Università degli Studi di Milano - Bicocca

Department of Informatics, Systems and Communication

Master Course in Data Science

A Bayesian Networks approach to Epistasis Detection

Supervisor: Prof. Fabio Stella

Co-Supervisor: Prof. Kristel Van Steen

Master Thesis by:
Edoardo Gervasoni
Nr. 790544

Academic Year 2019-2020

Contents

Introduction

In this work I present my approach to one of the most important and challenging problems that the fields of computational biology and genetics are now facing: *Epistasis*. This phenomenon takes place when the effect on an allele at a genetic variant depends either on the presence or absence of another genetic variant. The detection of Epistasis implies several issues, mainly statistical and computational, given that genomic datasets are characterized by very high dimensionality. Nonetheless, its study is fundamentally important to understand both the structure and function of genetic pathways and the evolutionary dynamics of complex genetic systems, and to discover the causes of genetic-related diseases. In fact, comprehending epistatic interactions may be the key to understand complex diseases, such as Alzheimer's disease, diabetes, cardiovascular diseases, and tumors.

In my work I addressed the problem of Epistasis detection thanks to a powerful statistical tool which I found very interesting and stimulating: Bayesian Networks. These are graphical models that are widely employed in causal inference, since they can discover and express the relationships between the variables considered. For example they are used in epidemiology in order to represent the probabilistic links between diseases and symptoms, while in our case they are used for discovering relationships between genetic loci and phenotype. Their most notable feature is that they represent causality in a very simple and direct way.

I carried out my project while being part of a multidisciplinary research group of the University of Liège comprising statisticians, data scientists, geneticists and computational biologists, with whom I collaborated in order to develop an approach to epistasis detection. My contribution can be summarized in these fundamental points:

- I extensively searched for scientific literature concerning epistasis in general, the state-of-the-art of its detection techniques, and a wide variety of models and strategies that address the problem. I analyzed and summarized about 40 different articles, from which I selected the most promising ones, considering their clearness, the quality of the additional material eventually available and the Bayesian Network approach that I decided to pursue. These papers are mentioned throughout the thesis and can also be found in the final *Bibliography*.
- I designed and developed an analysis framework employing the theoretical ideas and the procedures described in the selected papers. An important part of my approach consists of a modified version of Epi-GTBN [1], an epistasis identification

model which is based on Bayesian Networks. In chapters 2, 3 and 4 is possible to find a detailed account about the models and the methodology utilized.

- I overcame several practical issues in the implementation of the algorithms, which were mainly caused by the typical high dimensionality of the data ($\sim 10^4$ features involved). Moreover, I fixed a critical bug in Epi-GTBN code, which didn't allowed to handle too many columns at once. Hence, I created a new version of the model which can be found at `https://github.com/EdoardoGerva/Thesis`. A description about these arguments can be found in chapter 5.

- Finally, a remarkably challenging part consisted in testing my approach on real *Inflammatory bowel disease* data. They have been provided by the BIO3 laboratory of the University of Liège and are currently the subject of several research studies, since still little is known about the causes of these conditions. Over the course of all my work, I presented the methods and the results that were gradually achieved in periodic meetings with all the members of the research group. They also provided me with external contributions (such as suggestions and results from other algorithms) that helped me to develop my framework. The final results obtained are shown and discussed in chapter 6.

Chapter 1

Epistasis

Genome Wide Association Studies, **GWAS**, are studies that researchers and scientists use for identifying small variations of genetic markers in the genome-wide set of a sample of individuals that can determine different outcomes at a phenotypic level. One of the most popular classes of genetic markers are *Single Nucleotide Polymorphisms* (SNPs). Each SNP represents a difference in a single nucleotide at a specific position in the genome (hence the name *locus*), and it allows comparison of allelic frequencies -the frequencies of the different forms in which a gene can appear- within a sample of *cases* and *controls*. When SNPs occur within a gene or in a regulatory region near it, they may play a more direct role in disease by affecting the gene's function. Although most SNPs have no effect on health or development of an organism, some of these genetic differences have proven to be very important in the study of human health. Researchers have found SNPs that may help predict an individual's response to certain drugs, susceptibility to environmental factors such as toxins, and risk of developing particular diseases. SNPs can also be used to track the inheritance of disease genes within families.

In the context of classical genetics, the standard approach to GWAS consisted in testing SNPs one by one for statistical association with the disease, considering them to have independent effects on the phenotype. Through this method, only additive effects were considered, and the results were often not as appealing as expected [2]. This is because, analyzing one locus at time, only a little part of the genetic variance explained the phenotype, and the remaining part was referred to as *missing heritability*. If we consider 2 loci l' and l'', the additive approach can be described by the model:

$$y = b_0 + b_1 x' + b_2 x'' + \epsilon \tag{1.1}$$

where y is the gene expression, x' and x'' are the genotypes of the two loci, ϵ is a noise term and b are the coefficients.
However, after years of research it was understood that missing heritability is partly due to SNPs showing effects when they interact with one or more other variants, and in this context emerges the definition of Epistasis as a form of functional interaction between genetic loci that plays a fundamental role for the understanding of genotype-phenotype relationships.

Historically, the first definition of epistasis is due to Bateson (1909) [3], according to whom this was defined as the phenomenon where the effect of a gene on a phenotype is modified by one or several other genes. This asymmetrical interpretation was superseded

by the statistical model proposed in Fisher's works (1919) [4], where epistasis is defined as a *synergistic* effect of alleles of two or more loci when considering their contribution to a specific phenotype. This definition is based on the departure from additive effects of the different SNPs with regard to their contribution to the final outcome, and can be very useful in order to detect epistatic interactions with computational methods. This approach can be described by:

$$y = b_0 + b_1 x' + b_2 x'' + i x' x'' + \epsilon \tag{1.2}$$

where i is the interaction term that takes positive values in the presence of collaborative interactions and negative values in case of antagonistic interactions.

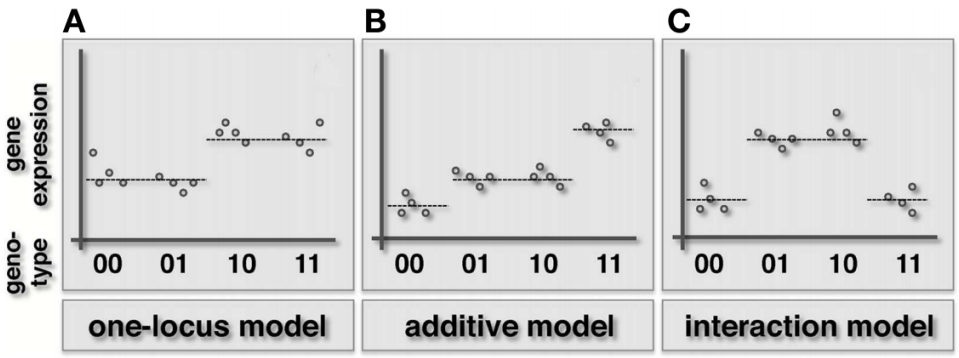

Figure 1.1: This figure represents the differences between the one-locus, additive and interaction models. On the x-axis there is the genotypic representation for each case, while on the y-axis we have the expression level. In the one-locus model only the genotype of the first locus determines independently the given phenotype (case A). In the additive model both loci contribute to the phenotype in an additive way (B). In the last case the effect of the two loci on the trait is non-additive and an epistatic interaction occurs (C).

Thus, epistasis detection has become an important field of research in human genetics: nowadays more complex models are studied, where combinations of genetic variants are examined in order to detect their association with a specific phenotypic outcome.

1.1 Challenges in Epistasis Detection

The main challenges related to the detection of epistasis can be categorized into 3 different groups: statistical, computational and interpretative challenges [5].

- ***Statistical Challenges***: one of the main problems in epistasis detection is the fact that genome wide data are typically characterized by high dimensionality; we are facing a *large p small n* problem, that occurs when the considered dataset has a large number of features, comparable or even greater than the number of observations, which results in a huge number of tests to be performed by the algorithms.

This leads to the necessity of a balancing between the false positive rate and the false negative rate obtained by the models applied.
Another statistical problem is represented by the presence of *minor allele frequencies*, or MAFs, which are the frequencies at which the second most common allele occurs in the given population. The MAFs can be very low, resulting in a sparseness of the data which could lead to the *curse of dimensionality*.

- ***Computational Challenges***: computational challenges are due to to the fact that the computational complexity of the problem grows linearly with the number of samples in the dataset, but exponentially with the interaction order. Therefore, the complexity in studying 2-loci interactions is quadratic, for 3-loci is cubic and so on. This implies that an exhaustive search for epistatic interactions of order 3 or higher would lead to a prohibitive computational load.

- ***Interpretative Challenges***: The last group of issues in epistasis detection concerns the final part of the analysis of the results, in which one tries to extract meaning at the biological level. In fact, the interpretation of statistical results from a biological point of view is not straightforward and not always the relationships found with statistical models correspond to actual interactions between SNPs.

1.2 Approaches to Epistasis Detection

In general, among the several strategies that can be exploited in order to identify epistatic effects, two main types can be distinguished: Exhaustive and Non-Exhaustive approaches.

1.2.1 Exhaustive Approaches

The models belonging to this class of methods are designed to detect mainly 2-loci epistasis, since they are usually not scalable enough to search for higher-order interactions, as I already explained in the previous section. One on the most common exhaustive approach to epistasis detection is represented by parametric regression models, which rely on strong assumptions about the probability distribution generating the data, and that perform badly when these assumptions are proved to be incorrect. In this category we can find logistic regression or penalized regression such as **SCAD** (smoothly clipped absolute deviation) or **LASSO** (least absolute shrinkage and selection operator) [6].

An important exhaustive model is the renowned **MDR** (Multifactor Dimensionality Reduction), a reference in the epistasis detection field, and usually used as benchmark for the evaluation of other strategies [7]. MDR proceeds by performing a 10-fold cross validation for training and testing the model in order to discriminate between low-risk and high-risk interactions. Despite being a rather flexible method which can also be extended to higher-order relations, it remains a *brute-force* approach which lacks of scalability and that cannot handle more than a few hundred of SNPs.
BOOST, Boolean operation-based testing and screening, is another model that performs an exhaustive search of all potential 2-loci relationships [8]. The data are binarized, and then the interactions are represented with contingency tables that are used to calculate log-likelihood ratios for evaluating the entity of the effects. Since BOOST heavily relies on the construction on contingency tables, it is particularly sensitive to low levels

of MAFs: in this case sparse tables will be generated, which will hamper the detection power of the model. BOOST is also particularly sensitive to type I errors.

Finally, it is worth mentioning the family of **ReliefF** filter methods which includes, in addition to RelieF itself, **TuRF** (Tuned ReliefF), **SURF** (Spatially Uniform ReliefF), **ECRF** (Evaporate Cooling ReliefF) and finally data-integration techniques such as **BioGRID** or **Biofilter** [9] [10] [11].

1.2.2 Non-Exhaustive Approaches

This type of models includes Machine Learning and combinatorial optimization methods. These approaches are non parametric, and usually make use of heuristics in order to scan for 2-loci or even higher order interactions. However, such non-exhaustive models can be affected by overfitting issues. In this class of algorithms we can find **Random Forest**, where a forest of classification trees is grown. In the case of GWAS data, the leaves of the trees represent SNPs, and at each step the algorithm looks for the predictor variable that optimally discriminates the population, so that a grown tree is a classifier which represents the set of features that allows the prediction of the phenotype of interest.

Another important class of non-exhaustive algorithms for the detection of epistasis is represented by Bayesian networks. These models consist of two components: a graphical one, called *Directed Acyclic Graph* (DAG), where the variables from our data are represented by nodes and the relationships between them by directed arrows, and a probabilistic one, which is the probability distribution associated with each node of the DAG. A remarkable Bayesian network model is **BEAM** (Bayesian Epistasis Association Mapping) [12]. Used as benchmark for other Bayesian network approaches, it is based on a *Monte Carlo Markov Chain algorithm* (MCMC), where each SNP is tested iteratively conditional on the other markers in order to obtain its posterior probability of association with the phenotype. However, this model suffers from the fact that it is not very scalable with modern GWAS data. Other common bayesian approaches are based on the *Markov blanket* method. Here the goal is to find the set of SNPs conditioned on which the class variable is independent from all the others in the dataset. A more detailed explanation of bayesian models is presented in the next chapter.

Finally, other examples of non-exhaustive approaches are Ant Colony Optimization algorithms, notably **AntEpiSeeker** [13], and Computational Evolution Systems, where a program is grown like a biological system. However also this last strategy is not scalable with modern GWAS datasets containing hundreds of thousands of SNPs.

1.3 Canalization

This section aims to explain the probable origin of the epistasis between gene loci, the **Canalization**. This term was coined by Conrad Hal Waddington in 1942 [14], to describe the robustness of phenotypes to perturbation, i.e. it represents a measure of the ability of a population to produce the same phenotype regardless of the variability of the environment (*environmental canalization*) or the variability of the genetic background (*genetic canalization*). Thus, in response to these variations, the modifications at a phenotypic

level are shielded, and canalized genotypes maintain a cryptic potential for expressing particular effects, which are only uncovered under particular decanalizing conditions. In this way, evolution has favored the creation of complex and robust systems resistant to perturbations, such that in biology is fairly common the observation of protein-protein or gene-gene interaction networks presenting redundant relationships, making these systems resistant to possible modifications. Hence, a particular disease state would be due to the accumulation of mutations, to the point of overcoming this resistance to the manifestation of the effects.

We can therefore understand why there is such a complex system of genetic interactions that contributes to a certain phenotypic state -precisely, epistasis- and why a large number of variations have just a little and hard to detect impact on external outcomes.

Chapter 2

Bayesian Networks

Bayesian networks are powerful and very intuitive tools for knowledge representation, inference, prediction and classification. The history of Bayesian Networks dates back to the works of Judea Pearl and others in the late 1980s [15], for which Pearl was given the Turing Award in 2011. Since then, Bayesian networks have evolved to become a fundamental part in the field of Data Science and are used in a wide range of domains, notably medicine and molecular biology.

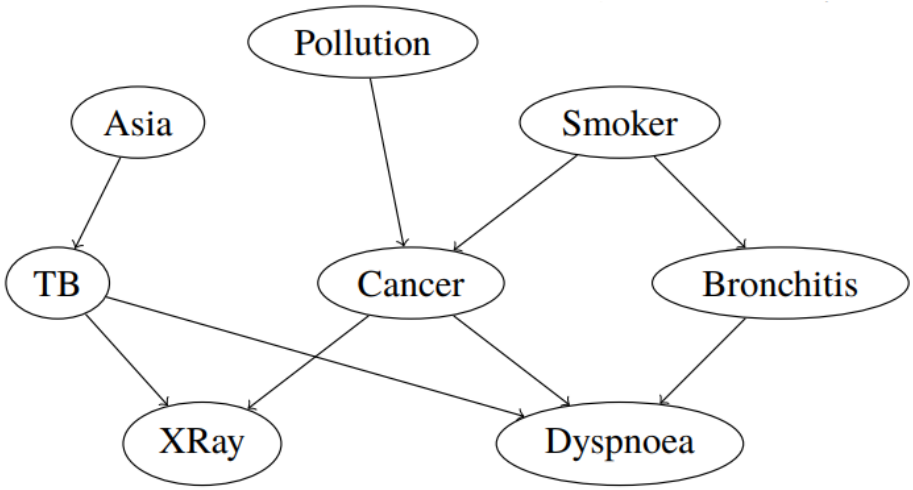

Figure 2.1: A graphical representation of a simple Bayesian network.

