DETERMINISTIC MODELING AND INFERENCE OF BIOCHEMICAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
MIDDLE EAST TECHNICAL UNIVERSITY
BY

DENİZ SEÇİLMİŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF BIOINFORMATICS

APRIL 2017

DETERMINISTIC MODELING AND INFERENCE OF BIOLOGICAL SYSTEMS

Submitted by DENİZ SEÇİLMİŞ in partial fulfillment of the requirements for the degree of **Master of Science in The Department of Bioinformatics Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin ______________________
Director, **Graduate School of Informatics**

Assoc. Prof. Dr. Yeşim Aydın Son ______________________
Head of Department, **Health Informatics**

Assoc. Prof. Dr. Vilda Purutçuoğlu ______________________
Supervisor, **Statistics**

Examining Committee Members:

Assoc. Prof. Dr. Yeşim Aydın Son ______________________
Health Informatics, Middle East Technical University

Assoc. Prof. Dr. Vilda Purutçuoğlu ______________________
Statistics, Middle East Technical University

Assist. Prof. Dr. Aybar Can Acar ______________________
Health Informatics, Middle East Technical University

Assoc. Prof. Dr. Serpil Aktaş Altunay ______________________
Statistics, Hacettepe University

Assist. Prof. Dr. Özlem Defterli ______________________
Mathematics, Çankaya University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this wok.

Name, Last name :	**Deniz Seçilmiş**
Signature :	________________

ABSTRACT

DETERMINISTIC MODELING AND INFERENCE OF BIOLOGICAL SYSTEMS

Seçilmiş, Deniz
MSc., Department of Bioinformatics
Supervisor: Assoc. Prof. Dr. Vilda Purutçuoğlu

April 2017, 66 pages

The mathematical description of biological networks can be performed mainly by stochastic and deterministic models. The former gives more information about the system, whereas, it needs very detailed measurements. On the other hand, the latter is relatively less informative, but, the collection of their data is easier than the stochastic ones, rendering it a more preferable modeling approach. In this study, we implement the deterministic modeling of biological systems due to the underlying advantage. Among many alternatives, we use the Gaussian graphical model (GGM) and evaluate its performance with respect to the random forest algorithm, which we suggest as an alternative approach of GGM. We estimate the model parameters, i.e., the structure of the networks, and then assess their findings based on their accuracies. Finally, we extend the study by using copulas in the description of data and apply the same modeling approaches to assess their effects.

Keywords: Systems biology, Gaussian graphical model, random forest algorithm, copulas.

ÖZ

BİYOLOJİK AĞLARIN DETERMİNİSTİK MODELLEMESİ VE SONUÇ ÇIKARIMI

Seçilmiş, Deniz
Yüksek Lisans, Biyoenformatik Bölümü
Tez Yöneticisi: Doç. Dr. Vilda Purutçuoğlu

Nisan 2017, 66 sayfa

Biyolojik ağların matematiksel tanımlaması, başlıca, stokastik ve deterministik modellerle yapılabilir. Bunlardan ilki, sistem hakkında daha çok bilgi veriyor olmasına rağmen, çok detaylı ölçümler gerektirmektedir. Öte yandan, ikincisi, nispeten daha az bilgi verir fakat verilerinin toplanması stokastikte olduğundan daha kolaydır. Dolayısıyla, daha çok tercih edilen bir modelleme yaklaşımıdır. Belirtilen avantajından ötürü, biz bu çalışmada, biyolojik sistemlerin deterministik modellemesini uygulamaktayız. Birçok alternatif arasından Gaussian grafiksel modelini (GGM) kullanmaktayız ve performansını, GGM'ye bir alternatif yaklaşım olarak önerdiğimiz rasgele orman algoritmasına göre değerlendirmekteyiz. Model parametrelerini, diğer bir deyişle ağların yapılarını, tahmin etmekteyiz ve sonrasında bulguların doğruluklarına göre değerlendirmekteyiz. Son olarak, çalışmayı, verinin tanımında kopulaları kullanarak genişletmekteyiz ve etkileri değerlendirmek için aynı modelleme yaklaşımlarını uygulamaktayız.

Anahtar Sözcükler: Sistem biyolojisi, Gaussian grafiksel modeli, rasgele orman algoritması, kopulalar.

