National Center for Biotechnology Information[2]) database while looking for protein coding genes. To avoid the connection process, a view of the Ensembl database will be previously downloaded so it can be accessed without need of internet connection, that leads to the requirement that the package must be updated with the local database any time the public database is updated.

There will be two different versions of the package coexisting as CRAN [17], the main R packages repository. The main packages repository, which can be accessed directly from R to easily install a package, does not allow any package which opens an external program as it is Cytoscape. However, there will be another version uploaded to the CRAN. This version will plot the top 6 clusters in the graph instead of opening an external program and then another package will be available for manual installation opening the graph in Cytoscape.

Moreover the package is self-contained and allows the user to obtain the graph and the most important genes without the need of using external software.

Finally, it is essential to create a workflow in the Galaxy system to be able to share it with the biologist and clinician community. Every method from this package will be made available for the scientific community throughout Galaxy.

1.3. Methodology

The methodology for this proposal comprises the following four consecutive sections. None of them can be done until the previous one is completed

- Design of the package: Study of the functional requirements and design of the package architecture.
- Full package implementation: Full iterative development of the package offering complete functionality.
- Testing and verification: Replication of previous studies to validate results and verify the package.
- Export package to Galaxy: Creation of a workflow and all necessary modules in Galaxy offering complete functionality.

[2] http://www.ncbi.nlm.nih.gov/

2. State of the Art

In this chapter we will discuss previous existing tools for the analysis of epistatic networks and related tools that make the analysis possible together with a general review about GWAS.

2.1. General review

A DNA mutation is a permanent alteration in the DNA sequence, which means that the sequence differs from what is found in DNA of most people. Mutations can affect from a single base pair to a large segment of even a chromosome that includes multiple genes.

A Single Nucleotide Polymorphism (SNP) is a mutation in an individual nucleotide base normally identified by comparison between two genomes [18]. It is the most common mutation occurring with a frequency around once per 500-1000 bases in the human genome [19]. The frequency means, in average, a total number of 14.4 million of humans SNPs [20]. Punctual mutations have an effect of paused evolution but it also offers biological diversity, making individuals within a same specie different between themselves. This explains for example why some patients react differently to the same medical treatment or have tendency to certain diseases. SNPs have been linked to evolution, familial traits or complex and common diseases [21].

This is the basic idea of what we are exploring. We want to know the effect of certain mutations (genotype) on the tendency of a patient to react in some way (phenotype) to external events.

As previously exposed, SNP can be located anywhere in the genome where this region may be coding or not. It is natural that SNPs located in coding regions may have a greater effect in the phenotype as they occur in areas which are related to the synthesis of proteins or may change the expression levels of such proteins.

That is why sometimes we talk about them in term of SNPs and other times in terms of genes where these SNPs occur. On the other side the flanks of the genes contain as well activating and other kind of signals for the genes, therefore the flaking distance will be set by default in 500 kbp following an approach previously taken in other studies [22].

This way, an epistatic network consists of a set of non-directional interactions between genetic elements such as SNPs or genes themselves.

The current approaches for epistasis detection are divided mainly in three groups [23]:

- Exhaustive analysis: Here all the possible interactions between SNPs within the dataset are tested.
- Data-driven filter: [24] Reduces the computational effort of the exhaustive analysis by applying statistical testing based on the information of the interaction.
- Biological filter: [25] Reduces the computational effort of the exhaustive analysis by applying biological filters such as a co-pathway membership.

Taking into account that from an input of 100,000 SNPs there are around 5×10^9 numbers of possible interactions and considering a study like the one performed for the Clarkson Disease [26] it is impossible to perform a complete epistatic analysis with a commodity computer even if the analysis was only pairwise incrementing exponentially the complexity according to the order of interactions as shown in Table 1. In the Clarkson Disease study it took approximately three days of running 800 cores and each of them with 4 GB of RAM from an input of 764537 SNPs.

n^{th} order of interactions	Size of the input	Total number of interactions
2	100.000	5×10^9
2	500000	12.5×10^{11}
3	100.000	1.67×10^{14}
3	500.000	2.08×10^{16}
4	100.000	4.17×10^{18}
4	500.000	2.6×10^{21}

Table 1 number of interactions according to input size and order

Nowadays there is quite a defined workflow to follow when we want to perform an epistatic analysis as defined in figure 2, finishing with an enrichment analysis indicating which is the biological functionality most affected by the set of genes obtained.

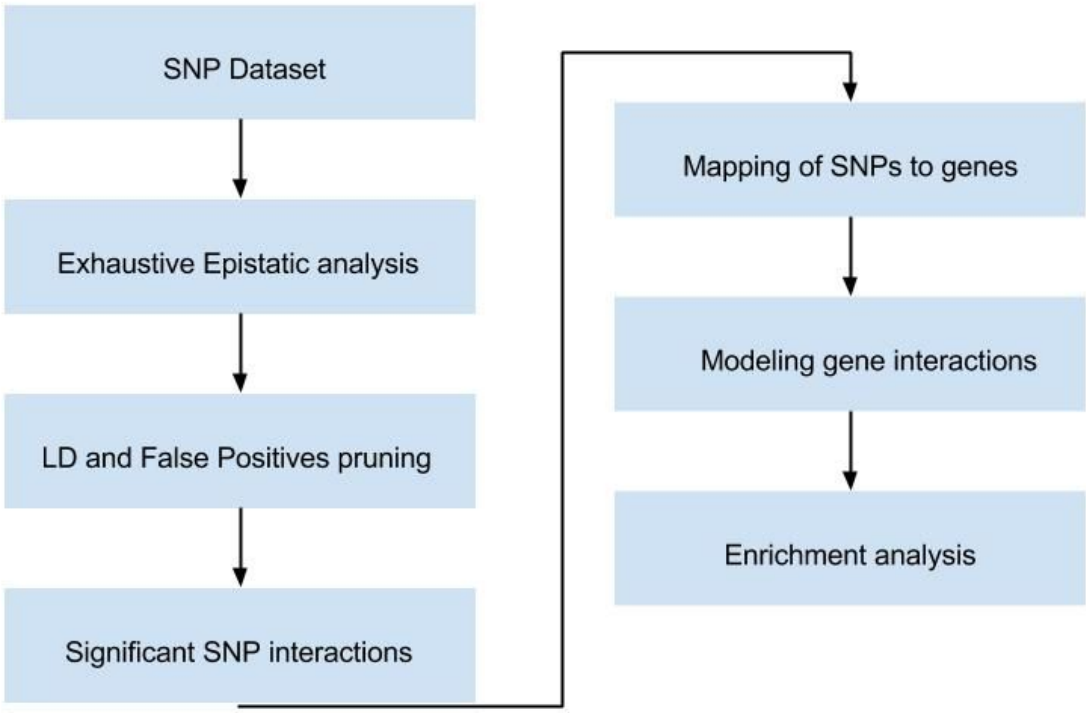

Figure 2 Example of complete workflow for epistatic analysis

Galaxy is one of the most used and cited softwares for workflow management in the field of bioinformatics. Regardless the fact that there are many other softwares, Galaxy is the most used in biology due to to its facility to reproduce and to the growing community that is creating and publishing new workflows [27].

2.2. The R programing language

Although most of the specialized software in the area of biology and even data science is developed to be used under the UNIX operating system, R can run independently in any of the three most used Operative systems OS (Windows, Ubuntu and Mac). Therefore only has to be taken into account on the moment to create functions which directly interact with the shell. We also have to think on an average final user who typically is not enough trained in computer knowledge and uses the most mainstream OS, which is Windows, just for the comfort that it offers to the user, therefore the package must be implemented to work in both OS.
The language in which this project will be implemented is R, as it is a free open language easy to get a basic knowledge , which allows to use it from the lowest level to the highest level based on the packages, where our target is placed. Apart from this, R has a great developers community around with thousand of contributors and more than two million users[28]. These circumstances make it easier to find any support or help on the web and at the same time provides a continuous flow of incoming new R packages.

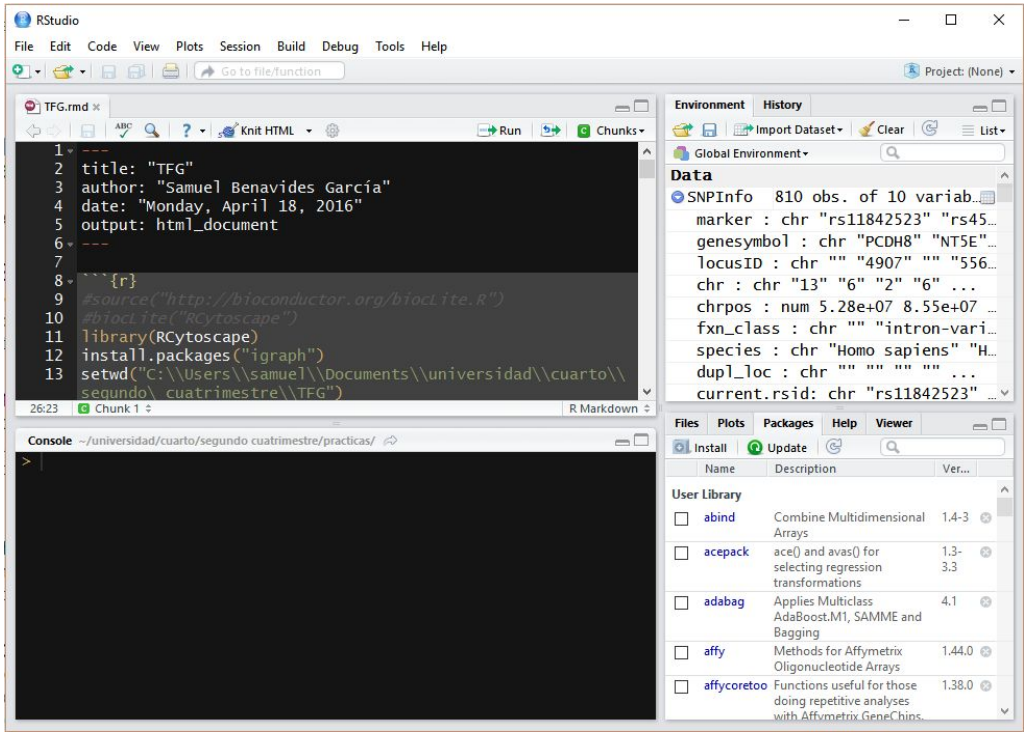

Figure 3 RStudio default layout, including a package manager (bottom-right), a console (bottom-left), an environment data display (top-right) and a text editor (top-left), this distribution might change according to the user.