Bayesian networks belong to the family of probabilistic graphical models and interpolate between the *Optimal Bayes*, an optimal classifier where every variable is directly linked to every one else, and the *Naive Bayes*, based on an opposite approach where every explanatory variable is conditionally independent.

These graphical structures are used to represent knowledge about an uncertain domain, and combine principles from graph theory, probability theory, computer science. They

recently became very popular in fields such as statistics, machine learning, and artificial intelligence. As we already said, Bayesian networks consists of two parts: a DAG and a probability distribution. The graphical part itself, the DAG, includes the set of nodes (vertices) and the set of directed edges. The nodes represent random variables and are drawn as circles labeled by the variable names. The edges represent direct dependencies among the variables and are drawn by arrows between nodes. In particular, an edge from node X_i to node X_j represents a statistical dependence between the corresponding variables. Thus, the arrow indicates that a value taken by variable X_j depends on the value taken by variable X_i, i.e. that the variable X_i *influences* X_j. Node X_i is referred to as *parent* of X_j, while X_j is referred to as *child* of X_i. An extension of these genealogical terms is commonly used to define the sets of *descendants* -the nodes that can be reached on a direct path from the considered node- and the *ancestors* —those from which the node can be reached on a direct path-. By convention, each node is a trivial ancestor and descendant of itself. Finally, the structure of the DAG being acyclic ensures that the principle of cause and effect between the nodes is not violated. Such a condition is of vital importance to the factorization of the joint probability of a collection of nodes[16][17].

Given a Bayesian network described by a DAG G and a probability distribution P, G and P satisfy the so called *Markov condition*: each node in G is conditionally independent of the set of all its nondescendent nodes in G given the set of all its parents. This is equivalent to the following theorem:

Theorem 1 (Markov condition)**.** *Given a set of variables $V = X_1, ..., X_n$, their probability density $P(x_1, ..., x_n)$, and a DAG G, let $pa_G(x_i)$ denote the set of all parents of X_i in G. Then (G,P) satisfies the Markov condition, and therefore is a Bayesian network, if and only if P is said to* ***factorize*** *according to G, that is:*

$$P(x_1, ..., x_n) = \prod_i P(x_i|pa_G(x_i)) \tag{2.1}$$

If X_i has no parents, its local probability distribution is said to be *unconditional*, otherwise it is *conditional*; if the variable represented by a node is *observed*, then the node is said to be an evidence node, otherwise the node is said to be hidden or latent.

The Markov condition is used to reduce, sometimes significantly, the number of parameters that are required to characterize the joint probability distribution of the variables, thus providing an efficient way to compute the posterior probabilities given the evidence. In fact, in this way the parameters are described in a manner which is consistent with the Markov property, so that the conditional probability distribution at each node depends only on its parents. For discrete random variables, the conditional probability is often represented by tables listing the local probabilities taken by the considered node, with respect to every combination of possible probability value taken by its parents. The joint distribution of a collection of variables can be uniquely determined by these local conditional probability tables.

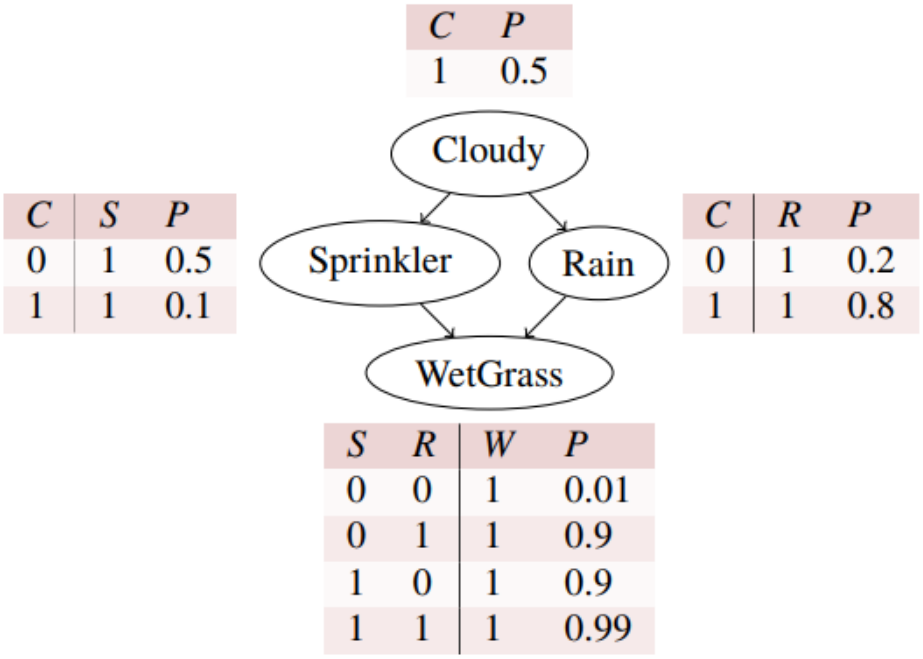

Figure 2.2: Another example of Bayesian network. The probability tables for each node are represented.

The figure above allows us to illustrate some of the characteristics of the Bayesian networks. It represents a classical example of BN, known as the sprinkler network. It can be used in order to predict whether the grass will be wet or not on the following information: if it is cloudy, if the the sprinkler is on or if it is raining. The conditional probability table of each variable is displayed besides the corresponding node. Here, *Cloudy* is the parent of both *Sprinkler* and *Rain*, and both of them are parents of *WetGrass*. Following the BN independence assumption, two independence statements hold in this case. Firstly, when *Sprinkler* and *Rain* are observed, the variables *Cloudy* and *WetGrass* are conditionally independent; in second place, *Sprinkler* and *Rain* are conditionally independent when *Cloudy* is observed. The conditional independence statement of the network provides a compact factorization of the joint probability distribution, that is:

$$P(C, S, R, W) = P(C)P(S|C)P(R|C)P(W|S, R) \tag{2.2}$$

where C denotes *Cloudy*, S denotes *Sprinkler*, R denotes *Rain* and W denotes *WetGrass*. It has to be noted that the BN form reduces the number of the parameters needed to estimate the model, which belongs to a multinomial distribution. In our case, the $2^4 - 1 = 15$ parameters that has to be evaluated in the Optimal Bayes case, are decreased to only 9. Such a reduction provides great benefits to inference, learning (i.e. parameter estimation) and, more generally, lowers the overall computational burdens.

2.1 d-Separation

A very important practical criterion that helps to investigate the structure of the joint probability distribution modeled by a BN is called **d-separation**. It allows us to graphically derive both the conditional independence and dependence statements that are implied by the Markov condition on the random variables. Given a DAG G and a probability density P that factorizes according to G, d-separation is a *sufficient but not necessary* criterion for conditional independence; that is, if two nodes X and Y are d-separated by a set of variables $\mathbf{Z}$, then $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ is guaranteed to hold, while if X and Y are not d-separated by $\mathbf{Z}$, then $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ may or may not hold. Let's define the concept of *path* in a Bayesian network as a sequence of variables in which each adjacent pair is connected by an edge. This definition differs from the classic graph-theoretical concept of a path, since it is allowed to move against arrow directions. The following theorem define how to apply d-separation.

Theorem 2 (d-Separation criterion)**.** *Consider a DAG G of a Bayesian network with variables $V = V_1, ..., V_n$. We say that $\mathbf{Z} \subseteq \mathbf{V} \setminus V_i, V_j$ d-separates V_i and V_j in G if, for every path $\pi = (V_i, V_{k_1}, ..., V_{k_n}, V_j)$, $n \geqslant 0$,*

- *π contains a* collider structure *$X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, such that X_i is not an ancestor of any node in $\mathbf{Z}$; or*
- *π contains a non-collider and:*
  - *$X_{i-1} \rightarrow X_i \rightarrow X_{i+1}$* (Markov chain structure)*, or*
  - *$X_{i-1} \leftarrow X_i \leftarrow X_{i+1}$* (reverse Markov chain structure)*, or*
  - *$X_{i-1} \leftarrow X_i \rightarrow X_{i+1}$* (fork structure)*,*

  where $X_i \in \mathbf{Z}$.

So, if $\mathbf{Z}$ d-separates X and Y in a DAG G as described above, then $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ in every probability density P that factorizes according to G[18].

2.2 Inference via Bayesian networks

Given a BN and its joint probability distribution in a factored form, it is possible to evaluate all inference queries by *marginalization*, which consists in discarding irrelevant variables by summing out over them. Two types of inference support are often considered: *predictive support* for the node X_i, based on evidence nodes connected to X_i through its parent nodes (also called *top-down reasoning*), and *diagnostic support* for the node X_i, based on evidence nodes connected to X_i through its children nodes (also called *bottom-up reasoning*).

A major problem regarding inference in Bayesian networks is that, even when considering binary variables, the joint probability distribution has size $O(2^n)$, with n number of nodes. Hence, summing over the probability distribution takes exponential time. In general, the full summation (or integration in the continuous case) over discrete variables is called *exact inference* and it is known to be an **NP-hard** problem: even for sparse models with, for example, $\leqslant 3$ parents, computing marginal probabilities remains a very

difficult challenge and no polynomial time algorithm is known. However, there exist some efficient methods to solve the exact inference problem in a restricted class of networks, known as *polytrees*, whose skeleton has one, and only one path between any pair of nodes. One of the best known algorithms of this kind is called *elimination ordering*, and consists in an ordered factorization of the probability distribution such that in each step, one variable is eliminated, and a function of only one variable is kept[19].

Another notable technique which also works for polytrees is the *message passing* algorithm that solves the problem in $O(n)$ steps (linear in the number of nodes). This method performs all computations directly on the *factor graph*, which is a representation of the factorization of a probability distribution. Each node in the factor graph is considered as a processor with the capabilities of sending and receiving messages (functions) along edges and processing messages by multiplication and addition. When considering Bayesian networks that are not polytrees, the factor graph will contain cycles and we will have to use *loopy message passing*, an extension of the message passing algorithm by Lauritzen and Spiegelhalter[20] that also works for graphs with cycles. This is an approximation algorithm, which is not even guaranteed to converge in most cases, but appeared to work well in practice. With loopy message passing, messages are passed in both directions along every edge of the factor graph.

There are other examples of approximate inference methods that were proposed in the literature, such as the *Monte Carlo sampling* that gives gradually improving estimates as sampling proceeds[21]. A variety of standard Markov chain Monte Carlo methods, including the *Gibbs sampling* and the *Metropolis–Hastings algorithm*, were also used for approximate inference[22]. Another method is the *variational model* that exploit the law of large numbers to approximate large sums of random variables by their means[17].

2.3 Learning

In a Bayesian network, the DAG encodes conditional independence relationships that must hold among the corresponding random variables. Several DAGs can encode exactly the same set of conditional independence relationships. Such structures are called *Markov equivalent* and form a so called *Markov equivalence class*, which can be uniquely represented by a completed partially directed acyclic graph (CPDAG). We refer to the operation of estimating the CPDAG of a Bayesian network from the data as *structure learning*. There exist three main families of structure learning algorithms: *constraint-based*, *score-based* and *hybrid* algorithms. The first type of models typically starts with a complete undirected graph and then prunes it by deleting all the arrows that do not satisfy the conditional independence test (that could be information based, ad-hoc or statistical), in order to obtain the final skeleton of the CPDAG. Subsequently it uses some specific rules trying to find the direction of the edges. Famous example of constraint-based methods are the *PC algorithm* and its variants. They have been widely applied to high-dimensional datasets since they were shown to be consistent in sparse high-dimensional settings where the number of variables is allowed to grow with the sample size and also because they scale well to sparse graphs with thousands of features[23].

Conversely, score-based algorithms use a score function (the same used in optimization

problems) to evaluate the actual network given a dataset. Starting from an initial solution which usually is an empty graph, they attempt to improve the accuracy going step by step towards the optimal result. *Greedy equivalence search* (GES) is a popular example of score-based methods which involve greedy search strategy. This kind of structure learning algorithms have proven to perform well in the conditions where the number of variables remains fixed and the sample size tends to infinity[24]. However, such models fail to find the global optimum in a high-dimensional scenario and they do not scale well to large graphs.

Finally, hybrid algorithms combine score-based with constraint-based methods, and they often use a greedy search on a reduced solution space in order to achieve computational efficiency. This is carried out by using conditional independence tests or variable selection methods. Common choices for the restricted search space are an estimated skeleton of the CPDAG (CPDAG-skeleton) or the conditional independence graph (CIG)[25]. Hybrid algorithms generally scale well with respect to the number of variables, but their consistency results are generally lacking even in the classical settings.

2.4 Markov Blanket

In the context of Bayesian networks, another important concept is the *Markov Blanket*. Given a set of variables V, the Markov blanket of a target variable T, $\mathbf{MB}(T)$, is the minimum set of variables that can make T independent from all other variables in V that do not belong to $\mathbf{MB}(T)$. In other words, all the variables that are not included in $\mathbf{MB}(T)$ are independent of the variable T conditional on the Markov blanket of T:

$$\forall\, X \in V \setminus \mathbf{MB}(T), \quad X \perp T \mid \mathbf{MB}(T) \tag{2.3}$$

The concept of Markov blanket is particularly relevant since it should represents the optimal set of variables to predict the value of T. Therefore, in order to reduce the computational cost of the model, any variable which is not in $\mathbf{MB}(T)$ can be ignored without significantly affecting the performance of the learned predictor.

2.5 Detecting Epistasis Using Bayesian Networks

In the previous chapter it was already mentioned that Bayesian networks are one of the non-exhaustive methods adopted in the search for interacting loci. Compared to other models, a great advantage of Bayesian networks is the fact that the causal relationships between features are immediately recognizable thanks to the oriented arrows drawn between the nodes of the DAG. Hence, it is possible to understand in a very simple way what are the causes of a given observed outcome. Therefore it is clear their effectiveness in causal inference, biology, epidemiology and also in the detection of epistatic effects, where they can highlight relevant connections between SNPs that can influence the phenotype. Moreover, they benefit from being able to handle fairly large datasets (hundreds of thousands of SNPs), building causal connections between objects directly extracting implicit knowledge, and the ability to manage data of different types and that contain noise or non-linear relationships. However, they also suffer of two main drawbacks, since learning algorithms are quite computationally expensive and they are also likely to fall

into a local minimum.