To my dad :)

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Vilda Purutçuoğlu for her endless support and smiling face. It was a pleasure for me to have the chance to study with her.

I would like to express my gratitude to Yeşim Aydın Son for her everlasting support, being smiling all the time and for all the other things. I would also like to thank all my professors Aybar Can Acar, Nurcan Tunçbağ and Rengül Atalay from Bioinformatics Department at METU for their support. I would also like to thank TÜBİTAK grant (Project no: 114E636) and the BAP project at Middle East Technical University (Project no: BAP-01-09-2016-002) for their support.

I would like to thank Eda Kumcu for being like a sister to me; and to my friends Umut Ağyüz, Tuğba Kaya, Atefeh Lafzi, Güngör Budak, Özlem Özkan, Burcu Yaldız and Ayşegül Tombuloğlu from the Bioinformatics Department. I would also like to thank my friends Ezgi Güzelküçük, Onur Öztornacı, Emre Bilgin and Can Arslan.

There are very important teachers in my life, Doğan Kaya and Eser Ördem. I am not sure if thanks would be enough to tell what they mean to me, it was a pleasure to learn from them.

I would like to thank my grandma Hatice Sevilen for her endless support and my grandpa Mahmut Seçilmiş for the history lessons and backgammon rounds and for being such a wonderful grandpa. I would also like to thank my uncle Haluk Seçilmiş for teaching me the basis of programming and for his endless interest in my studies.

I would like to thank my roommate Tuğba Pişkin and the team 101: Hande Güneş, Eti Levi, Duygu Tolunay, Selen Yamak and Nur Yıldırım. They are like family to me. I could not be happier to knowing them.

I would like to thank my amazing cat Cevdet. She constitutes a very important and indispensable part of my life. Also, I would like to thank my mouse Protein who recently died; though he was new, he made himself loved by his super cute attitudes.

Special thanks go to my brother Tuna Seçilmiş. He has always been a wonderful brother. I wish him all the best things in his life. I would like to thank my mother Emine Seçilmiş for always being there for me with her own way :)

Finally, the biggest gratitude goes to my dad, Ata Seçilmiş. Words are not enough to define what he means to me. He has always been not only the best dad ever, but also the best friend to me. None of these would have even been possible without him. He is my biggest chance in life. Thank you…

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CART	Classification and Regression Trees
GGM	Gaussian Graphical Model
glasso	Graphical Lasso
RFA	Random Forest Algorithm
FDR	False Discovery Rate
FN	False Negative
FP	False Positive
FPR	False Positive Rate
JAK-STAT	Janus Kinase Signal Transducer and Activation of Transcription
MCC	Matthew's Correlation Coefficient
PPI	Protein-Protein Interaction
TN	True Negative
TP	True Positive
TPR	True Positive Rate

CHAPTER 1

INTRODUCTION

We are living in the era of information, where it is easy to access a broad variety of knowledge ranging from biological datasets, such as gene expression, cell signaling etc. to all the personal information of almost all the people across the world, rendering technological and scientific developments highly possible. The measurements of blood or tissue samples from sick to healthy people are recorded and kept in the electronic environment based on their approval, and researchers have several options to use those data in their studies. There are several useful databases, i.e., NCBI, Ensembl, from which we can obtain a wide range of biological data. However, the biological data are not remarkably meaningful without being processed under different statistical and computational techniques. For instance, although it is possible to find a lot of protein sequences in the existing databases, only a small part of those has the structural information. Hence, identifying their structures can be considered as an issue in the molecular biology and genetics. Keeping proteins on the agenda, there are many other issues to deal with during the collection of the data as well.

So, in the basis, DNA carries all the genetic information included in the coding and the non-coding parts that the living organisms need it to keep themselves alive. The central dogma of the molecular biology states that the genetic information carried by the coding part of DNA makes RNA by the process called the transcription. Then as a follow-up process, named as the translation, RNA makes the protein. The proteins have three-dimensional structures providing them with their biological functions. Those functional proteins tend to interact with each other in addition to acting by themselves. Their interactions cause biological pathways and many other incidents, which are all represented by networks consisting of nodes (proteins) and edges connecting between the nodes. One way to save the raw data from being meaningless is to infer those interaction networks as accurate as possible, which has a huge importance for understanding biological incidents and disease pathways. Therefore, it is crucial in the personalized medicine.