RStudio is an user friendly IDE for R, as seen in Figure 3, which is also an independent platform. It is commonly used by biologists and all kind of clinicians because of its graphic interface and due to the fact it is easy to install and get to work with it as it requires almost no previous knowledge of the R syntax. These are the reasons why the package must be valid to use in RStudio.

Preferably the package should be accessible through the most used way to download a package which is the CRAN. It is simple to download a package from RStudio with just clicking on install and typing a name or with the sentence *>install.package("packageName")*.
The CRAN repository have quite strict rules to follow for a package to be published. The most restrictives are:

- Source packages may not contain any form of binary executable code.
- The code and examples provided in a package should never do anything which might be regarded as malicious or anti-social. That includes start an external software e.g. PDF viewer.
- Packages should be named in such way which does not conflict with any current or past CRAN package, nor any present in the Bioconductor [29] package.

Considering all previously mentioned it seems necessary that whole code will be written in R, even though R can run compiled C code. Using R opens the possibility of using external libraries such as '*igraph*' to build and manage part of the graphs to be created in the package, not only this but it makes the code fully open to the user.

Another point mentioned before is the name of the package. It may seem as an unimportant decision but it is the first contact with the user and as the user may not be looking for a specific package the name of the package should be descriptive. Many packages incorporate the letter 'R' in upper case into their names relating the package with the R technology. Considering this a good descriptive name would be '*epiR*', however, such naming convention is not currently in use neither it was used in any previous existing package in the CRAN or in Bioconductor as mentioned in the CRAN policy.

2.3. Postgwas

This R package offers several different functions for the interpretation of the results obtained by GWAS, including a method for identifying a gene giving a SNP and a flaking distance or linkage disequilibrium. It is also aimed to create a graph of SNPs to obtain a gene enrichment network. A more detailed view of the methods contained in the package can be seen in Figure 4.

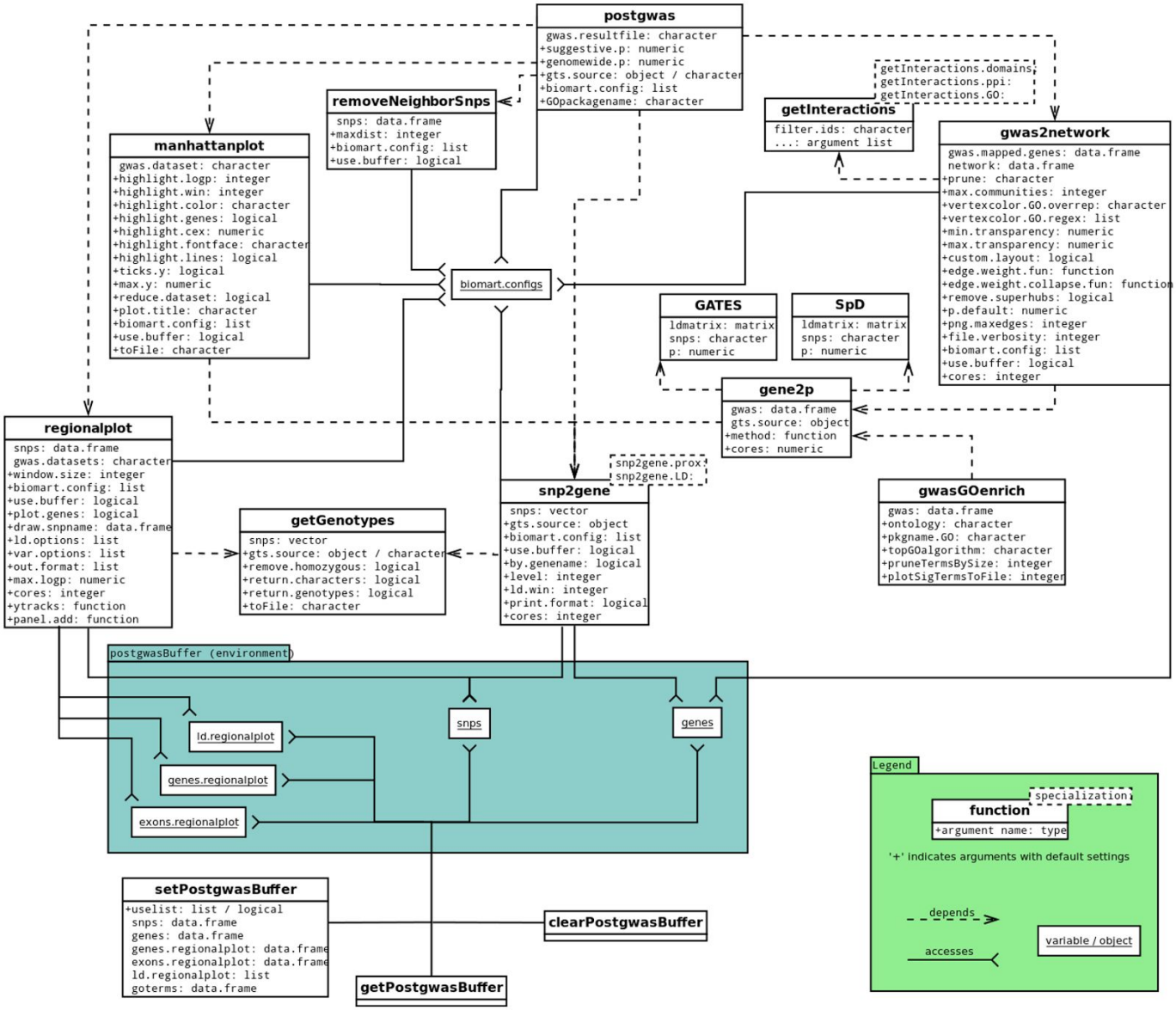

Figure 4 Diagram of functions in the postgwas package [16]

The package works focused in a dataframe which is the main enviroment, this dataframehas a function to create it "*setPostgwasBuffer*", a function to get a clone of the buffer "*getPostgwasBuffer*" and a function to clear it "*clearPostgwasBuffer*".

From this buffer all functions are applied, "*getGenotypes*" is a function developped to be used mainly by other functions, this retrieves when possible the genotype of a given set of SNPs, "*snp2gene*" is one of these functions and adds to the given dataframe a field named "gene" with the gene where the SNP is located, two specifications are created, one is "*snp2gene.prox*", which is the one arousing more interest, this functions adds a flaking distance where a SNP can be located to be placed on that specific gene; other specification is "*snp2gene.ld*" which takes into account the linkage disequilibrium to assign a gene to a SNP.

There are two plotting functions in this package, one of them is "*regionalPlot*", it creates regional plots for a given window size around a given list of SNPs. These regional plots contain one or more p-value graphs, gene information and LD between SNPs from the GWAS. Other plotting function is "*manhattanPlot*" it plots a manhattan plot, with SNPs exceeding a user-defined threshold being highlighted and annotated with closest genes.

Four other important functions are "*gene2p*", it calculates aggregate p-values for genes assigned to multiple SNPs, by taking into account the dependency between SNPs. Another important function is "*getInteractions*" which uses biomart to determine pathway annotations for each gene, then the common pathways between genes form interactions. A function to visualize previous network is "*gwas2network*" which displays the given network and highlights all given genes and their connecting edges using transparency effects. The last function is "*gwasGOenrich*" provides an interface for GO term enrichment, it returns a dataframe with id, terms and P-value columns containing identifier, description and enrichment p-value of all GO terms tested.

This might be one of the strongest references as it carries most of the functions we want to implement. The first use of this package will be essential in order to measure its performance and evaluate the user's experience and will be used to compare our developed software.

Firstly the package was removed in December 2013 from the CRAN[3] for no response nor update was received after a request to clean up warnings. Therefore it must be downloaded externally in a zipped file and installed manually instead of doing it directly from the CRAN repository. Normally when such a situation occurs it is because there is no package update adapting it to the rules of the CRAN.

Surprisingly no further work has been done with the package, as pointed out in the CRAN. The files are available in github, where you can encounter a misleading information in the help section which says that the package can be downloaded from the CRAN, being this version same as the last version available from the CRAN's archive this version is 1.11.