In addition to the aforementioned BEAM, there are a lot of Bayesian networks approaches that have been adopted in epistasis detection. **BNMBL** [26], Bayesian network minimum bit length, is a method where the structure is learned through a score-based algorithm according to the *Minimum Description Length principle*, for which the optimal results are those that minimize the number of bits to encode the model. In **EpiBN** [27] a *Branch-and-Bound* iterative procedure is exploited in order to learn the networks, that are subsequently evaluated with a score function that is made of two terms: one indicating how well the current structure fits the data and the other measuring how complex the Bayesian network is.
More recent than these models, **Epi-GTBN** [1], Epi Genetic Tabu Bayesian Network, performs network learning through a genetic algorithm, which is improved with a tabu search strategy. This method overcomes the typical issues of the application of Bayesian networks in GWAS, since the genetic algorithm is characterized by having a rapid global search and also tends to avoid falling into the local minimum, while tabu search helps to enhance the diversity of population and thus to obtain the global solution. Consequently, given its strengths, I opted for this technique in carrying out my work of epistasis detection. An in-depth explanation of this model is presented in the next chapter.

As regards the Markov blanket approach in the context of GWAS, they are used to avoid the time-consuming processes like the structure learning of a full Bayesian network, since they are able to identify the minimum set of markers that directly affects the phenotype. In the past years several proposals were made to efficiently learn an optimal Markov blanket from genome wide data. In the case of **DASSO-MB** [28], the Markov blanket is learned through a two-stages approach: in the first one (forward phase), relevant variables are added to the candidate Markov blanket, which is initially empty. The variables are selected considering those that show the strongest statistical dependence with the phenotype conditional on the SNPs already present in the blanket. In the second phase (backward phase) the algorithm removes false positives that were included in the previous step. However, this approach has the problem that in the forward phase, the first SNP added to the candidate Markov blanket is picked on the basis of a univariate test (since initially the blanket is empty), and the conditional independence tests in the remaining iterations will heavily rely on this inclusion. Thus, the detection of marker combinations when marginal effects are slight or nonexistent will be a major obstacle.
SMMB [29], Stochastic Multiple Markov Blanket, overcomes this problem by combining Markov blanket learning with stochastic exploration of groups of SNPs. In particular, here the forward phase includes a stochastic algorithm which takes into account potential epistatic interaction between SNPs, so that markers that display little or no marginal effect can be detected.

Finally, another important Markov blanket method is the **IAMB** (Incremental Association Markov Blanket) algorithm[30]. It discovers a unique Markov blanket of a target variable T, and it comprises two phases, the *growing phase* and the *shrinking phase*. During the growing phase, as many candidate nodes as possible that can constitute the $\mathbf{MB}(T)$ are collected. The shrinking step removes one-by-one the features that do not belong to the $\mathbf{MB}(T)$ by testing whether a variable X from $\mathbf{MB}(T)$ is independent of T given the remaining elements of $\mathbf{MB}(T)$. In order to do so, IAMB makes use of a

heuristic function $f(X, T|\mathbf{MB}(T))$, where X is the candidate variable, which returns a non-zero value for every variable that is a member of the Markov blanket, and typically measures the association between X and T. It is crucial that f is an informative and effective heuristic so that the set of candidate variables after phase I is as small as possible for two reasons: time efficiency, so not to spend time considering irrelevant variables, and sample efficiency since samples larger than what is absolutely necessary are not needed to perform tests of conditional independence. Generally, this operation is done using the information-theoretic heuristic function *CMI* (conditional mutual information), so that:

$$X \perp T \mid \mathbf{MB}(T) \implies X \notin \mathbf{MB}(T) \quad iff \quad CMI(X, T \mid \mathbf{MB}(T)) < k \tag{2.4}$$

where k is a predetermined threshold.

Alongside these methods, other Bayesian-related approaches to epistasis are represented by **Tree Augmented Naive Bayes** and **Chow-Liu Trees**. Tree augmented naive Bayes, or TAN, are models first described by N. Friedman in 1997 [31] and are characterized by a relaxation of the strong independence assumption between all variables typical of naive Bayes, and connections between attributes are allowed (*augmenting edges*). In this way a more flexible model is obtained, while maintaining the computational simplicity and robustness of the naive Bayes. Chow-Liu trees, on the other hand, are constructed through the computation of the *Mutual Information* between each pair of variables, which is used to build the skeleton of a tree. Then directions are added by choosing a feature as root and setting the direction of all edges to be outward from it. In the case of GWAS data these models can prove to be useful since, when learning the structure of the network, the algorithm chooses the attributes that are relevant for predicting the class [32][33][34]. In other words, the learning procedure performs a feature selection which can be helpful for detecting relevant SNPs.

Chapter 3

Epi-GTBN

Epi-GTBN, an approach to epistasis mining based on genetic Tabu algorithm and Bayesian networks, is a fairly recent strategy developed by Guo et al [1]. Epistasis detection is accomplished by constructing the network of gene loci for a specific phenotype. Given the characteristic issues of Bayesian networks when dealing with genome wide data, the structure learning task is carried out thanks to a genetic algorithm improved with tabu search. Genetic algorithms (GA) are *metaheuristics* that, taking inspiration from the process of natural selection, select chromosomes (individuals) that are more suitable for the environment to reproduce. After the construction of the first individual and the creation of the initial population, another generation is built thanks to the genetic operations of *selection*, *crossover*, and *mutation*. In this way, the algorithm evolves generation by generation, and finally converges to one of the most adaptable chromosomes. However, genetic algorithms tend to have the problem of falling into a local minimum, i.e. generating the same individual over the course of iterations, which undermines the diversity of the population. In fact, the selection step directly chooses the best individuals as parents for the subsequent generation, easily leading to a local solution.

3.1 Tabu Search

Tabu search is a widely known heuristic search algorithm, which exploits the memory function of the tabu list to avoid the production of the same result, thus ensuring diversification. Moreover, the tabu strategy can accept sub-optimal solutions in the search process and therefore has a higher climbing ability, since it allows to jump out of the local minimum and look for other regions of the solution space, greatly increasing the likelihood of obtaining better or global minima.

In order to solve the problems of the genetic algorithm mentioned above, Epi-GTBN applies the memory function of the tabu search to the crossover operator and the mutation operator. In this way, the performance of the model is improved and the optimal solution is found faster and more accurately. Furthermore, all the individuals on whom the tasks of the genetic algorithm are performed, are Bayesian networks and the search process occurs in the space of Directed Acyclic Graphs. The structures of the networks are represented by *adjacency matrices*: given a graph with vertex set $U = u_1, \ldots, u_n$, its adjacency matrix is a square $n \times n$ matrix A such that its element $A_{ij} = 1$ if there is an edge from u_i to u_j, and $A_{ij} = 0$ otherwise.

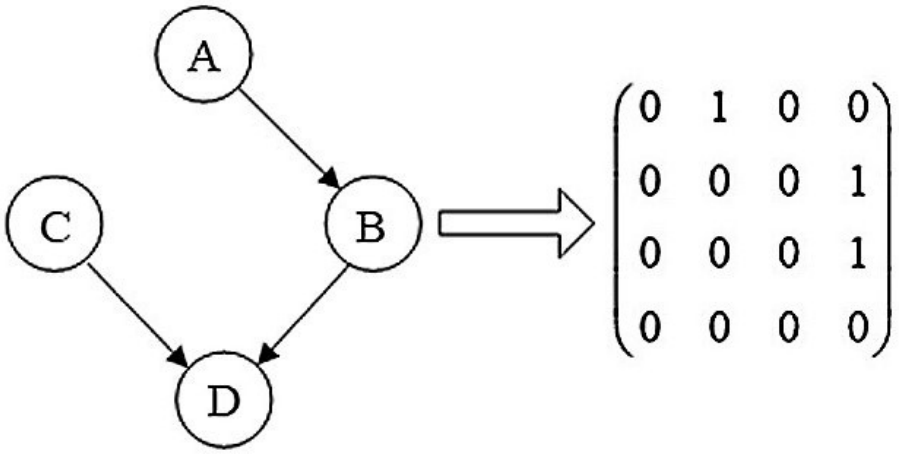

Figure 3.1: Matrix representation of a simple Bayesian network.

3.2 Genetic algorithm

The evolutionary mechanism of genetic algorithms is based on a fitness function, which plays a very important role considering that it guides the research process by evaluating the individuals. In Epi-GTBN this task is performed thanks to the function *Bayesian Information Criterion* or BIC, which allows to control the model complexity since it rewards simpler models with a lower number of parameters. The BIC is calculated as:

$$BIC \;=\; -2 \cdot ln(L) \;+\; k \cdot ln(n) \tag{3.1}$$

with L indicating the maximum likelihood of data given the network, k the number of free parameters, and n the sample size.

Here we present a brief description of the structure learning process.

- **Construction of the first individual:** firstly, *Mutual Information MI* is computed for all possible combination of 2 SNPs and the phenotype with the formula:

$$MI(Class|SNP_1, SNP_2) \;=\; H(Class) + H(SNP_1, SNP_2) - H(Class, SNP_1, SNP_2) \tag{3.2}$$

  where $Class$ represents the phenotype variable, and H is the entropy, defined as:

$$H(X) \;=\; -\sum_{x_i} P(x_i) log P(x_i) \tag{3.3}$$

  where x_i are the possible outcomes of the variable X with probability $P(x_i)$. In order to speed up the process, in Epi-GTBN the data are converted into binary Boolean data, and bitwise operations are used to calculate the mutual information directly. The node pairs are thus sorted according to the value of *MI* and then top-100 couples are extracted. If there are some SNPs that have not been included in the top-100, the first appearance of these nodes is selected in the remaining

pairs. Finally, the first individual is constructed according to the chosen couples. This step plays a very important role in achieving an optimal final result, since its outcome serves as the input for all subsequent operations of the algorithm.

- **Generation of the initial population:** the second individual is generated through adding, dropping or reversing an edge of the first network with the constraint of not creating a cycle. The next individual is constructed on the basis of the previous one in the same way as before, until the number of individuals reaches the population size defined by the user.

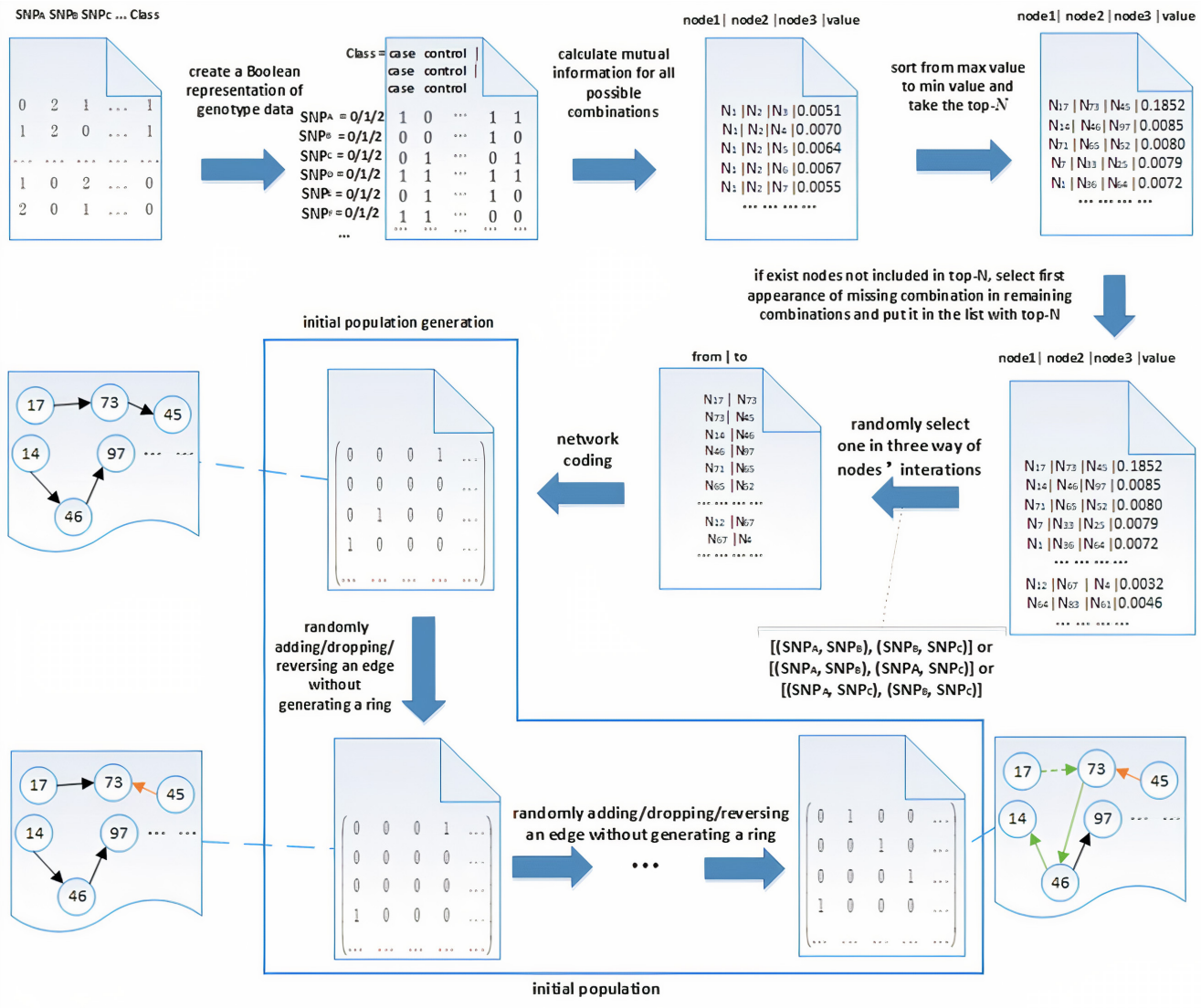

Figure 3.2: Process of generation of the first individual and the initial population.

- **Genetic manipulations:** when the construction of the initial population is over, the operation of *selection* takes place, with the aim of choosing a good individual to be the parent for the next generation in order to improve its quality. This task is

based on the principle of the *survival of the fittest* from the mechanism of natural selection, according to which the individuals who demonstrate greater adaptability will be those who will have a higher chance of transmitting their genetic heritage. Considering N total chromosomes, the k-th chromosome with fitting function f_k will have a probability of being selected equal to:

$$P_k = \frac{f_k}{\sum_{i=1}^{N} f_i} \tag{3.4}$$

with f_k computed thanks to formula 3.1. The second phase is the *crossover*, during which the offspring inherits the characteristics of its parents. Epi-GTBN performs a multi column crossover operation, where several columns are exchanged between individuals represented by contingency matrices, always under the constraint of not generating a ring structure. As we already said, during this phase the model can incur into the problem of generating the same offspring in subsequent generations. In order to avoid this stagnant situation, the algorithm is enhanced with the memory function of tabu search. For every generation, each new individual is compared with the networks already present in the tabu list. If the new descendant does not belong to the list, the algorithm will proceed to the generation of the next one and the individual will be stored. Otherwise the crossover operation will be performed repeatedly until a new network is obtained.