Hereby, if we see the biological data at a higher level, we start to deal with the network. Basically, the construction of the networks begins with connecting the two closest proteins with each other, and then keeping searching for the next closest pair of proteins to bind. The next one could be a completely new protein pair, or it could be a protein connecting with the previously bound couple. If the process goes with the second option, i.e., binding a single protein to a protein couple, then the three proteins together are called as a motif. The procedure continues until there is no protein remaining. Finally, we have a biological network consisting of motifs and modules resulting in networks. Hence, the identification of structures enables researchers to better understand the true activation of the biological process.

The biological networks can be modeled mainly by the two methods, namely, deterministic and stochastic approaches. The former gives the information about the presence or lack of interactions between two nodes; whereas, the latter provides an extra information by stating the direction of the interaction. However, as the information needed for the modeling is not very detailed with respect to the stochastic models, the application of deterministic models is very common. Furthermore, these models can successfully explain the steady-state behavior of the systems. Therefore, we use the deterministic modeling of complex networks in our analyses and consider their alternatives under parametric and nonparametric models.

Hereby, one of the major models, which we work on in detail, is the probabilistic Gaussian graphical model (GGM) as the graph-based approach. Combining it with the lasso regression, the graphical lasso (glasso) can be suggested as a suitable technique to infer GGM in biological systems. The lasso regression implies a linear regression model where each node is represented by all the other vertices under the conditional independency. It means that if there is no edge between two nodes, i.e., proteins, they are conditionally independent given all the other proteins in the system. In the construction of the network, GGM uses the covariance matrix of the data defined under the lasso regression. Accordingly, in the final stage of the inference, the estimated precision matrix, which is the inverse of the covariance matrix, is transformed as a binary adjacency matrix in order to construct the network. In this study, we further suggest a nonparametric modeling approach, which can describe the steady-state activation of the systems, and can be a strong alternative of GGM. Here, we apply the random forest algorithm.

In order to make a general categorization, it can be said that the machine learning can be considered as having two main subgroups: supervised and unsupervised learning. The former includes the classification and the regression; while, the clustering is in the subpart of the latter. The classification is a procedure of the allocation of each observation to the subgroups that are previously known. Here, in the calculation of the classification, the boosting and the bagging have important roles. The boosting is a procedure that the misclassified points are repeatedly voted with respect to their weights at each iteration in order to produce a strong ensemble. However, for the data having high variance, the boosting is not enough to provide with accurate trees. Since it is very important to construct the model with a high stability and accuracy, the bagging can be put forward in order to avoid this problem by also reducing the variance. The bagging has the advantage of not depending on the earlier trees in the model. However, it has a weak property that it does not change the bias while reducing the variance. Here, the adaptive bagging unravels this challenge by reducing both the variance and the bias. The bagging is suggested as a useful algorithm due to its following advantages. It uses the random features, resulting in an increase in accuracy; it provides accurate estimates for the generalization error (PE^*), strength and correlation; it infers the out-of-bagging error truly. However, we can also face with the problem of overfitting. The reason is that the model may start to fit not only the signal, but also, the noise. In order to overcome this problem, we can work on large datasets. However, it is not always possible to find such sets. In this study, we suggest the random forest algorithm (RFA) that has the basis of the adaptive bagging, and additionally, as an improvement, it creates an upper boundary to the generalization error in order to avoid the underlying overfitting problem. In the construction of the network, RFA maximizes the strength between the nodes and minimizes the correlations under a subset of nodes by taking majority votes in each iteration of the calculation. Considering all these properties, RFA provides with accurate and stable models without handling external values. Hence, in this work, the suggested algorithm is investigated deeply in the following sections. To illustrate the performance of RFA, we evaluate it under the multivariate normally distributed datasets and non-normal datasets generated by copulas with distinct margins. Furthermore, we use real systems with simulated and real observations in order to assess its accuracy in realistically complex structures. In all of these analyses, we compare our outputs with GGM in terms of various accuracy measures.