There is a summary function in the package named '*postgwas*', which has among its documentation a usage example. When running it not only there was no result but some errors were raised. Several attempts to run it were made and during none of them there was a positive result.

[3] https://cran.r-project.org/web/packages/postgwas/index.html

After carefully observing the messages obtained we can get to the conclusion that it is a problem using an external service as the error is raised every time a external service was invoked, it is discarded a possible mistake on the input data as it was used the sample data provided by the package.

Therefore none of the functions in this package could be used in future implementations as the lack of update together with the impossibility to run the code and its removal from the CRAN make this source code an unstable basis for further developed functions.

2.4. Other software

Although Postgwas is one of the packages with higher similar functionality than our package we want to create, there are many other softwares for their use in GWAS studies.

There exist a webpage which sort many kind of software for their use in omics science named “Omic Tools”[30], it offers as well tools for GWAS but these tools are mainly focused to single association studies.

There are many methods for detecting epistasis in a wide range of interfaces, I will describe only the few most used although there are many others, PLINK is one of the most well-known method, it takes a regressive approach and it’s interface is the command line, it is widely used and has a wide range of input formats, it supports bit-level based parallelism; BOOST [31] takes a regressive approach as well and has a command line interface, it performs an exhaustive search but has statistical power limitations; BEAM [32] takes a bayesian approach and also has a command line interface, it incorporates posterior information but does not support parallelism natively; MB-MDR [33] takes a data mining approach and is used through the command line, it is widely used and updated regularly but does not offer native support for Windows OS; FaST-LMM [34] takes a linear mixed model approach and is used also through the command line, it performs an exhaustive search but is computationally expensive.

A deeper analysis of the previous software will be done in this paragraph. PLINK includes a regression-based epistasis test. However it takes quite a lot of time to perform a test. A faster implementation was developed later on including native parallelism support [35], while this method is much faster it is only applicable to quantitative traits and cannot be used in case/control studies. BOOST is able to evaluate a small set of SNPs in a standard desktop computer. Bayesian methods are one of the alternatives to exhaustive methods like previous, this not require all

interactions to be tested, an example of this is BEAM which divides SNP in three different categories based on inner probabilities. Another way for detection of epistasis is machine learning and data mining MB-MDR is an example of this, it categorizes genotypes into high-risk and low-risk reducing data to a single dimension, however as it is an exhaustive method as well, it is generally applied to studies where the dataset has previously been reduced.

Epistatic analysis present computational issues and as methods become more complex and data sets increase in size, computational

Normally after the epistasis detection there are used specific software to obtain results, this software input normally is generic, that means that is not specially designed for epistatic analysis.

There is no similar package for management of epistatic networks after the epistatic analysis, many GWAS software found is based on single association studies, and those found which work with epistasis performs only the epistatic detection.

In the area of workflows, although there are many options, one interesting WMS is Galaxy as it is a web based tool with a great community on the area of the life science as can be seen in Table 2. Being web based makes easy to create and share workflows and tools by creating an instance of a workflow in a server so anyone can both download and access the designed tool, this makes Galaxy a perfect tool for the reproducibility of any study following a workflow implementation.

Galaxy runs completely on the server therefore there is no specific requirement from the user's computer to run any program. It is also a easy to use software allowing non-computational users to run a program with no need of dealing with the command line.

Date	Posts	Threads	Questions	Answers	Users
2014/07/09	8223	2978	2959	2908	2517
2014/10/13	9230	3252	3220	3252	3555
2015/01/23	10298	3547	3505	3646	4700
2015/07/24	13275	4379	4303	4616	6342
2016/01/29	15742	5035	4958	5469	7909
2016/08/15	18823	5935	5826	6463	9731

Table 2. Statistics from the past two years of the Galaxy Biostar forum showing users, posts, threads, questions and answer.[4]

There are three ways to use galaxy, first one is trough the official usegalaxy.org portal which offers plenty of tools thought for bioinformatics bust doesn't allow the user to add it's own modules, other way to use Galaxy is accessing to any of it's many servers [5] which have specialized modules depending on their expertise, the last way how to use Galaxy is installing a local instance, where you can add all necessary modules, this offers the opportunity to publish it through a public server.

The main view from a Galaxy page can be seen in Figure 5 and it shows the groups of modules on the left side where the epiR group with all necessary modules will be.

[4] https://wiki.galaxyproject.org/GalaxyProject/Statistics
[5] https://wiki.galaxyproject.org/PublicGalaxyServers

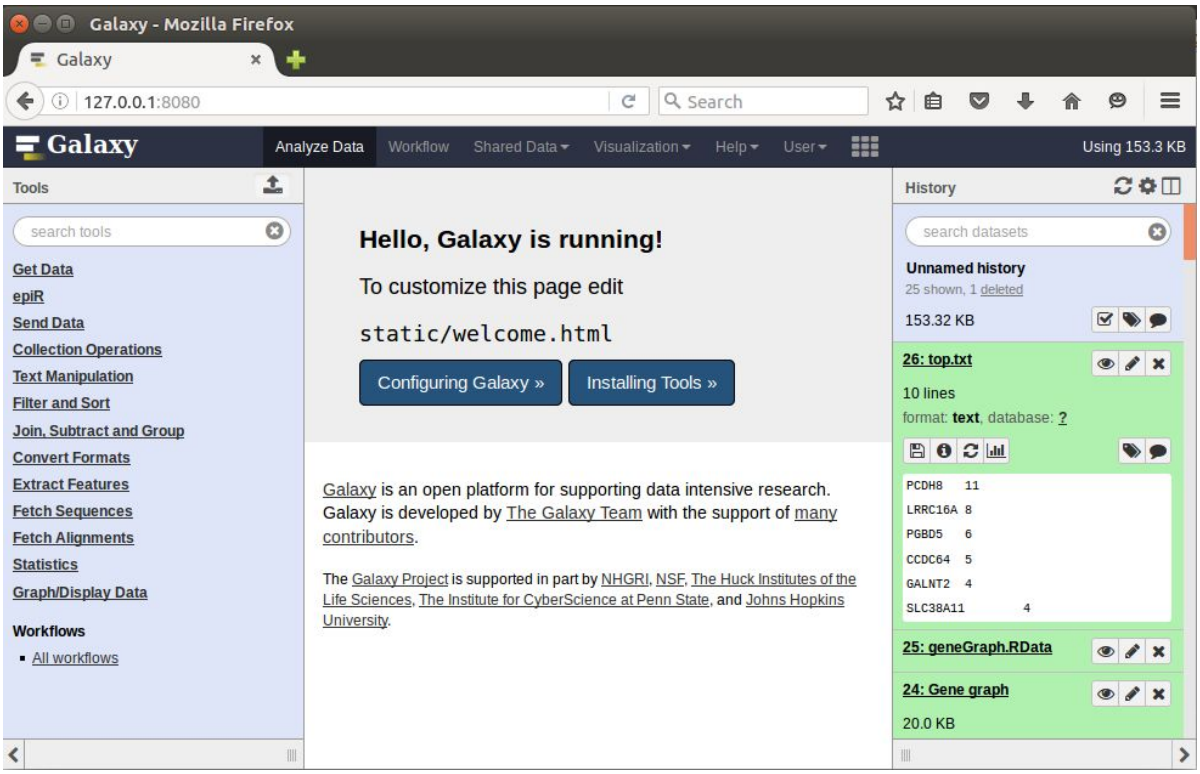

Figure 5 Main view of the galaxy home, on the left side there are the tools, on the right a history with all datasets and results obtained and in the upper part there is the menu.

On the area of the enrichment analysis David [36] which is the version developed by the NCBI or WebGestalt [37] developed by the zhang lab, both are interesting software for this purpose, and they are web-based but the second one offers more options at the time of performing the enrichment analysis. Considering the interface simplicity, as seen in Figure 6, and the variety of analysis WebGestalt is recommended.

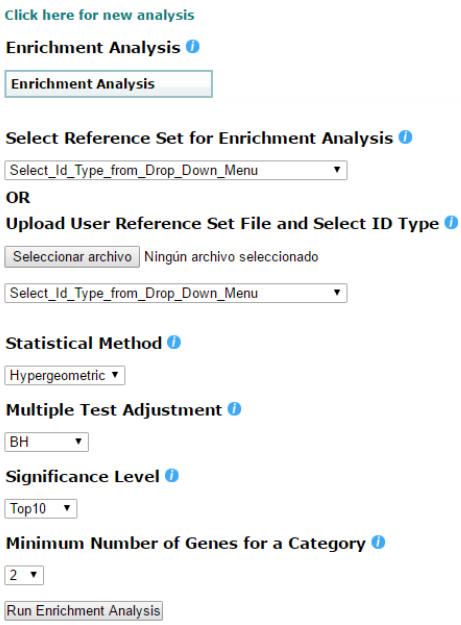

Figure 6 Main Webgestalt interface offering all possible parameters to the user

3. Analysis and Design

Taking into account the aim of this project the functionality that we want to implement must be chosen carefully, obtaining the requirements for our package.

The package has to offer a quick connection between the results and a software which is able to visualize the result graph, either a SNPs one or a genes one. Thus the first idea is to connect the package to Cytoscape, is a *de-facto* standard network visualization application in life science field. One option could be that a function directly launches Cytoscape with the required graph (this is the simplest option) and the other option would be to save the graph in a Cytoscape readable format and plot the top clusters in an image so the user can see the biggest clusters. While the first option seems to be the best, it requires to add Cytoscape to the system path, which is something that could cause some troubles to the users, on the other hand the second option offers low quality as plotted images cannot be zoomed.