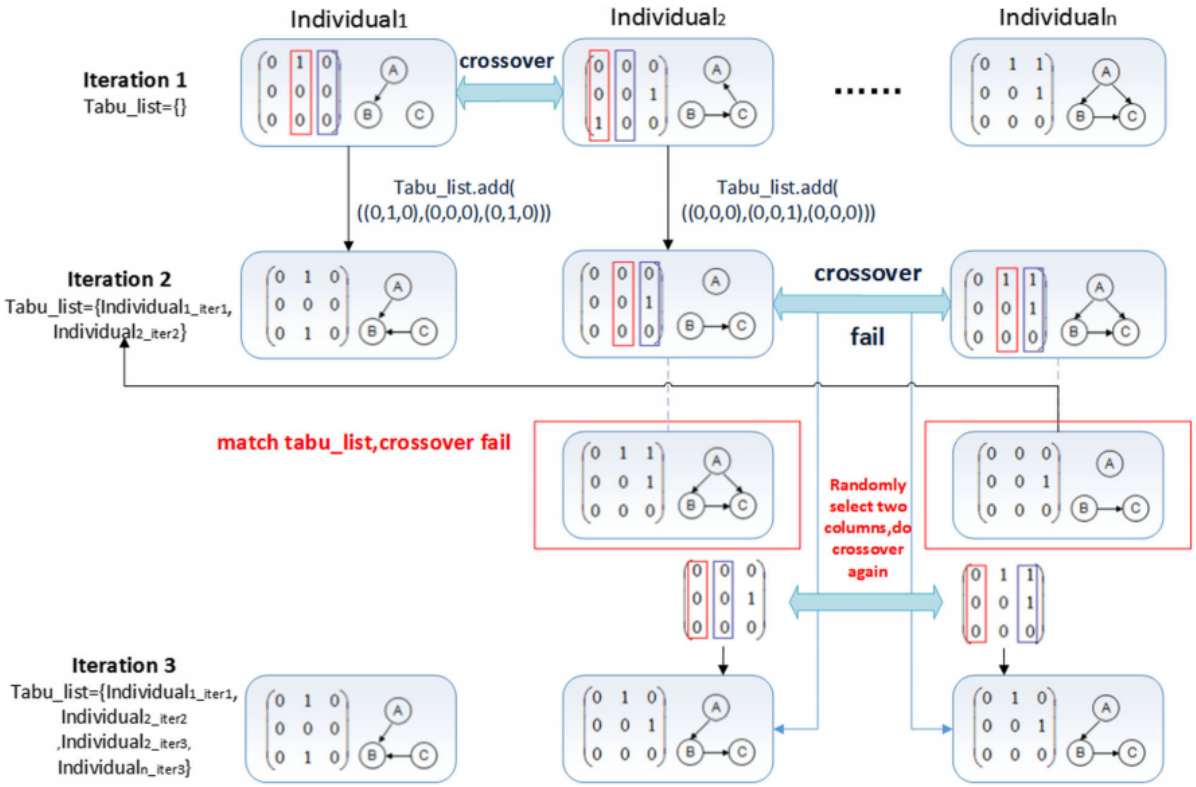

Figure 3.3: Tabu crossover operator.

When all the candidates are generated from their parents with the crossover operation, the *mutation* phase begins. Here, an individual from the population is selected randomly to change its structure with a certain mutation probability P_m. Since the standard mutation operator has strong randomness and may damage individuals with high fitness value, again, Epi-GTBN exploits the strength of tabu search: if the score of the mutated individual is lower than that of its parents, then it will be stored in the list, and new mutated networks will be generated until a better one will be obtained. In this way the loss of quality within the new generation is prevented, while guaranteeing the diversity of the population necessary for detecting a global solution.

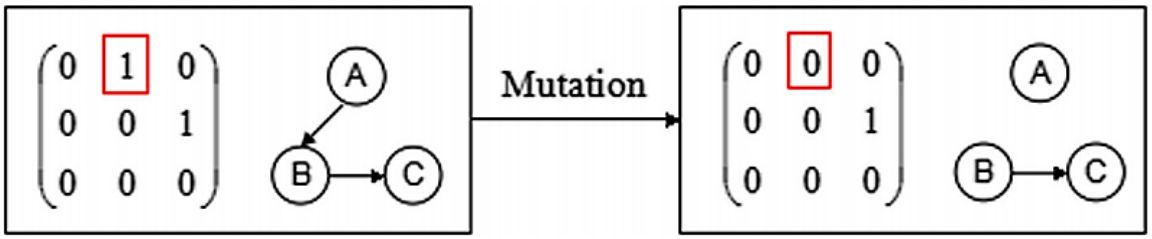

Figure 3.4: Mutation operator.

When the last operation ends, a new population is created and the algorithm will proceed again with the *selection*, *crossover* and *mutation* phases generation by generation until convergence is met, that is, when the fitness score of the best individual does not increase anymore.

3.3 Parameters

Epi-GTBN can be initialized with different parameters, which are:

- *max.iter*: maximum number of iteration of the GA algorithm if no convergence is met before
- *gtbn.population.size*: the number of individuals per generation
- *gtbn.crossover.pro*: the probability of crossover
- *gtbn.mut.pro*: the probability of mutation
- *gtbn.crossover.tabulist.length*: the number of elements to be considered in the tabu list

In order to select the most suitable values, several tests were performed and the advice by the authors of the paper was considered. Finally we set max.iter = 100, gtbn.population.size = 200, gtbn.crossover.pro = 0.7, gtbn.mut.pro = 0.008 and gtbn.crossover.tabulist.length = 125.

Chapter 4

Methodology

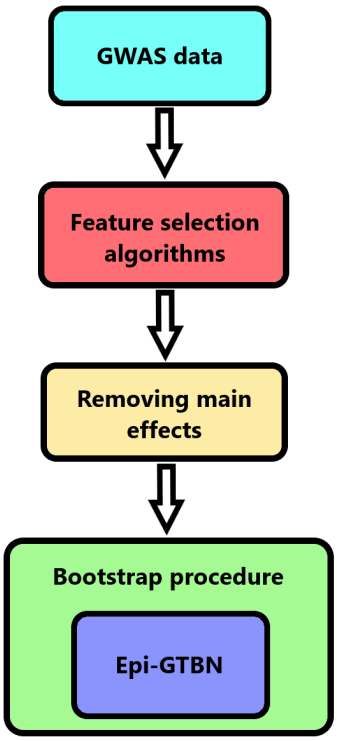

Figure 4.1: A chart representing the approach to the identification of epistatic interactions that we have developed.

The purpose of this work consisted in developing and testing on real data a new approach to Epistasis detection. In particular, it was decided to rely on Bayesian networks given their efficiency in managing GWAS data and the simplicity with which it is possible to extract knowledge in an intuitive way. In order to build our procedure, we opted to combine several techniques addressing this problem that have been found in literature. Going more in detail, we chose to employ Epi-GTBN as the core part of the approach, since it represents an improvement over the traditional Bayesian analysis methods and since it has been specifically developed for the identification of epistatic connections. Secondly,

we considered this model because it was among the most recent approaches that have been found. It was decided, in fact, to exclude methods antecedent to 2016, given that the time factor is fundamental in this field of research: techniques such as the aforementioned BEAM, DASSO-MB, BNMBL and EpiBN all aged very quickly with availability of increasingly larger genomic data sets that require ever greater computational capabilities. Furthermore, the Epi-GTBN article was the only one regarding Bayesian networks that presented the entire source code online and fairly well documented -the R library is available online at the link `http://122.205.95.139/Epi-GTBN/`, while the Github page is `https://github.com/Epi-GTBN/package-`. However, applying Epi-GTBN directly to the entire dataset was too computational burdensome, given the high number of SNPs and the limited material resources available to us. Thus, we took inspiration from the literature to carry out an initial filtering procedure.

4.1 Feature Selection

In general, feature selection methods can be grouped into three broad classes: *filters*, *wrappers*, and *embedded*. Filters are the simplest and least expensive methods. They build a ranking of the variables based on their relevance by detecting the intrinsic properties of features with respect to the class. They are fast and flexible, but they are also susceptible to noise and lose accuracy in case of too many irrelevant variables. On the other hand, wrappers work by evaluating a subset of features using a machine learning algorithm that employs a search strategy to look through the space of possible feature subsets, evaluating each subset based on the performance of the considered model. They are more precise than filters but they are also very computationally expensive. Finally embedded methods are a middle ground between filters and wrappers. They perform the feature selection process within the classification process of the algorithm itself. In other words, they carry out feature selection during the model training, which is why they are called embedded. Among these strategies we decided to focus on filters, given their simplicity and the low computational cost. As regards the wrappers methods, we have also tried to work with SMMB, but it proved to be too costly since the running time on our data exceeded the week.

The first approach that have been tried is the already cited **Relief**. Relief is a family of filters, which are characterized by the fact that they do not search through feature combinations, but rather exploit the concept of *nearest neighbours* to derive feature statistics that indirectly account for interactions. They are also fast in execution and flexible, since they assign feature weights that can be used not only to select top-K variables, but can also guide stochastic machine learning models such as evolutionary algorithms. The main idea behind Relief is to estimate feature importance according to how well explanatory variables can distinguish class values among instances that are close to each other. It cycles through N random training instances; for each cycle a target instance R_i is selected, its nearest neighbor with the same class (nearest hit) and its nearest neighbor with the opposite class (nearest miss) are identified and the vector of feature weights W is initialized. Then the weight of a feature A is updated if the feature value differs between the target instance R_i and either the nearest hit or the nearest miss. If the feature has a different value between R_i and nearest miss, its weight is increased since it is considered to be informative for the outcome, while in the opposite case, it is penalized. The

function used to compute the difference in value of the feature A between two instances is:

$$diff(A, I_1, I_2) = \begin{cases} 0 & \text{if } value(A, I_1) = value(A, I_2) \\ 1 & otherwise \end{cases} \tag{4.1}$$

where, in our case, I_1 is R_i and I_2 is either the nearest miss or the nearest hit.

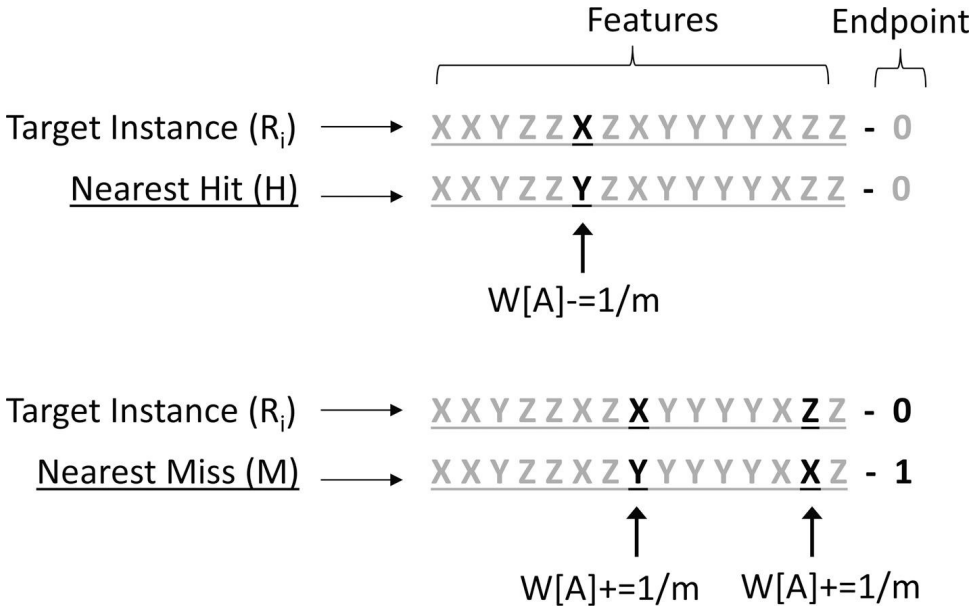

Figure 4.2: Representation of Relief updating the weight W for a given feature A considering R_i with its nearest hit and nearest miss.

In my work, it was chosen to use *ReliefF*, a modern version that has recently supplanted the original Relief method. This approach relies on user specified k number of nearest neighbours -instead of 1 as in the standard Relief-, can handle missing data and a multi-class target variable and N is equal to the total number of instances. ReliefF was initialized with the parameter k set to 10. The documentation of the python package is available here `https://epistasislab.github.io/ReBATE/using/#relieff`, while a more detailed description of the algorithm can be found in [9].

Besides ReliefF, another filter strategy that was adopted is the function **Episcan** of the homonymous R library. Episcan provides an efficient methods to scan for pairwise epistasis in case-control studies and its approach is adjusted from the *Epiblaster* strategy described in [35]. The reasoning here is to use the difference between the correlation coefficients of cases and controls across all possible SNP pairs as an indication of a significant interaction. Given a set of genotypes X and a binary phenotype Y, with n_1 cases and n_0 controls, the difference is defined as:

$$\Delta\rho(X^{(A,B)}, Y) = \left(\frac{1}{n_1} \sum_{i:y_i=1} \overline{x}_i^A \overline{x}_i^B - \frac{1}{n_0} \sum_{i:y_i=0} \overline{x}_i^A \overline{x}_i^B \right)^2 \tag{4.2}$$

where $\overline{x}_i^A$ and $\overline{x}_i^B$ are two SNPs A and B centered and rescaled by dividing them by the standard deviation for each phenotype class ($y_i = 1$ for cases and $y_i = 0$ for controls). Thus, the SNP pairs with the largest difference in correlation are most likely to exhibit an epistatic interaction. The output of Episcan function is a ranked list of the SNP pairs, sorted by the Z-score which is tabulated for each difference. The description of Episcan R library is available at `https://cran.r-project.org/web/packages/episcan/episcan.pdf`.

In order to assess the quality and the robustness of the results obtained by ReliefF and Episcan, their performances were tested on two different dataset, D_1 and D_2, consisting of two different samples of 20000 rows of the original data. Each algorithm was applied on the two test set separately, and then the outcomes were compared. If the results were similar, the model would have been considered robust and reliable, otherwise not. In our case, among the two top-1000 lists built by ReliefF on D_1 and D_2, 325 SNPs were the same, while for Episcan the common pairs were slightly less then 300. Since the two algorithms were able to identify a considerable amount (more than 25% in both cases) of mutual relations despite the different input data, they have been considered reliable and they have been put into practice.

4.2 Main Effects

An important problem in the context of GWAS analysis is the presence of the so called *main effects*. This is a phenomenon that occurs when specific SNPs influence the phenotype individually. Usually the impact of these markers on the trait dominates and masks out epistatic interactions, which are way more difficult to be identified and whose effect is relevant only when the actions of the SNPs are considered jointly [36]. Hence, alongside the two approaches to feature selection described in the previous section, it was decided to perform another filtering step, taking into account what the authors of Epigtbn did on the real dataset, as explained in the corresponding chapter in [1]. Since the model is not able to explicitly distinguish between interactions above and beyond main effects, it is necessary to perform chi-square tests for testing the conditional independence between each feature and the phenotype variable. Therefore, all SNPs for which the p-value is less than 0.01 are removed. In this way we discard loci that are directly linked to the Class variable, so that the epistatic interactions found between SNP couples and the phenotype are not a spurious effect of the fact that the two SNPs are individually related to the class, but they only influence the class together.