1.1. Aim of the Study

One approach commonly used in the description of the biological networks can be argued as the probabilistic Gaussian graphical model (GGM) which is based on the most popular and globally used distribution, Gaussian (Normal) distribution whereas, considering the biological systems, data are not always normally distributed, causing mistaken normality approximations in the construction of networks. Additionally, it requires external ad-hoc values such as the threshold point to force the resulting estimated precision matrices to be in a binary form to find the estimated adjacency matrix. Hereby, the aim of this study is to suggest a more accurate approach to infer the biological systems from both normally distributed and non-normally distributed data. For this purpose, we consider the non-parametric approaches forward, and perform the random forest algorithm (RFA) as the propounded technique presenting a non-parametric alternative. In order to reveal whether the non-parametric models can be strong alternatives to deterministic and probabilistic modeling techniques, such as GGM, in the construction of biological systems under steady-state conditions, we conduct comparative analyses between GGM and RFA based on distinct measures of accuracies.

1.2. Motivation

Developing technology renders the analysis of biological data by statistical and computational techniques highly easy. Since the accurate inference of biological systems would help the researchers to correctly identify specific disease pathways or any other biological incidents in terms of the regulatory elements of the system represented by nodes as well as the directed or undirected interactions represented by edges among these elements, inference of biological systems from the raw data can be regarded as one of the promising outcomes of this developing technology. Considering current methodologies for the construction of the biological networks, it can be clearly said that even though there exists a variety of modeling challenges, the choice of the most appropriate model for the data is still a problem. To explicitly, combining the Gaussian graphical model with the lasso regression, the description of the network is implemented under the Gaussian distribution. However, if the data are far away from the normality assumption, performing the analysis by regarding it as normally distributed would be misguided. In this study, our motivation is to suggest a non-parametric approach in an attempt to overcome the underlying challenge arising from inappropriate assumptions, and to construct the networks as accurate as possible without dealing with external requirements. Furthermore, in the selection of the alternative model, we take the capacity of the model into account under highly correlated and large scale, i.e., high dimensional, data as typically observed in the biological systems.

CHAPTER 2

BACKGROUND

This chapter consists of two main parts that are independent of each other. First part explains the general network definition and structure, as well as defining the different types of networks (Barabási et al., 2011). Second part of this chapter demonstrates the Classification and Regression Trees (CART) methodology (Lewis, 2000; Timofeev, 2004), which is the basis of the random forest algorithm suggested as an alternative to the Gaussian graphical model.

2.1. Networks

2.1.1. Definition of Network

A network can be defined in terms of its components such as its nodes and the edges connecting between its corresponding vertices. Entities of a system can be viewed by representing them under networks. Basically, a couple of nodes constitute motifs, and then by becoming larger units, named as modules, they compose the network structures.

The networks can be categorized into subgroups with respect to their explanations from different perspectives. To explicitly, networks can be separated into two branches in terms of their nodes and the properties of their edge or the distributional properties of these edges.

The edge properties should be assigned with respect to the components of the networks. For example, if the deal is to construct a gene regulatory network, the direction of the relationship must be included in the graph, requiring directed networks. However, some networks, such as protein-protein interaction networks, do not require the definition of the direction among its vertices. Therefore, undirected networks can be used to exhibit these systems.

Representing the biological systems as networks can be regarded as one of the important issues since the biological incidents mostly are not easily visible. The biological systems are very organized and everything is in order. However, in some conditions, specific differences may occur in these organized systems. In such kind of situations, identification of the regular system as well as exhibiting the system under the disease conditions help researchers to determine the cause of the situation, solve the problems and to find alternative pathways to render the broken system organized again. There are many other reasons to represent the biological systems as networks such as drug design. If the interactions among the molecules can be identified, researchers could have the idea that which molecule regulates (activates or suppresses) the other(s), and then based on the pathway, the selection process of the drug target becomes much clear. The examples can be extended but the general approach remains the same.