The final decision was to invoke Cytoscape directly from the function, loading the required graph automatically (as seen in Figure 7) and saving the graph in *.gml* format which is a format that can be imported by Cytoscape. This leads to a conflict with the CRAN repository. The conflict was solved by implementing the package in two versions. The first one is meant to be published in the CRAN, which will plot an image instead, and another one is meant to be manually installed, this last version would launch Cytoscape.

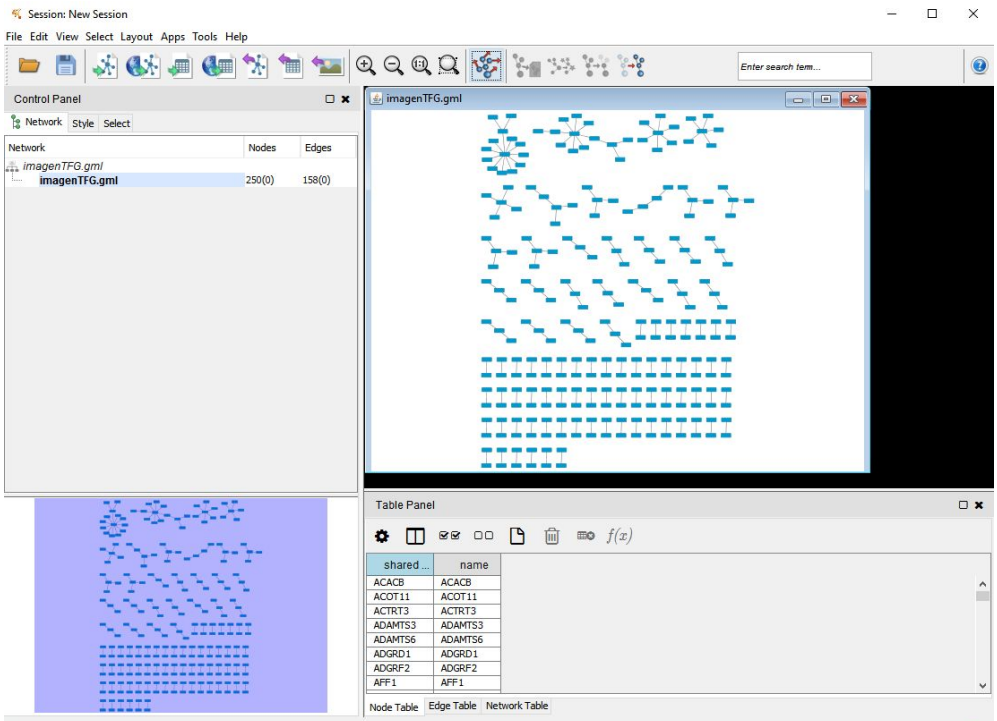

Figure 7 Cytoscape main view after loading a graph, in the lower left corner there is a miniature of the graph and a toolbar is placed on the uppest side.

Considering the big computational effort to perform an epistatic analysis it seems to be obvious that it has to be performed in a HPC infrastructure as computing time could be in the range of years if performed on an average user PC.

Therefore the input for the developed R package must be the result from the epistatic analysis, typically provided to biologists and clinicians in plain text edge list format as seen in Figure 7.

This input has also the advantage that as it is an edge list in plain text it is easily editable by experts applying their own knowledge (e.g., known SNPs which are not related to the disease, removal of certain relations as known from low information, etc). Another advantage is that the user can create his/her own dataset based on his/her experience or simply the user may want to perform an experiment without any previous need of obtaining biological samples.

A pipeline design will be extremely suitable for this project as it is based on a workflow, therefore the functions must share inputs and outputs and for this a WMS is a good option.

At this point we have a basic scheme about the workflow to be developed. Now we present a more detailed idea about the components:

As input we have an edge list which basically consist on the edges of a graph set in pairs as seen in figure 7. The edge list is made by pairs of RSIDs that correspond with the SNP's identificators. Second, we create a graph of SNPs and as well we will need a tool to get a gene from a given rsid and a flanking distance. As other previously existing functions like *snp2gene* from the package postgwas are currently out of service the gene graph will be generated from the SNP graph.

The input format is a plain text with two columns separated by tabulators where any of the columns has to contain a valid rsid as shown in figure 8. This file may not have a header.

SNP1	SNP2
rs1487013	rs1593270
rs12532565	rs17105765
rs12356602	rs11842523
rs129128	rs11842523

Figure 8 Example of an input edge list

To evaluate and visualize the graphs we need two tools, one to get the most relevant nodes from a graph according to a selected metric and another to visualize the graph. This issue has been already discussed with the conclusion that there will be one function plotting the top clusters and another opening cytoscape with a given graph.

The decision about which metric to use to evaluate the graph still needs to be made. To be able to make a proper decision about this matter firstly it is necessary to figure out how the final graph will look like and for that we have to in the first place think about what we are actually generating. We generate a graph from a set of SNPs which could be obtained in different ways. The most common relation will be between two SNPs but there will be many other higher order relations. Therefore our graph will be composed of many isolated clusters of SNPs related between themselves.

All previously exposed will be taken into account when deciding the metric from one of the exposed on Table 3, regarding that the graph will highly probable be composed by several disconnected networks, both betweenness and closeness centrality cannot be used because the distance between disconnected nodes from a graph is infinite thus the only remaining option is degree centrality.

Metric	Description	Algorithm	Reference
Degree centrality	The number of connections that a node has in a network.	$C_D(v) = deg(v)$	[38]
Closeness centrality	Inverse of the average length of the shortest paths to/from all the other nodes.	$C(x) = \frac{1}{\sum_y d(y,x)}$	[39]
Betweenness centrality	The number of network shortest paths that pass through a specific node.	$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$	[40]

Table 3 Most used graph theory metrics

The final design of the workflow may look like in Figure 9.

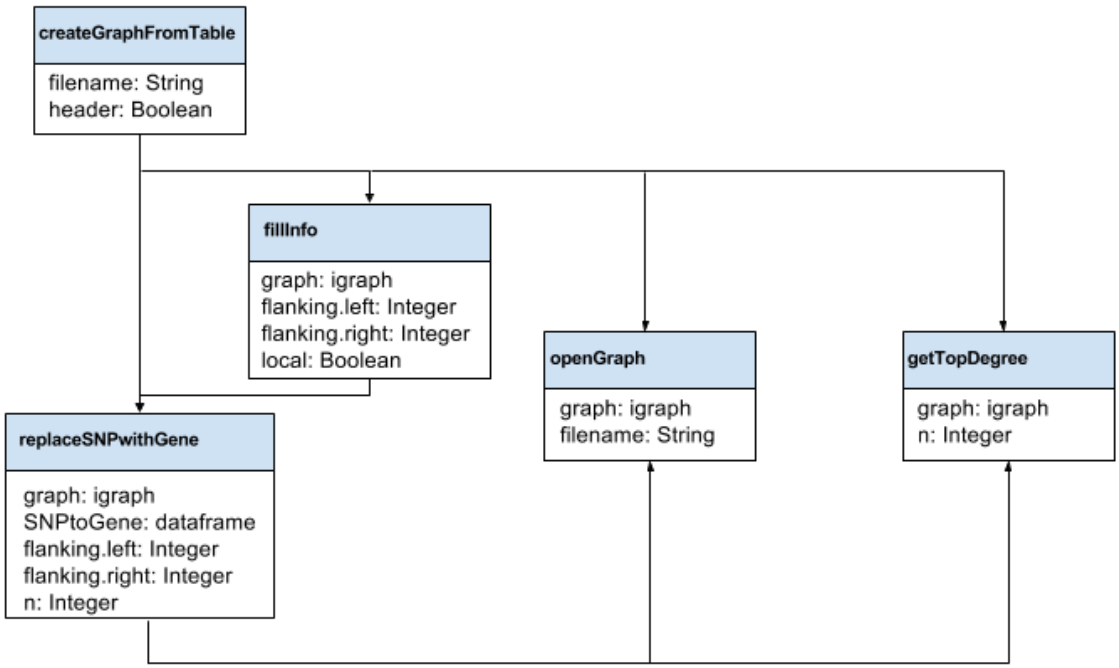

Figure 9 Final design of the software including all functions and connections

All inputs and outputs from the above functions are described from the Table 4 to Table 8, further information about functions can be found on the Development section.