4.3 Bootstrap Methodology

The idea behind our approach to epistasis detection, is to apply Epi-GTBN to retrieve the most relevant connections between SNPs thanks to the network structures produced by the model. In order to achieve this in a reliable and robust way, it was decided to apply the *Bootstrap approach* suggested in the works of N. Friedman and Dana Pe'er [37][38][39]. In these papers, the authors recommend a strategy to evaluate connection confidence in Bayesian networks. In particular this consists in generating *perturbed* datasets, applying the structure learning algorithm on them and then collecting the networks that have

been built. These networks will present changes in their structures due to the fact that they were generated from slightly different data. In this approach, such perturbations are carried out by random sampling the observations with replacement. More formally, given a dataset D,

- For $i = 1, 2, ...m$
  - Sample with replacement N instances from D. Denote the resulting dataset by D_i.
  - Apply the structure learning algorithm on D_i to obtain the network G_i
- For all unique connections f in the graphs, define:

$$\text{conf}(f) = \frac{1}{m}\sum_{i=1}^{m} f(G_i) \tag{4.3}$$

where

$$f(G_i) = \begin{cases} 1 & \text{if } f \in G_i \\ 0 & otherwise \end{cases} \tag{4.4}$$

Finally, a threshold for conf(f) is established, for which f has to be considered relevant. This value is typically set to 0.5. In the papers mentioned above it is showed that high confidence edges obtained with the bootstrap procedure are rarely false positive, even in cases where the datasets D_i are small compared to the original D.

The combination of these operations -Feature selection, the removal of main effects, and bootstrap procedure with Epigtbn- constitutes the methodology we have developed to mine epistatic relationships in GWAS data. As a final result, therefore, we obtain a set of relevant connections which can highlight important biological pathways within the complex network of relationships between genes.

Chapter 5

Application to IBD data

In order to test our approach, we decided to apply it to Inflammatory Bowel Disease data. IBD is a group of inflammatory conditions that affects the colon and the small intestine, and which includes Crohn's disease and ulcerative colitis. IBD are chronic disabling gastrointestinal disorders impacting every aspect of the patient's life and account for substantial costs to the health care system and society. It is estimated that 2.5–3 million people in Europe and 1.6 million in America are affected by this kind of diseases. Scientific evidence clearly pointed out the role of heredity in IBD. Studies have shown that up to 20% of patients have a first-degree relative with one of the diseases, and children of parents with IBD are at greater risk than the general population for developing it. While genetics is clearly a factor, the association is not straightforward. It is likely that interaction between multiple genes is at work, and it is clear that other factors, including environmental factors, must also come into play. Thus, epistatic analysis is of central importance in this context[40][41][42].

5.1 Data

Epi-GTBN analysis was performed on the datasets provided by the BIO3 laboratory of the University of Liège's Interdisciplinary Research Institute in the biomedical sciences. The BIO3 lab obtained the data from the International IBD Genetics Consortium (IIBDGC), which produced a trans-ethnic association study of IBD from an extended cohort of about 90000 European individuals, genotyped on the Immunochip SNP array[43]. The researchers from the University of Liège applied an initial filtration to the data, by discarding SNP models involving rare variants -i.e. those for which the minor allele frequency was less than 5%- or in Hardy–Weinberg equilibrium. Furthermore, SNPs pairs located in the HLA (human leukocyte antigen) gene were removed, as it is difficult to differentiate between main and non-additive effects in that region. Lastly, also SNPs presenting linkage equilibrium ($r^2 > 0.75$), were eliminated from the initial set of features. In this way it was obtained the **Standard** dataset. By applying further manipulations to the Standard data, BIO3 lab obtained 2 other datasets, named for the sake of simplicity **Functional** and **Imputed**. The functional data were produced thanks to the functional filters *FUMA eqtl* and *Biofilter*, as described in [44] and [45]. On the other hand, the imputed data were constructed by replacing the missing values present in the standard data

through the application of the *k-Nearest Neighbour* algorithm. Each dataset comprised $\sim 10^4$ columns, representing SNPs plus the phenotype, and $\sim 10^4$ observations. Since I am not the owner of the data, it is not possible to disclose more detailed information.

Given that the IIBDGC dataset aggregates different cohorts, and contains potentially confounding population structure, the BIO3 lab researchers used the first 7 principal components to model population stratification, and the phenotype variable was adjusted by regressing out those principal components. In this way, two versions (**Corrected** and **NotCorrected** for confounders) for each of the three datasets were generated. In the end, 6 different datasets were produced:

- Standard Corrected
- Standard NotCorrected
- Functional Corrected
- Functional NotCorrected
- Imputed Corrected
- Imputed NotCorrected

A more detailed description of the process through which these data were obtained can be found here [46]. More generally speaking, the SNPs of each dataset can assume the categorical values of 0, 1 and 2, corresponding respectively to *Homozygous for first allele*, *Heterozygous* and *Homozygous for second allele*, while for the class column, 1 represents a case and 0 represents a control. In the non imputed datasets the explanatory variables can also take the value -1, which stands for *Missing genotype*.

5.2 Experimental procedure

As explained in the previous chapter, from the two different feature selection strategies, two sorted lists of SNPs were obtained for each dataset that was provided; in one case the order was based on the weights assigned by ReliefF, while in the other one it was based on the Z-score calculated on the Episcan differences. As for ReliefF, the SNPs with main effects were eliminated from the list, and then the top-400 variables were extracted. In the case of Episcan, we constructed an ordered list according to the first appearance of each SNP in the pairs and then we proceeded as before by eliminating features presenting main effects and consequently extracting the top-400. In this way different datasets of 400 columns and $\sim 10^4$ rows were constructed. The number of top SNPs to be considered was chosen after trying several preliminary experiments on Epi-GTBN, seeking a trade-off between the number of features (and therefore the amount of information utilized) and the algorithm's calculation speed. Since the implementation of the feature selection algorithms is an operation that is up to the laboratory of the Liège University, at the time of writing this thesis, the calculation of the ReliefF ranking over the Standard

NotCorrected data is still in progress, and will be provided as soon as possible.

Finally, it has to be observed that, for each dataset, the number of common columns selected by these two filtering strategies is different. In particular, in the Standard Corrected case the common SNPs (chosen by both Episcan and ReliefF) are 16 out of 400; in the Functional Corrected are 42; in the Functional NotCorrected are 122; in the Imputed Corrected are 11; in the Imputed NotCorrected are 41. In this respect, it should be noted that feature selection is not a well-defined task and there is not always one technique that is *a priori* better than another; so, the final results are heavily influenced by the algorithm that one chooses to employ, since the reduced datasets may vary widely.

As regards the bootstrap procedure, we decided to set the number of iterations m to 10, and the data were sampled considering $N = 10000$, maintaining the same ratio between cases and controls for the phenotype variable. Furthermore we chose to take into account *undirected* connections instead of directed since Epi-GTBN is a heuristic-based structure learning algorithm and edge reversal may happen. Such a conclusion was also made because, when detecting epistatic interactions, the direction of arrows is of secondary importance compared to the connection itself. As a final remark, it was decided to consider *relevant* edges those that have appeared in more than half of the final networks G_i, that is those for which $\text{conf}(f) \geq 0.5$.

The experiment was carried out on the linux server made available by the University of Liège. The calculations were made with the node *urtgen001* of the partition *utgen_unlimit*, with the configuration of Intel Core with 12 total CPUs and a 547 Gb RAM.

5.3 Epi-GTBN issues

Prior to applying Epi-GTBN to the real data, several tests were performed on the model, which demonstrated that it was sensitive to the structure of the input data. In particular, it was noted how the algorithm produced networks in which a large amount of edges pointed to (or came from) the node corresponding to the last SNP of the dataset when considering more than 5000 observations. This was noticed also when performing a random shuffling on the position of the columns. Moreover, this trend was more and more evident as the number of observations considered increased, to the point that almost all connections pointed to the last SNP. Since this behavior would not have allowed us to use a large number of instances, but would have restricted us to only a small part of the total information available, I looked for the source of the problem in the code of Epi-GTBN. This issue arose from the function *mi.2*, utilized for the boolean computation of the mutual information to construct the first individual in the first generation. As I already mentioned above, this step has a great influence over the outcome of all the other subsequent operations and it is fundamental in order to obtain the optimal final result. In order to overcome this problem, I changed *mi.2* with the non-boolean formula 3.2, obtaining a version of Epi-GTBN which is slower but non sensitive to the number of instances taken into account.

As for the missing values contained in the non imputed datasets, given that the model cannot recognize them as such, it was decided to remove the rows containing them from the resized datasets before applying the bootstrap strategy.

Chapter 6

Results

In this chapter we present the results obtained by our approach over the different datasets available, subdivided according to the filtering algorithm employed. Every Epi-GTBN run in the bootstrap procedure took slightly more than 1 hour with the parameters set as previously described, for a cumulative duration of 12 hours. The results are presented in terms of confidence conf(f) of the interactions between SNPs. Here we show only connections that resulted *relevant* according to the criterion set out above.

RELIEFF

Standard Corrected:

ID	SNP1	SNP2	conf(f)
1	rs34463536	rs13086717	1
2	rs719654	rs241410	0.9
3	rs17152571	rs76785757	0.5

Imputed Corrected:

ID	SNP1	SNP2	conf(f)
1	rs17810546	rs17809756	1
2	rs62257823	rs62260377	0.6
3	rs2229092	rs9391858	0.5

Imputed NotCorrected:

ID	SNP1	SNP2	conf(f)
1	rs17810546	rs17809756	1

Functional Corrected:

ID	SNP1	SNP2	conf(f)
1	rs17427599	rs17500468	1
2	rs719654	rs241410	1
3	rs9268365	rs17496307	1
4	rs62257823	rs62260377	1
5	rs4148872	rs2284190	0.6
6	rs17500468	rs4148872	0.5

Functional NotCorrected:

ID	SNP1	SNP2	conf(f)
1	rs2233966	rs2233966	1
2	rs3094214	rs3094214	1
3	rs2517403	rs2233966	0.9
4	rs3095302	rs3095302	0.9
5	rs9268365	rs17496307	0.9
6	rs3130559	rs1265115	0.8
7	rs3132129	rs9261387	0.8
8	rs2857210	rs2621377	0.7
9	rs719654	rs241410	0.7
10	rs1264551	rs1264551	0.6
11	rs17427599	rs17500468	0.6
12	rs2844575	rs2844508	0.6
13	rs2844662	rs2532929	0.6
14	rs2844662	rs2844651	0.6
15	rs130073	rs3130931	0.5
16	rs130073	rs879882	0.5
17	rs2516464	rs2734574	0.5
18	rs2517403	rs3130559	0.5

EPISCAN

Standard Corrected:

ID	SNP1	SNP2	conf(f)
1	rs202162667	rs55840985	1
2	rs9268365	rs17496307	1
3	rs1475961	rs17496307	0.9
4	rs719654	rs241410	0.9
5	rs12524487	rs397081	0.8
6	rs3094214	rs3130559	0.8
7	rs17427599	rs17500468	0.7
8	rs200734705	rs2284190	0.7
9	rs397081	rs10947262	0.7
10	rs17500468	rs200734705	0.6
11	rs17500468	rs719654	0.6
12	rs2621377	rs1044043	0.6
13	rs2857210	rs2621377	0.6
14	rs9268365	rs9391858	0.6
15	rs17496307	rs2239701	0.5
16	rs17496307	rs241424	0.5
17	rs200734705	rs3763366	0.5
18	rs2248372	rs2071463	0.5
19	rs6936863	rs11756897	0.5
20	rs6936863	rs2284190	0.5
21	rs10172063	rs10172063	0.5

Standard NotCorrected:

ID	SNP1	SNP2	conf(f)
1	rs12524487	rs2596532	1
2	rs202162667	rs11738255	1
3	rs202162667	rs11958190	1
4	rs202162667	rs55840985	1
5	rs34670647	rs35480350	1
6	rs9268365	rs17496307	1
7	rs9924308	rs34670647	1
8	rs200665452	rs2263316	0.9
9	rs13409	rs805286	0.8
10	rs2233966	rs3130559	0.8
11	rs2844575	rs2844508	0.8
12	rs3094214	rs3130559	0.8
13	rs3181216	rs202162667	0.8
14	rs9268365	rs9391858	0.8
15	rs17496307	rs241424	0.7
16	rs2857210	rs2621377	0.7
17	rs3095302	rs3130559	0.7
18	rs397081	rs10947262	0.7
19	rs1579219	rs2894145	0.7
20	rs1475961	rs17496307	0.6
21	rs17427599	rs17500468	0.6
22	rs200734705	rs241424	0.6
23	rs200734705	rs241425	0.6
24	rs2523656	rs2844502	0.6
25	rs2844662	rs2532929	0.6
26	rs1634731	rs12524487	0.5
27	rs17496307	rs241425	0.5
28	rs17500468	rs200734705	0.5
29	rs17500468	rs719654	0.5
30	rs200665452	rs2516464	0.5
31	rs200734705	rs2284190	0.5
32	rs200734705	rs3763366	0.5
33	rs3117016	rs3130160	0.5
34	rs3130186	rs2076312	0.5
35	rs34725611	rs11085732	0.5
36	rs4141060	rs762705	0.5
37	rs719654	rs241410	0.5
38	rs9391858	rs241424	0.5

Functional Corrected:

ID	SNP1	SNP2	conf(f)
1	rs202162667	rs55840985	1
2	rs9268365	rs17496307	1
3	rs1475961	rs17496307	0.9
4	rs3094214	rs3130559	0.9
5	rs12524487	rs397081	0.8
6	rs17427599	rs17500468	0.8
7	rs719654	rs241410	0.8
8	rs17500468	rs719654	0.7
9	rs2621377	rs1044043	0.7
10	rs397081	rs10947262	0.7
11	rs6936863	rs3763366	0.7
12	rs13409	rs805286	0.6
13	rs17500468	rs200734705	0.6
14	rs200734705	rs3763366	0.6
15	rs2857210	rs2621377	0.6
16	rs9268365	rs9391858	0.6
17	rs200734705	rs2284190	0.5
18	rs6936863	rs11756897	0.5
19	rs6936863	rs2284190	0.5
20	rs738331	rs2069235	0.5
21	rs9391858	rs241424	0.5

Functional NotCorrected:

ID	SNP1	SNP2	conf(f)
1	rs12524487	rs2596532	0.9
2	rs9268365	rs17496307	0.9
3	rs3094214	rs3130559	0.8
4	rs397081	rs10947262	0.8
5	rs719654	rs241410	0.8
6	rs13409	rs2075788	0.7
7	rs17500468	rs719654	0.7
8	rs2233966	rs3130559	0.7
9	rs2404573	rs1491940	0.7
10	rs2894145	rs5030609	0.7
11	rs12524487	rs397081	0.6
12	rs17496307	rs241424	0.6
13	rs17496307	rs241425	0.6
14	rs2517403	rs2233966	0.6
15	rs2523656	rs2844502	0.6
16	rs2844662	rs2532929	0.6
17	rs2857210	rs2621377	0.6
18	rs3095302	rs3130559	0.6
19	rs9268365	rs9391858	0.6
20	rs12524487	rs2516470	0.5
21	rs12524487	rs2857605	0.5
22	rs13409	rs805286	0.5
23	rs1475961	rs17496307	0.5
24	rs1475961	rs9391858	0.5
25	rs17427599	rs17500468	0.5
26	rs17496307	rs2239701	0.5
27	rs2239707	rs12661281	0.5
28	rs1579219	rs2894145	0.5

Imputed Corrected:

ID	SNP1	SNP2	conf(f)
1	rs202162667	rs55840985	1
2	rs17496307	rs9268365	0.8
3	rs17427599	rs17500468	0.7
4	rs397081	rs12524487	0.7
5	rs1044043	rs2621377	0.6
6	rs13409	rs805286	0.6
7	rs17496307	rs2239701	0.6
8	rs200734705	rs17500468	0.6
9	rs200734705	rs2239701	0.6
10	rs2284190	rs200734705	0.6
11	rs241410	rs719654	0.6
12	rs3177928	rs2239701	0.6
13	rs3177928	rs9268365	0.6
14	rs9262615	rs6940467	0.6
15	rs9391858	rs9268365	0.6
16	rs62257803	rs11914934	0.6
17	rs11756897	rs6936863	0.5
18	rs200734705	rs3763366	0.5
19	rs241437	rs17500468	0.5
20	rs2857210	rs2621377	0.5
21	rs3130559	rs6940467	0.5
22	rs4148872	rs17500468	0.5
23	rs719654	rs17500468	0.5
24	rs9391858	rs1475961	0.5
25	rs9391858	rs2239701	0.5

Imputed NotCorrected:

ID	SNP1	SNP2	conf(f)
1	rs12524487	rs2596532	1
2	rs202162667	rs11958190	1
3	rs202162667	rs55840985	1
4	rs34670647	rs35480350	1
5	rs9924308	rs34670647	1
6	rs202162667	rs11738255	0.9
7	rs3130559	rs2233966	0.9
8	rs17500468	rs17427599	0.8
9	rs9268365	rs17496307	0.8
10	rs200665452	rs2263316	0.7
11	rs200665452	rs2516464	0.7
12	rs2621377	rs1044043	0.7
13	rs2621377	rs2857210	0.7
14	rs3130559	rs3094214	0.7
15	rs9268365	rs3177928	0.7
16	rs17496307	rs2239701	0.6
17	rs200734705	rs2284190	0.6
18	rs2523656	rs2844502	0.6
19	rs2532929	rs2844662	0.6
20	rs397081	rs12524487	0.6
21	rs397081	rs2857605	0.6
22	rs6936863	rs3763366	0.6
23	rs719654	rs241410	0.6
24	rs879882	rs6909636	0.6
25	rs13409	rs805286	0.5
26	rs17500468	rs200734705	0.5
27	rs17500468	rs4148872	0.5
28	rs200734705	rs2071540	0.5
29	rs200734705	rs2239701	0.5
30	rs200734705	rs3763366	0.5
31	rs202162667	rs3181216	0.5
32	rs2233966	rs2233966	0.5
33	rs2844508	rs2844575	0.5
34	rs2894145	rs1579219	0.5
35	rs3130559	rs3095302	0.5
36	rs4141060	rs762705	0.5
37	rs879882	rs2844575	0.5

There are a few observations that has to be done about the results presented. First of all, Epi-GTBN managed to retrieve a larger number of relevant relations with the data reduced by Episcan than with those reduced by ReliefF. This difference is particularly significant for the results achieved on the Standard and the Imputed data, where the model detected 21, 38, 25 and 37 relevant connections using the first filtering strategy, and only 3, 3 and 1 using the latter one. However, in almost all the cases the algorithm produced outputs containing at least one edge with maximum confidence value. Looking at the Episcan Standard NotCorrected and the Episcan Imputed NotCorrected data, it even managed to obtain respectively 7 and 5 connections that appeared in all the 10 networks from the bootstrap procedure.

Indeed the results heavily changed depending on the feature selection algorithm used, since every time they identified different subsets of features whose intersections had a rather low number of elements. It is interesting to note that, in the Functional NotCorrected case, where the filtering strategies detected the largest number of common SNPs (122), is also the case where Epi-GTBN obtained the greater number of relevant relations with data reduced by ReliefF. Furthermore, it is also remarkable that especially in the outcomes from Episcan, there is a considerable amount of SNP pairs that appeared multiple times as relevant connections, giving rise to a recurrent sub-network within the final outcomes.

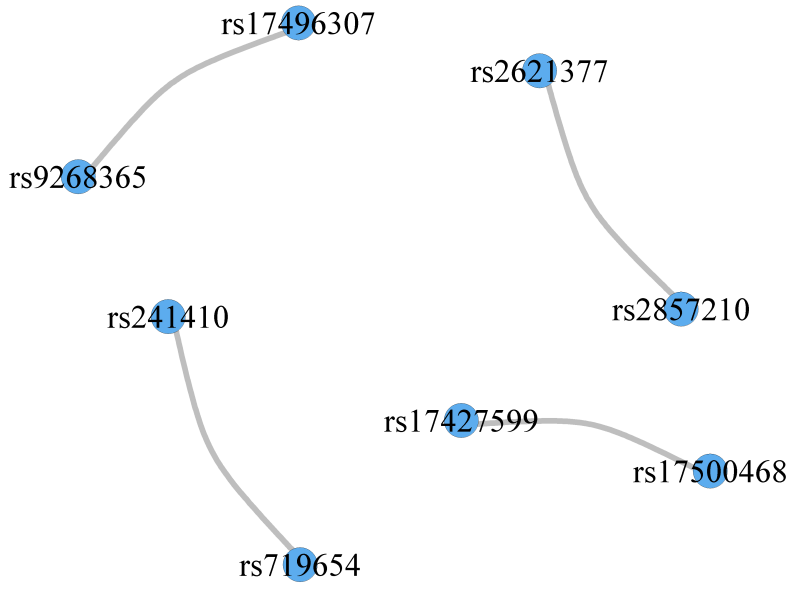

Figure 6.1: Relevant connections shared by all the results obtained from the data reduced by Episcan.

The representation of the structure in figure 6.1, suggests the idea that the connections between these loci play an effectively important role in determining the patient's phenotype, considering also that they almost always have a very high confidence value.

As a matter of fact, the connections rs17427599 ∼ rs17500468, rs719654 ∼ rs241410 and rs9268365 ∼ rs17496307 appear also among the results obtained with Functional data reduced by ReliefF, further reinforcing this hypothesis. Finally, always considering the ReliefF case, it has to be noted that the relation rs719654 ∼ rs241410 can also be found in the outcome from the Standard Corrected dataset, while rs17810546 ∼ rs17809756 is relevant for both the Imputed data.

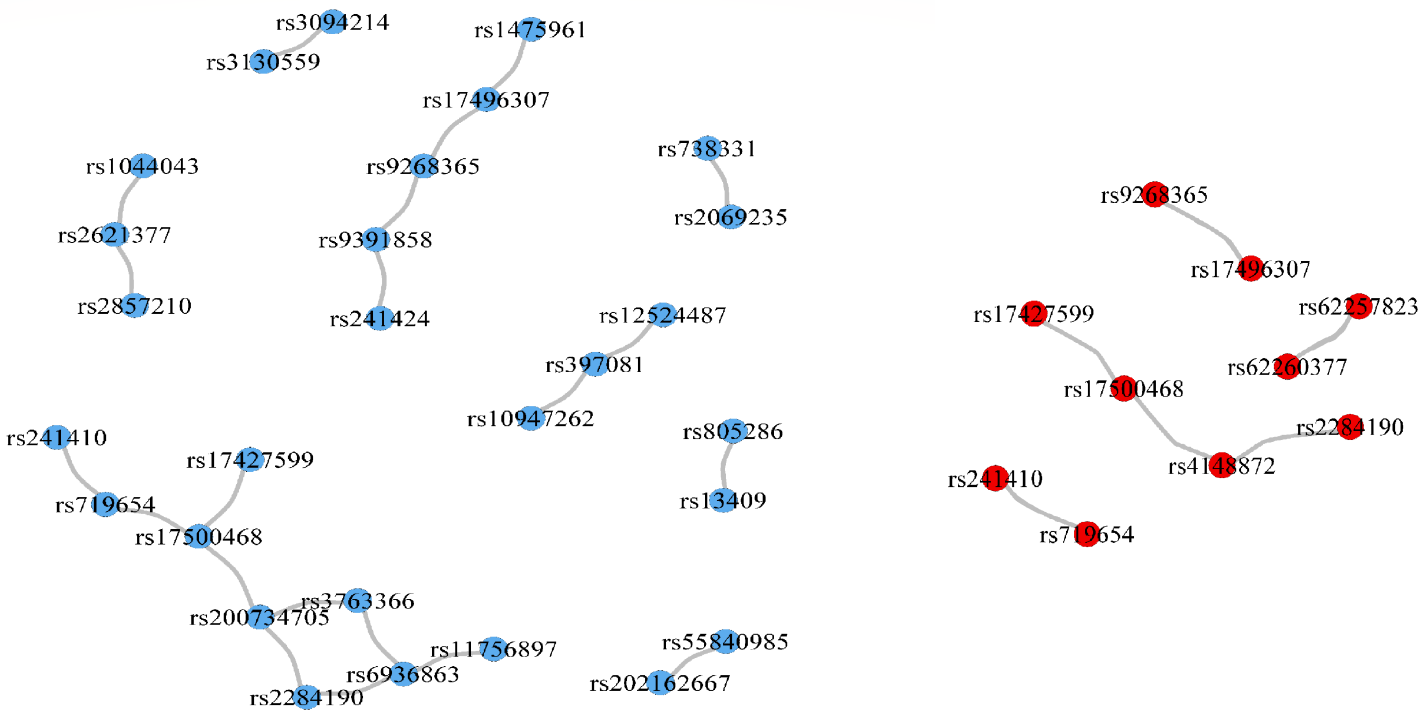

Figure 6.2: Comparison between the subnetworks of relevant connections obtained with the Functional Corrected data. Episcan is in blue, while ReliefF is in red.

The figure above depicts two subnetworks formed by the highly confident connections that have been obtained through the approach we have developed, using Functional Corrected data reduced by the Episcan and ReliefF. As we can see, Episcan managed to retrieve a more complex system of relations between the SNPs considered, detecting an important cluster of 9 markers centered on $rs20073470$ in the bottom-left corner of the figure. Significant differences, however, remain. In particular these can be found not only between networks obtained with different filtering methods, but also between networks constructed within the same bootstrap procedure. In fact, as we already said, the algorithm is sensitive to the perturbed data that are used in the *structure learning* phase, thus giving rise to networks with different skeletons. This is evident in the case of ReliefF,

since most of the time there are only few high-confidence connections (only 1 when considering the Imputed NotCorrected data, for example). As for Episcan, a good amount of SNP pairs occurred in almost all outputs, and the algorithm produced networks with more comparable skeletons and a greater number of confident interactions. Indeed, this strengthens our confidence in the relevance of the highlighted links, since they appeared in most of the results despite the high variability of the structures obtained thanks to the perturbations.

Examples of networks obtained with the bootstrap procedure are presented in the following pages. It should be noted that almost all the relevant edges that have been previously shown in the respective tables appear in the figures. On the other hand, connections between the other nodes are much less significant since they are very variable within the several outputs, having low confidence. Consequently these tend to represent actual relationships between SNPs with much less probability. For this reason, in addition to having clearer images, it was decided to highlight only high-confidence relationships and the corresponding nodes.

6.1 Examples of Bayesian networks

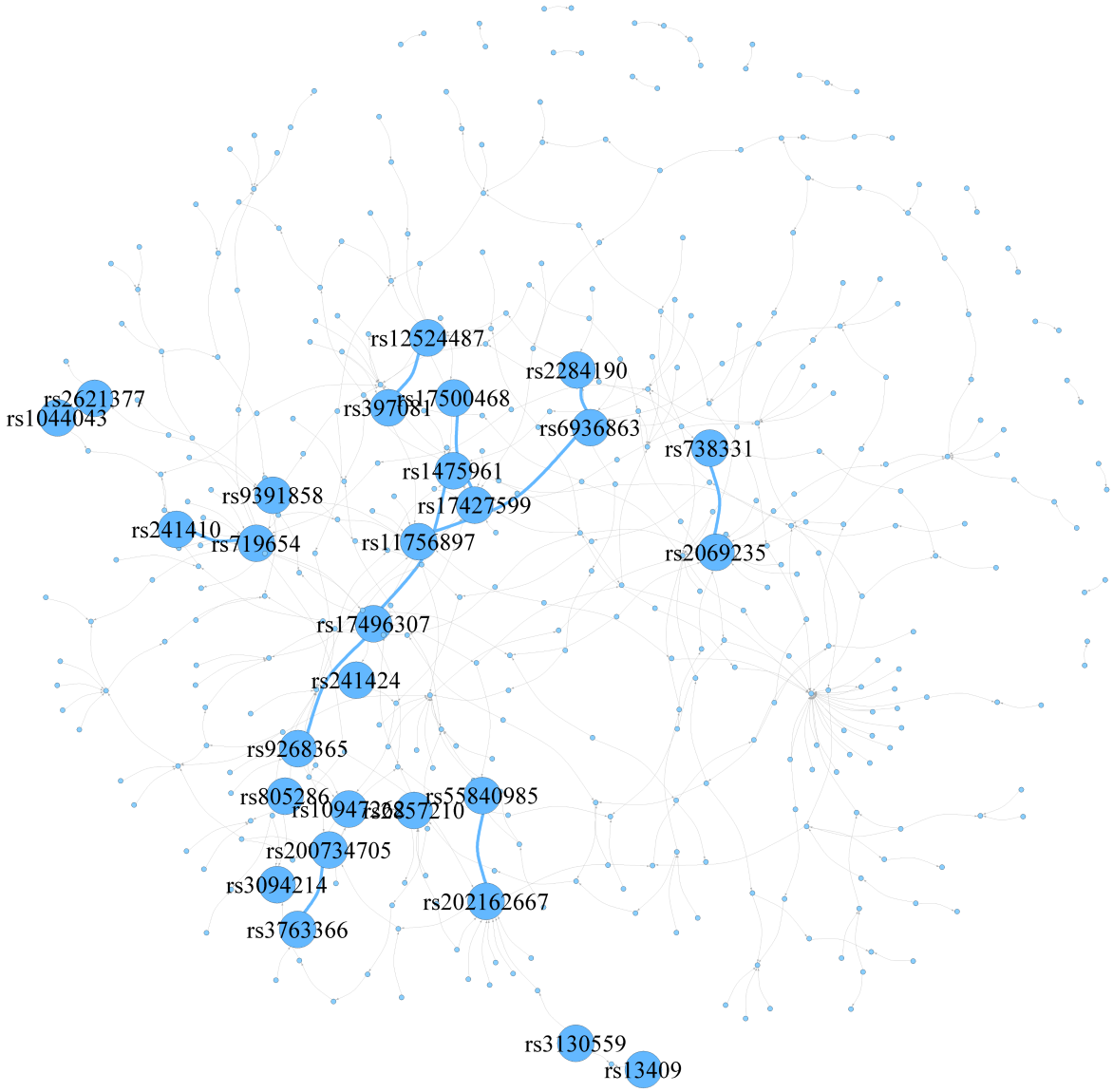

Figure 6.3: Figure depicting the network obtained in iteration 6 with Episcan Functional Corrected data. In blue are represented high confidence connections and their SNPs. We can see the relations rs202162667 - rs55840985 and rs9268365 - rs17496307 (those with conf(f) = 1) in the lower central part of the image.