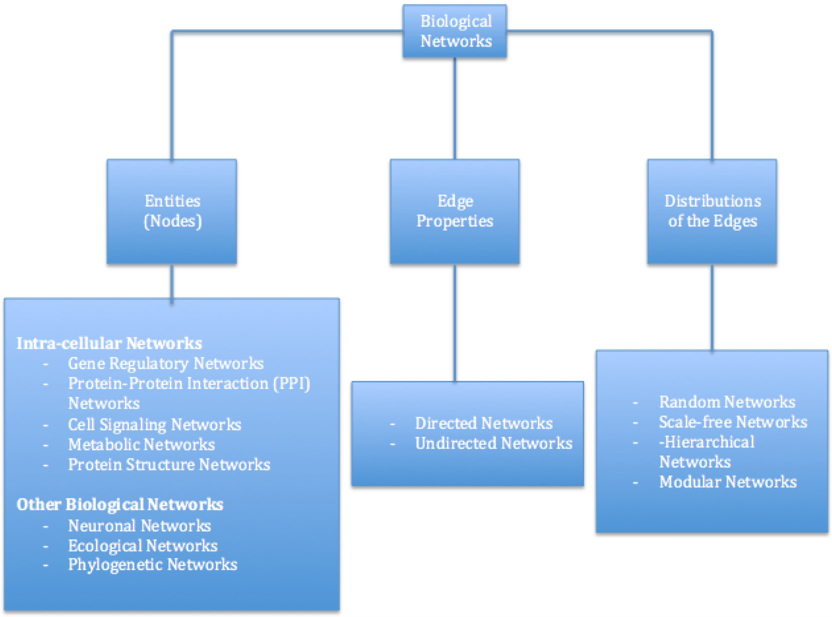

Figure 2.1.: Graphical view of different types of networks.

When the deal is to construct a biological system, the choice of a scale-free network would be the most appropriate network type for not losing the sparsity. Since our main focus is to construct protein-protein interaction networks, undirected graphs with edges distributed under scale-freeness are investigated in this study.

Scale-free Networks

In the molecular biology, there exist a huge number of popular nodes, called hubs. These hubs tend to interact with up to millions of other biological entities in the system. Therefore, the presence of hubs in the network construction plays a key role, and missing the interactions of hubs among other entities may result in an increased number of false negatives. Hereby, the scale-free networks, in the basis, enable viewing the hubs in the systems. Therefore, it is preferable when the deal is to model a biological network.

2.1.2. Modeling of Biological Networks

The high-dimensional biological networks can be modeled by one of the three main methods: Boolean networks, differential equations and stochastic models.

Boolean Networks

The Boolean models are mainly used for explaining the biochemical systems due to its underlying property that it has proved itself in the large biochemical networks by reaching a promising performance covering the overall behavior of the system. The expression level of each gene is referred as the state, and each state is considered as functionally related to the state (gene expression level) of other genes by using the binary logic.

A biochemical system can be viewed by a Boolean model mainly by two ways: the truth table and the finite-state machine. The truth table includes three Boolean operators which are AND, OR and NOT. To illustrate, AND denotes both p and q are assigned by 1, while the OR rule implies one of p and q must be assigned by 1.

Table 2.1.: The truth table of Boolean operators in which i and j stand for the variables.

i	j	i AND j	i OR j	NOT j
1	1	1	1	0
1	0	0	1	0
0	1	0	1	1
0	0	0	0	1

The finite-state machine demonstrates a system in terms of its states and transitions. This model can be viewed as a diagram in which the nodes correspond to the states while the edges refer to the transitions. The states can be represented either by "current" term or by "next" term. As the names are obvious, current denotes the present state of the nodes; whereas next stands for the next state of the corresponding node i.e., gene, protein etc.

Even though the Boolean model is suitable for the high-dimensional networks since the use of binary logical operators simplifies the structure of the large data, still more complex models satisfying different expression levels are required (Ayyıldız, 2013).

Differential Equation Models

It is previously mentioned that the expression levels of the genes are considered as the states of the gene regulation systems. Since the expression levels are time-dependent and continuous, it is possible to compute the rates in terms of the ordinary differential equations considered as responsible for controlling the steady-state behavior of the model. Additionally, the ordinary differential equations are deterministic since they control the steady-state behavior of the system. Therefore, their use is not suitable when the system results in more than one outcome (Bower & Bolouri, 2001).

Stochastic Models

Instead of controlling the steady-state behavior of the gene regulatory systems, in stochastic models, the changes in the states (gene expression levels) are investigated in the discrete form which was the continuous form in the differential equation models (Bower & Bolouri, 2001).