	Parameter	Description	Default
Input	filename	String containing the path to an edge list text file	-
	header	Boolean indicating whether the file has header or not	TRUE
Output	SNPs graph	A R object containing a graph of the class *igraph*	-

Table 4 Parameters of the function createGraphFromTable

	Parameter	Description	Default
Input	graph	An *igraph* object containing a graph whose nodes are rsid	-
	flanking.left	Number of nucleotides to consider as flanking distance to the start	500,000
	flanking.right	Number of nucleotides to consider as flanking distance to the end	500,000
	local	Whether to use or not the local database	FALSE
Output	rsid, gene symbol and SNP position	A data frame containing the rsid, gene symbol (if found) and SNP position	-

Table 5 Inputs and outputs of the function fillInfo

	Parameter	Description	Default
Input	graph	An *igraph* object containing a graph whose nodes' name are rsids	-
	SNPtoGene	Dataframe resulting from fillInfo	-
	flanking.left	Number of nucleotides to consider as flanking distance to the start	500,000
	flanking.right	Number of nucleotides to consider as flanking distance to the end	500,000
	n	Maximum cluster size to be removed from the resulting graph	2
Output	gene graph	A R object containing a graph of the class *igraph*	-

Table 6 Parameters of the function replaceSNPwithGene

	Parameter	Description	Default
Input	graph	An *igraph* object containing a graph	-
	n	Size of the returning list	20
Output	top nodes	A named list containing the top n nodes	-

Table 7 Parameters of the function getTopDegree

	Parameter	Description	Default
Input	graph	An *igraph* object containing a graph	-
	filename	Name of the file where the graph will be saved, it may be a path	"graph.gml"
Output	Openable graph	In both cases the output will be a *.gml* file containing the graph in an Cytoscape importable format.	-

Table 8 Parameters of the function openGraph

The workflow ends with a biological enrichment analysis which provides the biological meaning to result. The enrichment analysis finds the most over-represented Gene Ontology [41] terms among them. The Gene Ontology is a computational model of system biology. In this way we can observe the biological paths expressed through an ontology and therefore when we perform the enrichment analysis on it we will obtain the most probably affected biological result. This will finish our analysis as we started it with a case control study selecting the SNPs that are most probably interacting in a pathology. We will create a graph and from it we will select the most relevant genes. At the end we will finish with selecting the most represented function in the given genes set. This function will probably be the cause of the pathology and the affected genes those resulting from the selection.

The enrichment analysis requires a lot of parameters and a good visualization to enable to satisfy the expert needs. However these requirements would lead to a very complicated function interface including many possible configuration parameters that possibly request an user interface. Nevertheless this is not the only problem, the visualization would have to be complex to be done in R and able to satisfy the user's requirements. Therefore we have made the decision to leave the enrichment analysis for an external resource, as explained in the state of the art WebGestalt is the state of the art.

4. Development

4.1. Introduction

During the development an iterative and incremental approach has been followed, thus the first version was developed and then we have been adding more functionality in different stages.

We made a decision to enable flexibility and customization regarding how and where to store the default values for parameters. We decide to store the default values in external storage and not hardcoded inside the function itself, so in case of need for change, the user only has to change the values in the external file. The chosen format to store those values is *.rda* although it is a binary file it is quickly read by R with no need of external functions.

4.2. createGraphFromTable

This function creates the graph from an edge list. It is based on the package *igraph*, which offers a function to create a graph of the *igraph* class from a matrix. So firstly a matrix is created from the text contained in the file. This function can as well create a gene graph or any other graph while the input format is a plain text edge list using tabulators as separators, therefore there will be no extra data validation as the input does not have to be a rsid.

4.3. fillInfo

This is one of the main functions as it is in charge of getting the genes from a set of rsid. The main idea was to replicate some studies using snp2gene function from the postgwas package but it is unfortunately unavailable, therefore the whole function had to be developed using only auxiliary packages to obtain the SNP information.
The package NCBI2R [42] has been used as an interface to get information from the NCBI. From a SNP graph the package is able to return the chromosome position and the gene where it is located and this information will be stored into a data frame. That will be the first step of the function and it will need an internet connection as it works through a web service. This fact is very important as new SNPs could be identified and uploaded to the NCBI database easily so there is no possibility of using a local database which would be quickly deprecated.

The next step is to fill the empty gene fields of the dataset according to a given flanking distance. In this step a different database will be consulted and that is a

gene database obtained from Ensembl [43], it consists of the complete list of known genes from the hg19 to be coherent with the result obtained by querying the NCBI database. Hg19 is a reference human genome and it is used to locate markers in a genome, therefore the location of genes will be relative to the hg19. So we obtain an instance of it by querying it to obtain protein coding genes symbols, chromosomes, the start positions and the end positions. Both start and end positions mean the beginning and the ending of the transcribed region of the gene. We keep a local copy of the view as presumably there will be no big changes in the database. This copy is stored in data frame format in a R data file and can be easily updated. The option of using the online database is offered to the user through a parameter on the function.

The closest gene is chosen for every gene by using the *min* function and subsequently an examination if the gene is located within the flanking distance is provided. In the affirmative case it is added to the data frame. Once the process for all the SNPs is finished, the data frame is returned. Function's pseudocode is shown in Figure 10.

```
Get a SNP graph
Get a rsid list from graph
Query the NCBI database
Get gene database instance
For each SNP not within a gene
        Get the closest gene
        If gene is in the SNP flanking distance
                replace empty space on the dataset
        End if
End for
Return the data frame
```

Figure 10 pseudocode of the function fillInfo

4.4. replaceSNPwithGene

This function creates an *igraph* object containing a graph whose nodes are gene symbols. This is basically to clone the SNP graph and replace the SNP by the one given in the data frame. After this is done, all not paired SNPs will be removed. Afterwards every isolated node will be removed as well, during this phase the user can choose to remove clusters of certain size so the smaller clusters will be removed. Once all this is done the graph has to be simplified as many nodes will be duplicated. As a result all duplicated nodes will join into a common node.

After several iterations we realized that it could be convenient to offer the opportunity of reassigning the flanking distance without need to use again the fillInfo function. Therefore this opportunity has been added and it will be triggered by setting as flanking distance a different number than default. The function pseudocode can be seen in Figure 11.
All graph management is performed by using the *igraph* package.

```
Clone SNP graph
If left fd or right fd not equal to default
        Get local database
        For each SNP not within a gene
                Get the closest gene
                If gene is in the SNP flanking distance
                        replace empty space on the dataset
                End if
        End for
End if
For each node in the SNP graph
        Get the node name
        If node name has associated gene
                Replace node name by gene symbol
        Else
                Remove node
        End if
End for
Assign same node to nodes sharing name
Simplify relations between nodes
Remove isolated or under given size nodes
Return graph
```

Figure 11 Pseudocode of the function replaceSNPwithGene

4.5. getTopDegree

This is another main function as it allows the user to obtain results, as previously explained it uses degree centrality metric to measure the top genes. The measuring returns a list of nodes with its score, then it has to be sorted and only the top number of scores will be returned. Aforesaid is again performed using the degree centrality function from the *igraph* package. It is important to remark that this is an auxiliary function developed only to get results in the desired format and to complete the workflow. The function pseudocode is explained in the Figure 12

If input object is a graph
Get degree centrality from the graph
Sort the degree centrality list
Cut the list to the first N nodes
Return List
Else
Error message
Return
End if

Figure 12 Pseudocode of the function getTopDegree

4.6. openGraph

This functions has two variants, the main one takes a graph, saves it and open it in Cytoscape, the variation from this function plot the top minimum value between six and the number of clusters as seen in Figure 13. The saving format is *.gml* as it is the format that share the package *igraph* and Cytoscape, there is no need to write the extension as if it is not or it not correct the function will place it automatically so it can be recognized by Cytoscape the pseudocode of both options is described in the following Figure 14.

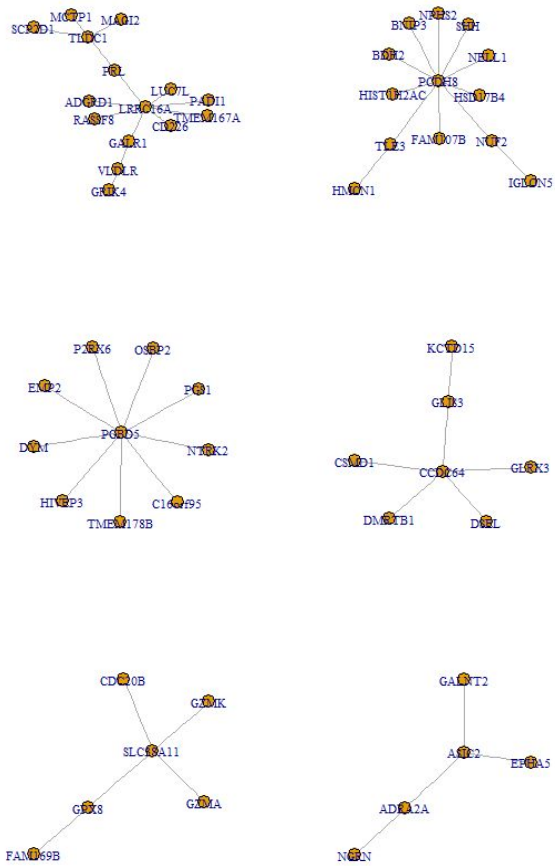

Figure 13 Plot representation from a sample graph using openGraph function

```
If input is a graph
   If Cytoscape is on the system path
      If filename is not ending on .gml
        Add .gml to filename
      End if
      Save graph
      Invoke to Cytoscape with the file path
   Else
      Send error message
   End if
Else
   Send error message
End if
```

```
If input is a graph
   If Cytoscape is on the system path
      If filename is not ending on .gml
        Add .gml to filename
      End if
      Save graph
      Initialize i to the min of 6 or the size
      Divide the plotting area into i/2:2
      For first i clusters
         Plot cluster on corresponding area
      End for
   Else
      Send error message
   End if
Else
   Send error message
End if
```

Figure 14 Pseudocode of both versions from the function opengraph

4.7. Workflow implementation in Galaxy

As part of the planned work for this project the port of the package to Galaxy is additionally one important part of it as it makes simpler to run experiments with no need to face difficulties of using the UNIX shell, although to port the R code to galaxy we need firstly to make it run in the shell. Other programing languages such as C or Java can run directly in the shell after being compiled, unfortunately this is not the case of R as is an interpreted language therefore it must be run using it specific tools, in this case the RScript command.