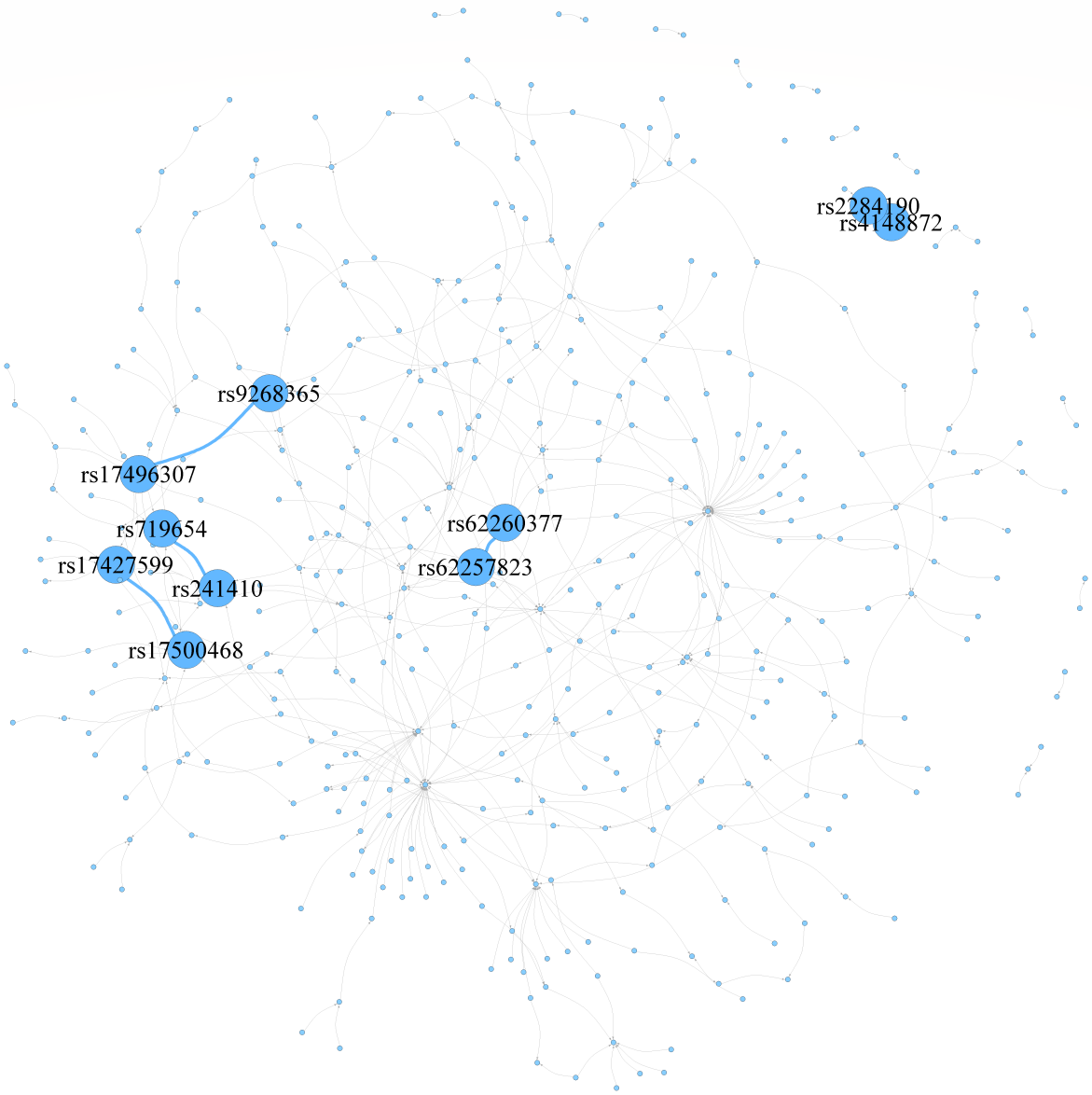

Figure 6.4: Image of the network from iteration 7 in the case of ReliefF Functional Corrected. All connections with maximum confidence, rs17427599 ∼ rs17500468, rs719654 ∼ rs241410, rs9268365 ∼ rs1749630 and rs62257823 ∼ rs62260377, together with rs2284190 ∼ rs4148872, are highlighted.

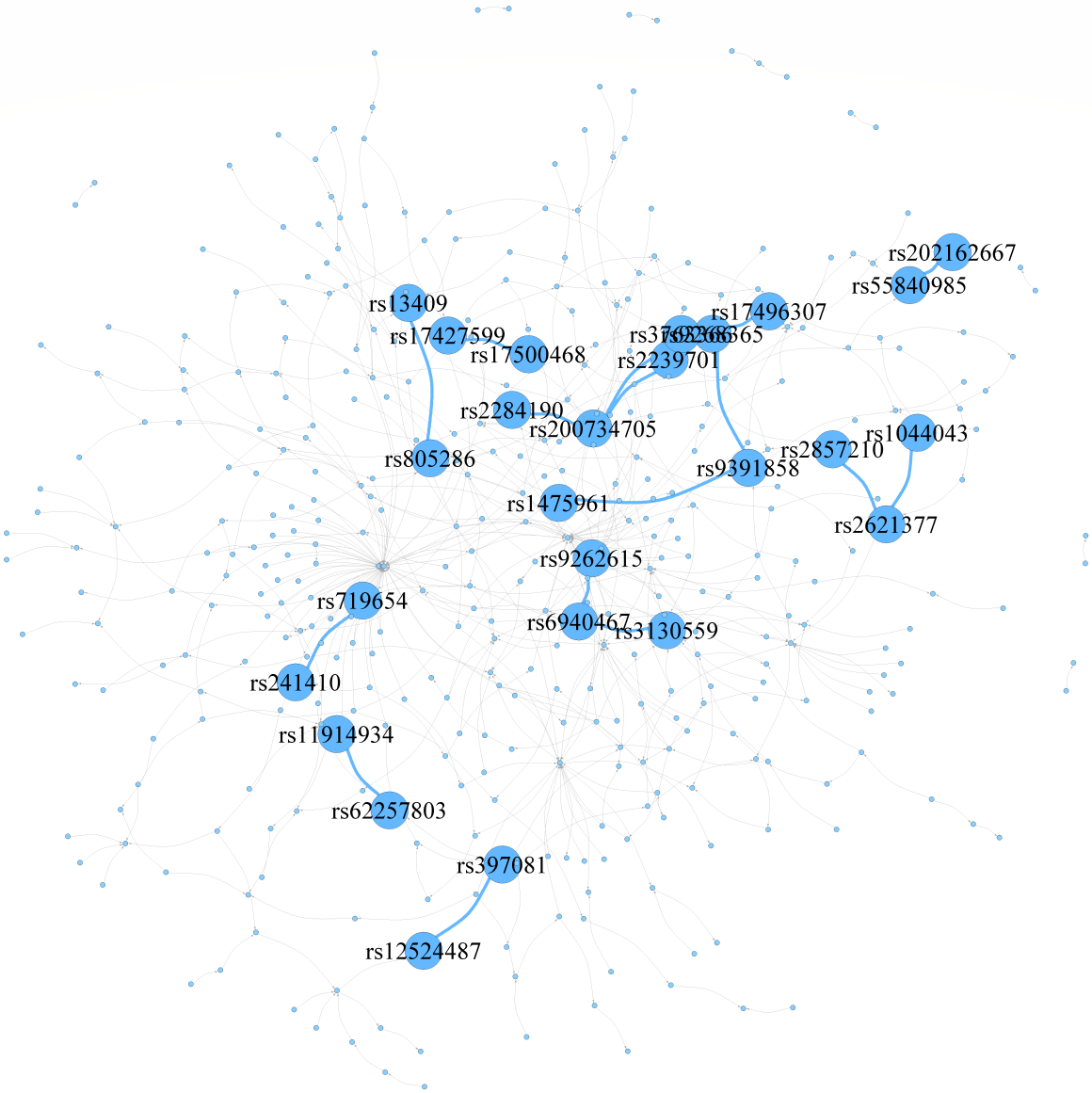

Figure 6.5: Image of the network from iteration 10 using Episcan Imputed Corrected data. Connection rs202162667 ∼ rs55840985 can be found in the top-right corner.

6.2 Comparison with external knowledge

The laboratory of the University of Liège provided us also external information through which it has been possible to compare the results obtained by Epi-GTBN. In particular, BIO3 researchers examined the same IBD data that we used with the **MB-MDR** algorithm, looking for epistatic interactions. MB-MDR, acronym which stands for Model-Based Multifactor Dimensionality Reduction, was first proposed in [47] in order to overcome some major drawbacks that the classical MDR method (already described in the first chapter) suffered from. For example, it struggled to detect important interactions due to pooling too many cells together, and it could not adjust for main effects and for confounding factors. These issues were solved with the model-based version of MDR, which, since its proposal in 2008, has been widely used for epistasis detection in GWAS. The analysis of interactions is performed in three steps: in the first one, each genotype is tested for association with the phenotype variable and is classified as *high risk*, *low risk* or *not significant*, and all genotypes of the same class are merged. In step 2, for each risk categories, high and low, a new association test is performed. The result provides a Wald statistic for the high and for the low categories. Finally, in the third step, the significance is explored through a permutation test on the maximum Wald statistics, thus producing a ranking of relevant SNP pairs.

BIO3 lab researchers obtained two different outputs with MB-MDR, one from the Functional dataset, and one from the Standard dataset. The results of the comparison with the outcomes achieved by Epi-GTBN are summarized in the table below:

	Episcan	ReliefF
Standard Cor.	12	1
Standard NotCor.	19	X
Functional Cor.	8	4
Functional NotCor.	13	5

Table 6.1: SNP interactions that were identified as *relevant* by both MB-MDR and Epi-GTBN. As already said, the Standard NotCorrected data reduced with ReliefF were not provided.

We can clearly see that when considering the data filtered by Episcan, our model managed to detect significant more connections that were considered relevant also by MB-MDR. For example, in the results obtained with Episcan Standard NotCorrected data, 19 SNP pairs were also present in the results of MB-MDR, while in the case of Episcan Functional NotCorrected, there were 13 common couples. However, this is biased by the fact that Epi-GTBN managed to retrieve a greater number of high-confidence interactions with Episcan than with ReliefF. It is also important to highlight that 3 of the connections forming the recurrent subnetwork represented in figure 6.1, that is

rs17427599 ∼ rs17500468, rs719654 ∼ rs241410 and rs9268365 ∼ rs17496307, appear also among the relevant interactions detected by MB-MDR.

In conclusion, we can state that a strong bias remains due to the specific feature selection technique that was chosen. Epi-GTBN performed better with the data reduced by Episcan than with those reduced by ReliefF; this is even more substantiated by looking at the relevant locus-locus interactions retrieved by an independent method such as MB-MDR. In fact, almost all the relevant connections detected by our model were also considered relevant by MB-MDR, and this is particularly evident in the case of Episcan Functional NotCorrected, with 19 out of a total of 28 connections in common with the external knowledge. However, it is important to remember that the data we used here were in first place employed in order to test the whole Bootstrap-enhanced Epi-GTBN methodology that we developed and, as a matter of fact, our approach was found to be effective, considering also the observations made in this section.

Here below we present two examples of subnetwoks comprising the SNP pairs which both our model and MBMDR considered relevant.

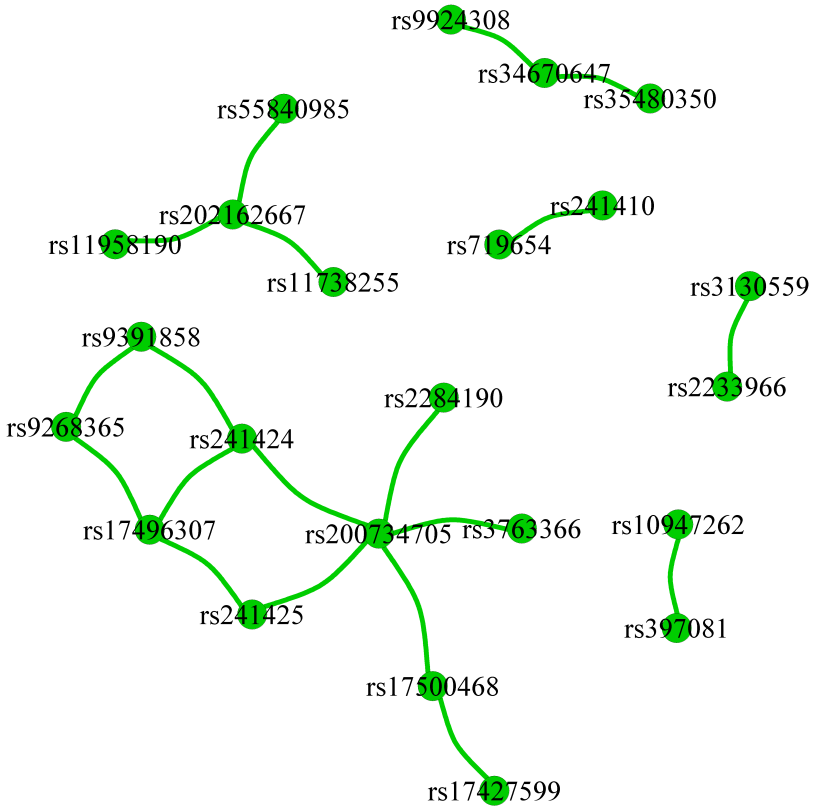

Figure 6.6: Relevant connections detected by both MBMDR and Epi-GTBN in the case of Episcan Standard NotCorrected data.