2.2. Classification and Regression Trees (CART) Methodology

Classification and Regression Trees are used in order to construct decision trees from the learning samples (historical datasets), including prior classes for each observation in the dataset, for classifying the resulting data. CART is a binary and recursive method that processes by asking binary answered questions and constructing the tree by adding the nodes additively. Here, each node is assigned as a class and then parent nodes are divided into child nodes, followed by each child node to become a parent node. This procedure continues until there is no observation left to assign. These classification and regression trees can have highly complex structures as well as they could have quite simple ones.

2.2.1. Steps of CART Analysis

The basic steps of Classification and Regression Tree analysis can be argued as tree construction by splitting the nodes with respect to splitting rules and each node is considered as a predicted class based on its distribution; stopping the growth of the tree and it probably would overfit and require another procedure; tree pruning in which simpler trees are produced from the raw one and selection of the best tree that fits the requirements.

2.2.2. Tree Construction

Let us say t_p is the parent node and $t_l$and t_r are its left and right child nodes, respectively. Assignment of the children nodes is done with respect to the probabilities from the distribution of the data.

2.2.2.1. Splitting Rules

All the observations in the dataset are included in this first step— tree construction. Since the selection of the splitting rule depends on the type of the tree, whether it will be classification or regression tree, this constitutes one of the key points of this tree construction step.

2.2.2.1.1. Splitting Rules in Classification

The splitting rule differs according to the type of the tree. In the case of a classification tree, the response variable includes classes in a binary form. Dealing with the classes, the splitting can be performed mainly by impurity functions or the Twoing rule. The former, impurity functions, can be defined via Entropy, chi-square, misclassification rate, maximum deviation, and Gini rules. Among these alternatives, the Gini rule is the most preferred one and its basic mathematical explanation can be found via Equation (2.1.).

$$i(t) = 1 - \sum_{k=1}^{K} p^2(k|t). \tag{2.1.}$$

In Equation (2.1.), k refers to the class in the interval $[1, K]$ and $p(k|t)$ denotes the conditional probability of k of node t.

Additionally, the splitting is done with respect to the change of this Gini impurity function by solving the maximization implied via Equation (2.2.).

$$arg \max_{x_j \leq x_j^R} \Delta i(t) = - \sum_{k=1}^{K} p^2\left(k\middle|t_p\right) + P_l \sum_{k=1}^{K} p^2(k|t_l) + P_r \sum_{k=1}^{K} p^2(k|t_r). \tag{2.2.}$$

Here, in Equation (2.2), $x_j \leq x_j^R$ can be considered as the best split question, where $j \in [1, M]$ and M denotes the number of variables in the variable matrix X.

Apart from the impurity functions in the splitting procedure, the latter option, the change in the Twoing rule, let us imply it via T, can be used as demonstrated in Equation (2.3.).

$$\Delta T = \frac{P_l P_r}{4} \left[\sum_{k=1}^{K} |p(k|t_l) - p(k|t_r)|\right]^2. \tag{2.3.}$$

The splitting can be performed via Twoing rule by maximizing the change of Twoing function as shown in Equation (2.4.), where x_j^R refers to the best splitting value.

$$arg \max_{x_j \leq x_j^R} (\Delta T) \tag{2.4.}$$

Among these splitting rules in classification, the most commonly used ones are Gini and Twoing rules. In comparison, when the focus is to build more balanced tress, the Twoing performs better; however, it works highly slower than Gini. Therefore, in the case of obtaining the classification trees in limited time, the choice of Gini rule would be more beneficial.

2.2.2.1.2. Splitting Rule in Regression

Regression trees require a different splitting rule since the response variable we are dealing with is not a class, but a numeric or continuous variable.

Regression trees do not have classes. To predict, now we have response values for each variable, which are located in the variable matrix X. Therefore, splitting in the regression trees is performed by Squared Residuals Minimization Algorithm. This algorithm minimizes the summation of the expected variances of the resulting nodes. Accordingly, solving the minimization problem in Equation (2.5.), the splitting is completed.

$$arg \min_{x_j \leq x_j^R} [P_l Var(Y_l) + \ P_r Var(Y_r)]. \tag{2.5.}$$

In Equation (2.5.), $Var(Y_l)$ and $Var(Y_r)$ refer to the variances of the response vectors for left and right children of the parent node, respectively.