To make it work in the shell, the software should be adapted considering that there is not an enviroment where to store variables, therefore the first thing we must do is change the return values for functions to save the data, in this case the function *save* from the package *base* is enough to save R objects into a binary file, if what we want is to write a plain text file, as it happens in functions like *getTopDegree* we must use the function *write*.

Those are not the only changes that had to be done, the messages should be written through stdout or stderr so Galaxy can recognize it, if a message is sent through stderr then it will directly finish the module with an error status, also to adapt it to receive parameters through the console a header must be added, including a checking for the number of parameters, to the function file.

In order to insert the code in Galaxy a wrapper is needed, this is created by inserting all parameters with their class together with the command to run the software previously adapted to run in a console. Outputs need to be detailed in a XML file as well including their class.

Some complex classes which are not currently defined could be added to use it by adding it to a specific file for it.

It is very important that the input and output classes from different modules must be the same in order to allow them to be connected by the users.

After the porting is done the workflow must be created, a main workflow including creation of both graphs and both analysis and visualization of graphs has been designed as shown in Figure 15, alternatively there will be another workflow analyzing and visualizing only the gene graph as it is the final intention of the workflow. The user will anyhow have access to the modules to run them in an isolated way.

The user can create its own workflow even changing the order or using different modules to complement the results, this facility is one of the advantages of porting the package to a WMS, allowing the reutilization of code.

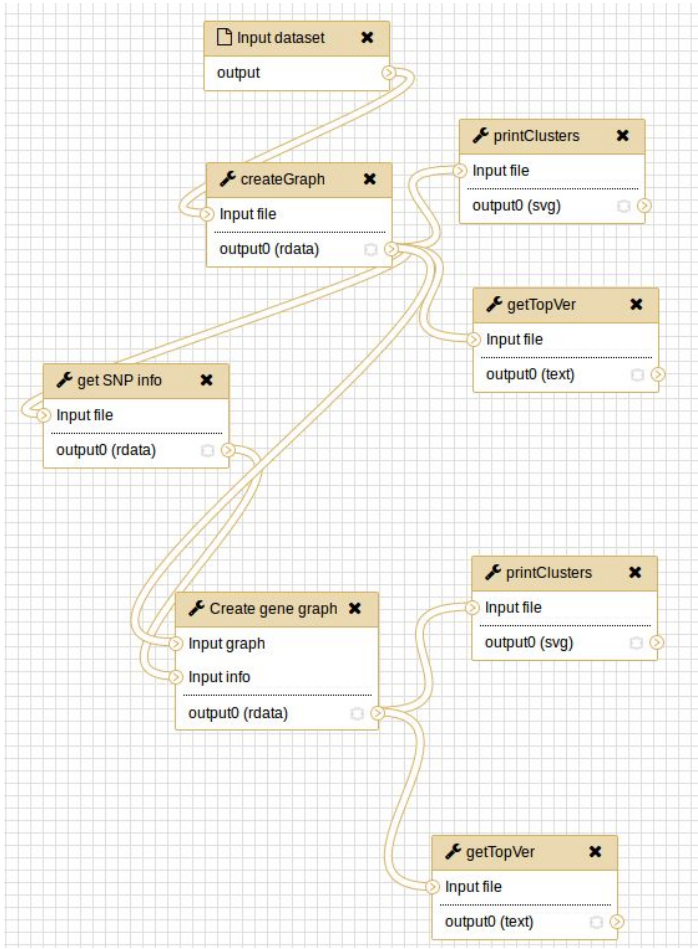

Figure 15 View of the complete workflow in the Galaxy workflow editor canvas.

5. Use case and validation

5.1. Dataset

The dataset used to perform this analysis is the one obtained during the study of Clarkson Disease [44], it consists of samples from 12 individuals with Clarkson disease, and 18 healthy control samples. Four of the patients were female, eight were male. In the control group, eight were female and ten were male. All of the patients were Caucasian, as it is the typical demographic of the disease, as were 16 of the 18 individuals in the control group. Genotyping was carried out with Affymetrix Genome-Wide Human Array 6.0 SNP chips. Prior to commencing the epistatic analysis of the dataset, single SNP association was carried out using the software package PLINK. Epistatic analysis was then carried out using the linear mixed model based FaST-LMM [45].

After more than three days of computing the exhaustive epistatic analysis an edge list of rsid pairs is obtained, this was shared by the Bitlab group so results can be replicated.

The dataset consists of an edge list of 695 SNP pairs in plain text with header. It will be provided digitally.

5.2. Software benchmarking

This section aims to replicate previously published results [44] to demonstrate the software is properly working

First step is to create a SNP graph, this will be done by using the function createGraphFromTable. To analyze and compare it to the published results we will use functions getTopDegree and openGraph. The list of the top 20 ranked SNPs is shown in Table 13. Running the function createGraphFromTable required less than a second as well as getTopDegree. This is a reasonable time as other methods depending on external services may take longer execution time depending on the connection of the user.

Then the association of the SNPs with the genes is performed using the function fillInfo and the result is shown in the Table 9.

Rank	SNP	Gene symbol	Chromosome	Rank	SNP	Gene symbol	Chromosome
1	rs11842523	PCDH8	13	11	rs7719321	MAP1B	5
2	rs4593336	NT5E	6	12	rs701170	PGBD5	1
3	rs17451360	SLC38A11	2	13	rs10191604	MYO3B	2
4	rs150533	LRRC16A	6	14	rs7420094	MMADHC	2
5	rs6927384	LRRC16A	6	15	rs12142665	PGBD5	1
6	rs1380237	MAP1B	5	16	rs7302874	CCDC64	12
7	rs6878132	MAP1B	5	17	rs11904398	MYO3B	2
8	rs7703322	MAP1B	5	18	rs12175817	GCLC	6
9	rs7704592	MAP1B	5	19	rs9382212	GCLC	6
10	rs7716699	MAP1B	5	20	rs10041715	RIOK2	5

Table 9 Result of the most significant SNPs and their related genes

Comparing results with those obtained from the original publication the first fourteen elements match perfectly with the first fourteen shown in the original table from the publication as seen in Table 10. The last changes, where we can see three of the last six ranked in the publication, do not mean that results are incorrect.

Differences in the results are due to the measure used and the size of the graphs. Degree centrality measure result is expressed in integers and therefore there can be many different nodes with same value. This problem increases when we are measuring small size graphs. In this case the results from rank fourteen have all degree centrality 6 being a total of 15 different nodes with this same score. Therefore amplifying the top to 29 would have all SNPs cited on the original study as can be observed on Table 11.

Rank	SNP	Gene symbol	Chromosome	Rank	SNP	Gene symbol	Chromosome
1	rs11842523	PCDH8	13	11	rs7719321	MAP1B	5
2	rs4593336	NT5E	6	12	rs701170	PGBD5	1
3	rs17451360	SLC38A11	2	13	rs10191604	MYO3B	2
4	rs150533	LRRC16A	6	14	rs7420094	MMADHC	2
5	rs6927384	LRRC16A	6	15	rs10079905	RIOK2	5
6	rs1380237	MAP1B	5	16	rs10065590	RIOK2	5
7	rs6878132	MAP1B	5	17	rs10079905	RIOK2	5
8	rs7703322	MAP1B	5	18	rs10476724	RIOK2	5
9	rs7704592	MAP1B	5	19	rs12175817	GCLC	6
10	rs7716699	MAP1B	5	20	rs9382212	GCLC	6

Table 10 Result provided in the publication of reference

Rank	SNP	Gene symbol	Chromosome	Rank	SNP	Gene symbol	Chromosome
15	rs12142665	PGBD5	1	23	rs10476724	NA	5
16	rs7302874	CCDC64	12	24	rs12652555	RIOK2	5
17	rs11904398	MYO3B	2	25	rs1560323	NA	5
18	rs12175817	GCLC	6	26	rs6556989	RIOK2	5
19	rs9382212	GCLC	6	27	rs6556992	RIOK2	5
20	rs10041715	RIOK2	5	28	rs10248777	SNX10	7
21	rs10065590	RIOK2	5	29	rs10238880	SNX10	7
22	rs10079905	RIOK2	5				

Table 11 Result of the SNPs with Degree centrality score six. Those showing NA did not find any gene within the flanking distance.

In order to validate the results obtained from the gene graph only the first 12 ranked genes will be displayed rather than the 20 in the original study. This is done due to

the decreasement of the size of the gene graph, being from rank 12 the score two or less, that makes a total of 42 genes with a score of 2.

As we can see in the comparison except of a switch between two genes, both scored 4 (SLC38A11 and TLDC1) and further changes in the order between those genes with score 3. The results are identical to those published as can be seen comparing Table 12 and Table 13.