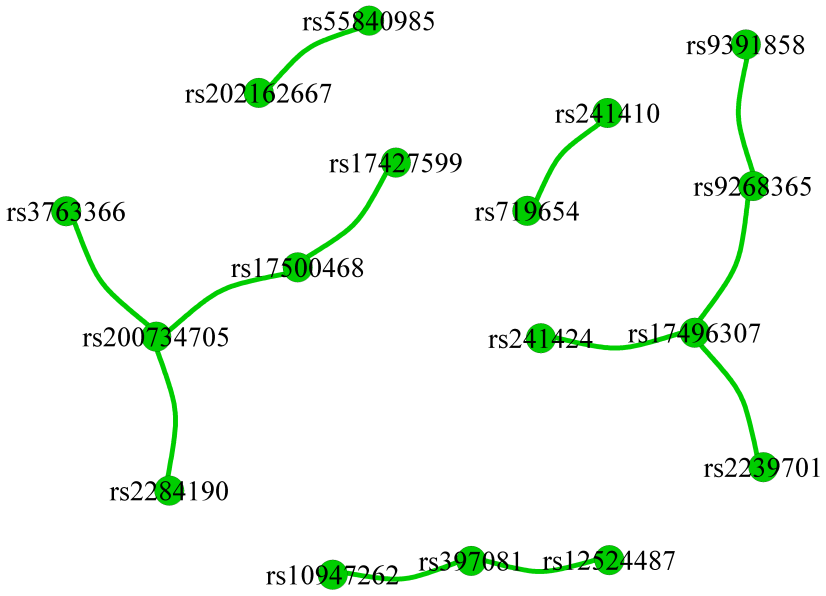

Figure 6.7: Relevant connections detected by both MBMDR and Epi-GTBN in the case of Episcan Standard Corrected data.

Chapter 7

Discussion

Over the course of this work I have described the structure and functioning of the approach to epistasis detection that I developed by joining together several methodologies that I found in the literature. A very important part of my *modus operandi* is the Bootstrap procedure which helps to solve various problems regarding the evaluation and the interpretation of the results. In this field of research, this is not a trivial task at all. In fact we are not facing a classification problem with external information and we cannot evaluate our model by computing *Accuracy*, *F-measure* or *Confusion Matrices*. As a matter of fact, in our case there is not a *ground knowledge* to compare our results with since we are literally searching for it. One strategy to overcome this problem is to look for literature that addresses the same problem and then compare the results obtained, as we did in section 6.2 where we showed how our outcomes are greatly confirmed by other independent applications. However this is generally not sufficient to support the validity of a methodology. In fact, although there exist few other articles about the identification of epistatic interactions in IBD data ([46], [48], [49]), none of them applied the same operations that were employed in our case. Thus, since the final results strongly depend on the particular techniques that have been adopted, and since our approach, as a whole, is new, a comparison with other studies is in general not straightforward and has to be taken with caution. Hence, for these reasons we further rely on the Bootstrap technique, in order to have a form of *internal* evaluation of the results, so that we are certain to obtain statistically significant connections.

As can be deduced from the different results obtained using ReliefF and Episcan, another problem in epistasis detection is that the final results largely depend on the initial filtering procedure, through which a limited group of SNPs is selected, thus restricting the search space. Indeed, this is a common issue for any approach that seeks to extract sensitive information from GWAS. In fact, algorithms capable of exhaustively analyze modern genomic datasets with hundreds of thousands of dimensions that can be run on normal machines in a reasonable time do not yet exist. The computational load would be too high. Therefore, an initial phase in which the analysis is restricted to a subset of genetic markers is necessary in all approaches to epistasis identification. For example, in [46] the authors applied different functional filters to the data using three different ways of mapping SNPs to genes -*Positional*, *eQTL* and *Chromatin*- in order to retrieve relevant interactions, and these different configurations produced different results. Furthermore, beyond the initial filtering step, even small changes in subsequent operations (Linkage Disequilibrium correction, use of biological insights about the disease, analytical method-

ology, et cetera) can produce widely variable outcomes [50].

Consequently, also in this case it is clear the importance of the Bootstrap procedure, since it guarantees the solidity and the robustness of the results retrieved. Moreover, another advantage is that it allows validation without the need for comparison with any external biological knowledge, so that even a non-domain expert is able to derive relevant information from what has been obtained.

Another fundamental point that characterizes my approach is the fact that it mines epistatic interactions in a fast and relatively simple way by exploiting the great possibilities offered by Bayesian networks. These statistical tools, in fact, can identify existing relationships between genetic markers and the disease status intuitively by constructing models of causal influence, through which it is possible to directly extract implicit information. Their drawbacks -mainly their tendency to fall into a local minimum which affects their accuracy- are overcame by Epi-GTBN, where the structure learning task is carried out thanks to a genetic algorithm combined with tabu search strategy. These enhancements ensure the diversity of population, help to achieve the global optimal solution and accelerate the convergence of the model. Still with regard to Epi-GTBN, however, several problems have been identified, for which the algorithm appeared to be dependent on the structure of the input dataset. These issues have been solved by replacing the function *mi.2* for the boolean computation of the mutual information with the more correct but slower 3.2.

Further improvements may include the use of more powerful machines (for instance equipped with GPUs) that would allow to use larger subsets of the data, or more generous parameters in the bootstrap procedure -for example setting $N = 50$ or 100 in order to obtain even more solid results- or in Epi-GTBN -considering a higher number of elements in the generations of the genetic algorithm or in the tabu list or a greater maximum number of iterations-. Another possibility that more powerful machines allow, would be to run other feature selection algorithms besides filters, such as wrappers, which are more computationally expensive.

Additional Material

All the codes that have been written in order to carry out this work are publicly available at `https://github.com/EdoardoGerva/Thesis`.

Acknowledgements

I would like to thank my thesis supervisor Prof. Stella, my co-supervisor Prof. Van Steen, and Prof. Textor from Radboud University, who first taught me about Bayesian Networks. I would also like to acknowledge every member of the BIO3 group and whoever helped me with researches and useful information and suggestions, especially Federico, Diane, Andrew together with Elena and my parents.

Bibliography

[1] Yang Guo et al. "Epi-GTBN: an approach of epistasis mining based on genetic Tabu algorithm and Bayesian network". In: *BMC bioinformatics* 20.1 (2019), p. 444.

[2] Yang Huang, Stefan Wuchty, and Teresa M Przytycka. "eQTL epistasis–challenges and computational approaches". In: *Frontiers in genetics* 4 (2013), p. 51.

[3] William Bateson and Gregor Mendel. *Mendel's principles of heredity*. Courier Corporation, 2013.

[4] Ronald A Fisher. "XV.—The correlation between relatives on the supposition of Mendelian inheritance." In: *Earth and Environmental Science Transactions of the Royal Society of Edinburgh* 52.2 (1919), pp. 399–433.

[5] Clément Niel et al. "A survey about methods dedicated to epistasis detection". In: *Frontiers in genetics* 6 (2015), p. 285.

[6] Jianwei Gou et al. "Stability SCAD: a powerful approach to detect interactions in large-scale genomic study". In: *BMC bioinformatics* 15.1 (2014), pp. 1–11.

[7] Marylyn D Ritchie et al. "Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer". In: *The American Journal of Human Genetics* 69.1 (2001), pp. 138–147.

[8] Xiang Wan et al. "BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies". In: *The American Journal of Human Genetics* 87.3 (2010), pp. 325–340.

[9] Ryan J.Urbanowicz. "Relief-based feature selection: Introduction and review". In: *Journal of Biomedical Informatics* (2018), pp. 189–203. DOI: `https://doi.org/10.1016/j.jbi.2018.07.014`.

[10] Andrew Chatr-Aryamontri et al. "The BioGRID interaction database: 2015 update". In: *Nucleic acids research* 43.D1 (2015), pp. D470–D478.

[11] Sarah A Pendergrass et al. "Genomic analyses with biofilter 2.0: knowledge driven filtering, annotation, and model development". In: *BioData mining* 6.1 (2013).

[12] Yu Zhang and Jun S Liu. "Bayesian inference of epistatic interactions in case-control studies". In: *Nature genetics* 39.9 (2007), pp. 1167–1173.

[13] Yupeng Wang et al. "AntEpiSeeker: detecting epistatic interactions for case-control studies using a two-stage ant colony optimization algorithm". In: *BMC research notes* 3.1 (2010), p. 117.

[14] Conrad H Waddington. "Canalization of development and the inheritance of acquired characters". In: *Nature* 150.3811 (1942), pp. 563–565.

[15] Judea Pearl. "Bayesian netwcrks: A model cf self-activated memory for evidential reasoning". In: *Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine, CA, USA*. 1985, pp. 15–17.

[16] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[17] Michael Irwin Jordan. *Learning in graphical models*. Vol. 89. Springer Science & Business Media, 1998.

[18] Dan Geiger, Thomas Verma, and Judea Pearl. "d-separation: From theorems to algorithms". In: *Machine Intelligence and Pattern Recognition*. Vol. 10. Elsevier, 1990, pp. 139–148.

[19] Nevin Lianwen Zhang and David Poole. "Exploiting causal independence in Bayesian network inference". In: *Journal of Artificial Intelligence Research* 5 (1996), pp. 301–328.

[20] Steffen L Lauritzen and David J Spiegelhalter. "Local computations with probabilities on graphical structures and their application to expert systems". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 50.2 (1988), pp. 157–194.

[21] Judea Pearl. "Evidential reasoning using stochastic simulation of causal models". In: *Artificial Intelligence* 32.2 (1987), pp. 245–257.

[22] Thomas Griffiths and Alan Yuille. "A primer on probabilistic inference". In: *The probabilistic mind: Prospects for Bayesian cognitive science* (2008), pp. 33–57.

[23] Diego Colombo and Marloes H Maathuis. "Order-independent constraint-based causal structure learning." In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 3741–3782.

[24] David Maxwell Chickering. "Optimal structure identification with greedy search". In: *Journal of machine learning research* 3.Nov (2002), pp. 507–554.

[25] Juan I Alonso-Barba, Jose A Gámez, Jose M Puerta, et al. "Scaling up the greedy equivalence search algorithm by constraining the search space of equivalence classes". In: *International journal of approximate reasoning* 54.4 (2013), pp. 429–451.

[26] Xia Jiang, M Michael Barmada, and Shyam Visweswaran. "Identifying genetic interactions in genome-wide data using Bayesian networks". In: *Genetic epidemiology* 34.6 (2010), pp. 575–581.

[27] Bing Han et al. "Genetic studies of complex human diseases: characterizing SNP-disease associations using Bayesian networks". In: *BMC systems biology* 6.S3 (2012), S14.

[28] Bing Han, Meeyoung Park, and Xue-wen Chen. "A Markov blanket-based method for detecting causal SNPs in GWAS". In: *BMC bioinformatics*. Vol. 11. S3. Springer. 2010, S5.

[29] Clément Niel et al. "SMMB: a stochastic Markov blanket framework strategy for epistasis detection in GWAS". In: *Bioinformatics* 34.16 (2018), pp. 2773–2780.

[30] Dimitris Margaritis and Sebastian Thrun. *Bayesian network induction via local neighborhoods*. Tech. rep. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1999.

[31] Nir Friedman, Dan Geiger, and Moises Goldszmidt. "Bayesian network classifiers". In: *Machine learning* 29.2-3 (1997), pp. 131–163.

[32] Luisa M. Reyes-Cortes et al. Hector E. Sanchez-Ibarra. "Genotypic and Phenotypic Factors Influencing Drug Response in Mexican Patients With Type 2 Diabetes Mellitus". In: *Front. Pharmacol.* (2018). DOI: https://doi.org/10.3389/fphar.2018.00320.

[33] Lawrence B. Holder. "Machine learning for epigenetics and future medical applications". In: *Journal Epigenetics* (2017), pp. 505–514. DOI: https://doi.org/10.1080/15592294.2017.1329068.

[34] Marwa Mostafa Abd El Hamid, Mai S Mabrouk, and Yasser MK Omar. "DEVELOPING AN EARLY PREDICTIVE SYSTEM FOR IDENTIFYING GENETIC BIOMARKERS ASSOCIATED TO ALZHEIMER'S DISEASE USING MACHINE LEARNING TECHNIQUES". In: *Biomedical Engineering: Applications, Basis and Communications* 31.05 (2019), p. 1950040.

[35] Bertram Müller-Myhsok et al. "EPIBLASTER-fast exhaustive two-locus epistasis detection strategy using graphical processing units". In: *European Journal of Human Genetics volume* (2010). DOI: https://doi.org/10.1038/ejhg.2010.196.

[36] Can Yang et al. "Identifying main effects and epistatic interactions from large-scale SNP data via adaptive group Lasso". In: *BMC bioinformatics* 11.S1 (2010), S18.

[37] Nir Friedman et al. "Using Bayesian networks to analyze expression data". In: *Journal of computational biology* 7.3-4 (2000), pp. 601–620.

[38] Dana Pe'er et al. "Inferring subnetworks from perturbed expression profiles". In: *Bioinformatics* 17.suppl_1 (2001), S215–S224.

[39] Nir Friedman et al. "Data Analysis with Bayesian Networks: A Bootstrap Approach". In: *arXiv e-prints* (2013).

[40] Jürgen Glas et al. "Novel Genetic Risk Markers for Ulcerative Colitis in theIL2/IL21Region Are in Epistasis WithIL23Rand Suggest a Common Genetic Background for Ulcerative Colitis and Celiac Disease". In: *American journal of gastroenterology* 104.7 (2009), pp. 1737–1744.

[41] Casper G Noomen, Daniel W Hommes, and Herma H Fidder. "Update on genetics in inflammatory disease". In: *Best Practice & Research Clinical Gastroenterology* 23.2 (2009), pp. 233–243.

[42] Fentaw Abegaz et al. "Epistasis Detection in Genome-Wide Screening for Complex Human Diseases in Structured Populations". In: *Systems Medicine* 2.1 (2019), pp. 19–27.

[43] Adrian Cortes and Matthew A Brown. "Promise and pitfalls of the Immunochip". In: *Arthritis research & therapy* 13.1 (2011), pp. 1–3.

[44] Kyoko Watanabe et al. "Functional mapping and annotation of genetic associations with FUMA". In: *Nature communications* 8.1 (2017), pp. 1–11.

[45] GTEx Consortium et al. "Genetic effects on gene expression across human tissues". In: *Nature* 550.7675 (2017), p. 204.

[46] Diane Duroux et al. "Interpretable network-guided epistasis detection". In: *bioRxiv* (2020).

[47] M Luz Calle et al. "Improving strategies for detecting genetic patterns of disease susceptibility in association studies". In: *Statistics in medicine* 27.30 (2008), pp. 6532–6546.

[48] Zhenwu Lin et al. "NOD2 mutations affect muramyl dipeptide stimulation of human B lymphocytes and interact with other IBD-associated genes". In: *Digestive diseases and sciences* 58.9 (2013), pp. 2599–2607.

[49] Jie Zhang et al. "Multiple Epistasis Interactions Within MHC Are Associated With Ulcerative Colitis". In: *Frontiers in genetics* 10 (2019), p. 257.

[50] Kyrylo Bessonov, Elena S Gusareva, and Kristel Van Steen. "A cautionary note on the impact of protocol changes for genome-wide association SNP× SNP interaction studies: an example on ankylosing spondylitis". In: *Human Genetics* 134.7 (2015), pp. 761–773.