Rank	Gene	Chromosome	Rank	Gene	Chromosome
1	PCDH8	13	11	EMP1	12
2	PGBD5	1	12	RIPK4	21
3	LRRC16A	6	13	CRYBA4	22
4	CCDC64	12	14	CHRD	3
5	TLDC1	16	15	TUSC1	9
6	SLC38A11	2	16	ETAA1	2
7	ASIC2	17	17	C16orf47	16
8	PSD3	8	18	GALR1	18
9	LHPP	10	19	TLE3	15
10	MAMLD1	X	20	MYO3B	2

Table 12 Results obtained on the reference paper

Rank	Gene	Chromosome	Score	Rank	Gene	Chromosome	Score
1	PCDH8	13	10	7	ASIC2	17	3
2	PGBD5	1	9	8	EMP1	12	3
3	LRRC16A	6	8	9	LHPP	10	3
4	CCDC64	12	5	10	MAMLD1	X	3
5	SLC38A11	2	4	11	PSD3	8	3
6	TLDC1	16	4	12	RIPK4	21	3

Table 13 Results obtained using the package epiR to replicate the results

To provide some biological meaning to the results obtained, an enrichment analysis must be performed. In this case we use WebGestalt, setting the enrichment type to GO enrichment, and then setting the test adjustment to Benjamini & Hochberg (BH) and at last defining the significance level as 0.1. We perform the analysis, obtaining the result after few seconds, stating that the two most affected functions are morphogenesis of epithelium as can be seen in Figure 16. When we go into depth by clicking on the boxes we can see that the genes related to these functions are PCDH8 and RIPK4, PCDH8 is the gene where more mutations has been detected so it makes sense that that gene is the cause of the illness, even more when according to some studies [46] state that the cause of the disease may be apoptosis of endothelial tissue, that means the death of the cell itself. This may be caused by a malfunction during the morphogenesis, therefore the regulatory mechanism from the cell would activate the cell's apoptosis.

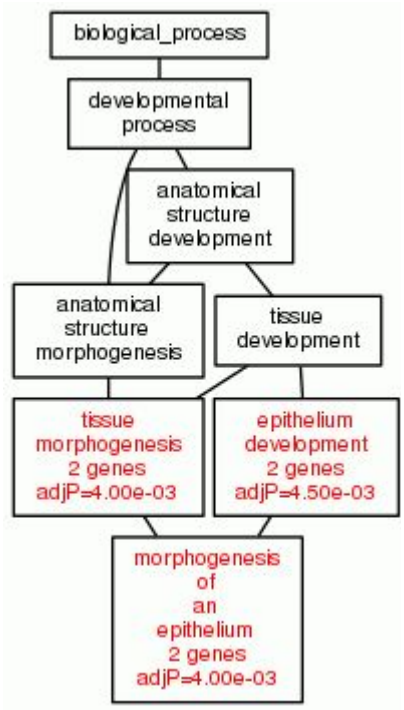

Figure 16 Result obtained from the enrichment analysis

5.3. Use case

The developed package will be used to perform a simple experiment to deduce whether the mutations affecting this biological function are located in the left flank or on the other hand on the right flank, To do this we will modify the flanking distance.

We will perform an identical analysis but this time we will set the left flanking distance to zero and right to 500,000 bp and vice-versa after performing this experiment

together with the enrichment analysis we would be able to determine whether the left flank is more decisive on this disease or on the other hand if the right one is more decisive. In order to perform this we only have to run once more the function replaceSNPwithGene from the previous study reassigning there the flanking distance, this avoid the internet connection and speed up the process.

After running the function we obtain the results shown in Table 14.

Left Flank			Right Flank		
Rank	Gene	Score	Rank	Gene	Score
1	PCDH8	10	1	PCDH8	11
2	LRRC16A	8	2	LRRC16A	8
3	PGBD5	6	3	PGBD5	6
4	GALNT2	4	4	CCDC64	4
5	SLC38A11	4	5	SLC38A11	4
6	TLDC1	4	6	EMP1	3
7	CCDC64	3	7	GALNT2	3
8	DNAH17	3	8	DNAH17	3

Table 14 Results obtained from performing the analysis with 45 mpb

The gene graphs obtained is shown in Figure 17 and in Figure 18, there we can visualize that PCDH8 is the gene most connected in the network, this provides a higher significance to that gene.

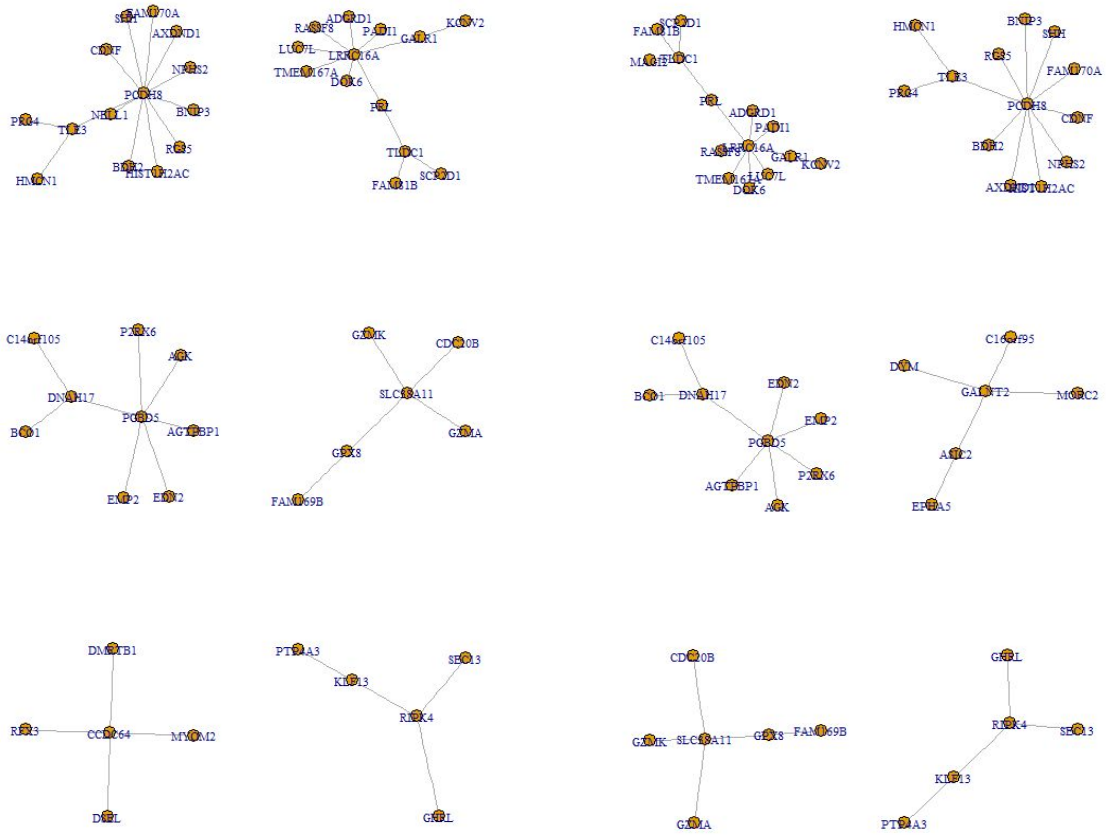

Figure 17 Gene graph obtained setting left flanking distance to zero in the two left columns and graph obtained setting right flanking distance to zero is in the right two columns

After this we will run a enrichment analysis with the genes obtained and same parameters that previous case, this has completely different results. This means that most of the mutations affecting the disease are spread equitably between both flanks. Another conclusion we can make is that most of the mutations affecting the most significant gene are located within the gene itself, from this results we can get that 8 of the

From this results we can conclude that in this case is was not necessary to set a high flanking distance, we can choose a smaller flanking distance while this is distributed between both flanks and not only on one flank.

6.1. Conclusions

The aim of this project has been to develop an accessible software for biologists including all necessary steps to complete a GWAS from an epistatic analysis. This objective has been achieved, providing accessibility and offering also the opportunity to publish the workflow made in Galaxy in a server to increase it even more.

The implemented software has proved its reliability reaching same results as previously published and validated studies which is very important as it shows that the developed software can reproduce results from other studies as well.

One of the main difficulties found during the developing time has surprisingly occurred when we were trying to mix two databases as annotations in the genome evolve continuously and so the databases keep updating accordingly making the references between them deprecated.

This software may help in reaching significant conclusions like the one reached on the last chapter, many other results can be obtained from this software applied by biologists.

Further steps in this project would go from including new metrics to evaluate graphs so they are showing an homogeneous result allowing to classify the nodes in a more reliable way without letting any subclassification in random factors or ordered by their first appearance. Also as seen in the use case it could be interesting to reduce the default flanking distance.

I personally believe that this project is a great closure of the degree I studied and even more for the mention in bioinformatics as it is working with one of the most valuable programing languages in the bioinformatic sector as well as the biological knowledge about annotations and the human genome together with the utilization of biological databases.

6.2. Conclusiones

El objetivo de este proyecto ha sido desarrollar un software accesible a los biólogos y que incluya todos los pasos necesarios para poder realizar un estudio completo a partir del análisis epistático. Este objetivo se ha marcado por la importancia de esta herramienta en varios estudios de tipo biológico. Además, atendiendo a los principales softwares utilizados por estos usuarios, se ha decidido implementar la herramienta desarrollada en el software de control de flujo de trabajo conocido como Galaxy, con esto se pretende aumentar la accesibilidad.

En base a los experimentos realizados, se ha demostrado la efectividad de la herramienta para el campo de identificación de genes relacionados con un factor de interés.

Una de las principales dificultades encontradas durante el desarrollo del proyecto ha sido la interrelación de información procedente de diferentes bases de datos. El problema radica en que las relaciones entre dos bases de datos no se encuentran actualizadas sino que hacen referencias en muchas ocasiones a bases de datos ya actualizadas a otra versión.

El software desarrollado puede influir en la identificación de las causas de enfermedades, como se ha demostrado en el último apartado de este documento, permitiendo incluso llegar a conclusiones más interesantes como el tipo de mutación causante de la enfermedad entre otros.

Una de las líneas principales de trabajo futuro en este proyecto debe ser implementar nuevas métricas para realizar la clasificación y ordenación de los nodos dentro del grafo. De este modo, no se deja una segunda clasificación de aquellos nodos con el mismo valor en manos de órdenes alfabéticos u ordenados por su primera aparición. Otra línea de desarrollo podría ser reducir la distancia lateral hasta dar con un punto óptimo, a ser posible que se calcule dinámicamente dependiendo de los datos de entrada.

Este proyecto se ha diseñado con la intención de implementar características de todas las competencias que un bioinformático debe poder desarrollar. Entre ellas se encuentra el uso de lenguajes de programación habituales en el ámbito, como es R, además de la integración de conceptos e información relacionada con temas biológicos, como ha sido en este caso el uso de anotaciones de genomas y la integración de información procedente de diferentes bases de datos biológicas.

7. References

[1] Metzker, M. L. (2010). Sequencing technologies—the next generation. *Nature reviews genetics*, *11*(1), 31-46.

[2]Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, *270*(5235), 467.

[3] Hancock, D. B., Romieu, I., Shi, M., Sienra-Monge, J. J., Wu, H., Chiu, G. Y., ... & Raby, B. A. (2009). Genome-wide association study implicates chromosome 9q21. 31 as a susceptibility locus for asthma in mexican children. PLoS Genet, 5(8), e1000623.

[4] Easton, D. F., Pooley, K. A., Dunning, A. M., Pharoah, P. D., Thompson, D., Ballinger, D. G., ... & Wareham, N. (2007). Genome-wide association study identifies novel breast cancer susceptibility loci. Nature, 447(7148), 1087-1093.

[5] Koschinsky, M. L., Boffa, M. B., Nesheim, M. E., Zinman, B., Hanley, A. J. G., Harris, S. B., ... & Hegele, R. A. (2001). Association of a single nucleotide polymorphism in CPB2 encoding the thrombin‐activable fibrinolysis inhibitor (TAFI) with blood pressure. Clinical genetics, 60(5), 345-349.

[6] Stoneking, M. (2001). Single nucleotide polymorphisms: From the evolutionary past...*Nature*, *409*(6822), 821-822.

[7] Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., ... & Sham, P. C. (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, *81*(3), 559-575.

[8] Team, R. C. (2013). R: A language and environment for statistical computing.

[9] Racine, J. S. (2012). RStudio: A Platform‐Independent IDE for R and Sweave. *Journal of Applied Econometrics*, *27*(1), 167-172.

[10] Hales, K., & Lavery, M. (1991). *Workflow management software: the business opportunity*. Ovum Ltd..

[11] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., ... & Li, P. (2004). Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, *20*(17), 3045-3054.

[12] Kallio, M. A., Tuimala, J. T., Hupponen, T., Klemelä, P., Gentile, M., Scheinin, I., ... & Korpelainen, E. I. (2011). Chipster: user-friendly analysis software for microarray and other high-throughput data. *BMC genomics*, *12*(1), 1.

[13] Reich, M., Liefeld, T., Gould, J., Lerner, J., Tamayo, P., & Mesirov, J. P. (2006). GenePattern 2.0. *Nature genetics*, *38*(5), 500-501.

[14] Goecks, J., Nekrutenko, A., & Taylor, J. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, *11*(8), 1.

[15] Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., ... & Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks.*Genome research*, *13*(11), 2498-2504.
[16] Hiersche, M., Rühle, F., & Stoll, M. (2013). Postgwas: advanced GWAS interpretation in R. *PloS one*, *8*(8), e71775.
[17] Ihaka, R., & Gentleman, R. (1993). R project. *URL http://www. r-project. org*.
[18] Edwards, D., Forster, J. W., Chagné, D., & Batley, J. (2007). What Are SNPs?. In *Association mapping in plants* (pp. 41-52). Springer New York.
[19] Sherry, S. T., Ward, M., & Sirotkin, K. (1999). dbSNP—database for single nucleotide polymorphisms and other classes of minor genetic variation. *Genome research*, *9*(8), 677-679.
[20] 1000 Genomes Project Consortium. (2010). A map of human genome variation from population-scale sequencing. *Nature*, *467*(7319), 1061-1073.
[21] Wang, Z., & Moult, J. (2001). SNPs, protein structure, and disease. *Human mutation*, *17*(4), 263-270.
[22]Wang, K., Li, M., & Bucan, M. (2007). Pathway-based approaches for analysis of genomewide association studies. The American Journal of Human Genetics, 81(6), 1278-1283.
[23] Sun, X., Lu, Q., Mukherjee, S., Crane, P. K., Elston, R., & Ritchie, M. D. (2014). Analysis pipeline for the epistasis search–statistical versus biological filtering. *Frontiers in genetics*, *5*, 106.
[24] Kogelman, L. J., & Kadarmideen, H. N. (2014). Weighted Interaction SNP Hub (WISH) network method for building genetic networks for complex diseases and traits using whole genome genotype data. *BMC systems biology*, *8*(2), 1.
[25] Edwards, A. M., Isserlin, R., Bader, G. D., Frye, S. V., Willson, T. M., & Frank, H. Y. (2011). Too many roads not taken. *Nature*, *470*(7333), 163-165.
[26] Upton, A., Trelles, O., & Perkins, J. (2015). Epistatic analysis of clarkson disease. Procedia Computer Science, 51, 725-734.
[27] Goecks, J., Nekrutenko, A., & Taylor, J. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome biology, 11(8), 1.
[28] Northeastern University. 2016. Why you should learn R. [ONLINE] Available at:http://www.northeastern.edu/levelblog/2016/05/17/why-learn-r-in-2016/. [Accessed 13 September 2016].
[29] Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., ... & Hornik, K. (2004). Bioconductor: open software development for computational biology and bioinformatics. Genome biology, 5(10), 1.
[30] Omic Tools. 2013. GWAS analysis. [ONLINE] Available at: https://omictools.com/gwas-category. [Accessed 19 September 2016].
[31] Wan, X., Yang, C., Yang, Q., Xue, H., Fan, X., Tang, N. L., & Yu, W. (2010). BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies. The American Journal of Human Genetics, 87(3), 325-340.

[32] Zhang, Y., & Liu, J. S. (2007). Bayesian inference of epistatic interactions in case-control studies. Nature genetics, 39(9), 1167-1173.
[33] Calle, M. L., Urrea, V., Malats, N., & Van Steen, K. (2010). mbmdr: an R package for exploring gene–gene interactions associated with binary or quantitative traits. Bioinformatics, 26(17), 2198-2199.
[34] Lippert, C., Listgarten, J., Davidson, R. I., Baxter, J., Poon, H., Kadie, C. M., & Heckerman, D. (2013). An exhaustive epistatic SNP association analysis on expanded Wellcome Trust data. Scientific reports, 3, 1099.
[35] Schüpbach, T., Xenarios, I., Bergmann, S., & Kapur, K. (2010). FastEpistasis: a high performance computing solution for quantitative trait epistasis.Bioinformatics, 26(11), 1468-1469.
[36]Huang, D. W., Sherman, B. T., & Lempicki, R. A. (2009). Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources.Nature protocols, 4(1), 44-57.
[37] Zhang, B., Kirov, S., & Snoddy, J. (2005). WebGestalt: an integrated system for exploring gene sets in various biological contexts. Nucleic acids research, 33(suppl 2), W741-W748.
[38] Freeman, L. C. (1978). Centrality in social networks conceptual clarification. Social networks, 1(3), 215-239.
[39] Okamoto, K., Chen, W., & Li, X. Y. (2008, June). Ranking of closeness centrality for large-scale social networks. In International Workshop on Frontiers in Algorithmics (pp. 186-195). Springer Berlin Heidelberg.
[40] Freeman, L. C. (1977). A set of measures of centrality based on betweenness. Sociometry, 35-41.
[41] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., ... & Harris, M. A. (2000). Gene Ontology: tool for the unification of biology.Nature genetics, 25(1), 25-29.
[42] NCBI2R, M. S. NCBI2R-An R package to navigate and annotate genes and SNPs. R package version,1(4).
[43] Yates, A., Akanni, W., Amode, M. R., Barrell, D., Billis, K., Carvalho-Silva, D., ... & Girón, C. G. (2016). Ensembl 2016. Nucleic acids research,44(D1), D710-D716.
[44] Upton, A., Trelles, O., & Perkins, J. (2015). Epistatic analysis of clarkson disease. Procedia Computer Science, 51, 725-734.
[45] Listgarten, J., Lippert, C., & Heckerman, D. (2013). FaST-LMM-Select for addressing confounding from spatial structure and rare variants. Nature Genetics,45(5), 470-471.
[46] Assaly, R., Olson, D., Hammersley, J., Fan, P. S., Liu, J., Shapiro, J. I., & Kahaleh, M. B. (2001). Initial evidence of endothelial cell apoptosis as a mechanism of systemic capillary leak syndrome. CHEST Journal, 120(4), 1301-1308.