2.2.2.2. Node Splitting

In each iteration of the tree construction procedure, the data are divided into smaller parts, in which parent nodes are divided into two children nodes. To illustrate, the division of the parent node is performed in terms of the probabilities of the parent node to be divided into the left child node defined as P_{Left} and the same parent node to be divided into the right child node demonstrated as P_{Right}. So, the division results in the selection of the best splitting value, x_j^R, computed according to the previously defined probabilities, where x_j is the variable j from the variable matrix X.

The CART algorithm considers all the possible split values belonging to all variables and decides the best split via $x_j \leq x_j^R$, and maximizes the change of impurity or Twoing function in classification, and minimizes the change in the squared residual minimization algorithm function in regression.

Furthermore, the assignment of nodes as classes can be performed via the criteria defined in Equation (2.6.).

$$\frac{C(j\backslash i)\pi(i)N_i(t)}{C(iJay)\pi(j)N_j(t)} \geq \frac{N_i}{N_j}. \tag{2.6.}$$

Here, $C(j\backslash i)$ implies the cost of classification of i as j, $C(i\backslash j)$ demonstrates the cost of classifying j as i; $\pi(i)$ and $\pi(j)$are the prior probabilities of i and j, respectively; N_i and N_j are the number of classes i and j in datasets, respectively; $N_i(t)$ and $N_j(t)$ are the number of classes i and j in node; and this equation is written for all values of the class j (Lewis, 2000).

2.2.3. Stopping the Tree Growth

In the first step as the tree construction, the tree could grow out of control since it needs to keep growing until it is impossible to continue (possible reasons to stop are no observation left; all observations constitute exactly the same distribution; and where to stop is previously defined in the algorithm. Therefore, most of the time it overfits. In order to solve this problem, a pruning procedure must be performed later (Lewis, 2000; Timofeev, 2004).

2.2.4. Tree Pruning and Selection of the Best Tree

Simpler tree sequences must be defined in order to overcome the overfitting problem in the constructed tree. Therefore, the calculation of the cost of the complexity and the optimization are necessary. The pruning procedure can be applied by one of the following two ways: (1) optimization by minimum number of points, (2) cross-validation.

In the former way, a required number, N_{Min}, is defined as prior; and then, when the number of observations in the specified node becomes less than N_{Min}, the splitting process is stopped. The latter, cross-validation, fully depends on defining an optimal proportion between the complexity and the misclassification error (the trade-off in the first way was between impurity and complexity). As the complexity increases, the misclassification error exhibits a decrease and it approaches to zero when the complexity is on its maximum level. It is not preferred commonly due to its time consuming disadvantage.

Finally, as a result of the performed steps, each observation in the learning sample is assigned as a class for a classification tree or as a response value for a regression tree.

Each of these assigned observations is now considered as a class or a response value for the new observations from the new dataset. There exist split questions similar to the previously defined ones (represented via $x_j \leq x_j^R$) for the new observations and based on the best split question, each of the new observations should be assigned to a class or a response value (Timofeev, 2004).

CHAPTER 3

3. MODELING

3.1. Gaussian Graphical Model in the Literature

The basis of the Gaussian graphical model (Whittaker, 2001) is one of the most popular distributions, the Gaussian distribution. Since it is widely used not only in statistics but also in many global applications, it is a trustable approach in different types of modeling.

Biological entities tend to interact with each other in several ways such as activation or inhibition, causing biological incidents and many disease pathways. Explanation of their relationship requires the use of graphs, and accordingly, networks. There exist different approaches to infer the presence or lack of interaction among biological entities. Some of them are directed graphs while some of them are not directed and not weighted. The former type is commonly used in gene regulatory networks; whereas, the latter, e.g., protein-protein interaction network, is quite enough to display the existence or absence of the relationship among its vertices, also called nodes. Gaussian graphical model is one of the most suitable approaches for this type of biological networks. Considering the system has p nodes referring the proteins, it is assumed in GGM that the system represented by a random vector $Y = (Y^{(1)}, \dots, Y^{(p)})$ displays a multivariate normal distribution ($Y \sim N(\mu, \Sigma)$) with the mean vector $\mu = (\mu_1, \dots, \mu_p)$ and the $(p\times p)$-dimensional variance-covariance matrix $\Sigma = (\sigma_{ij})_{ij}$, where σ_{ij} denotes the variance of $Y^{(i)}$ and $Y^{(j)}$ if $i = j$; otherwise, it is the covariance.

In a simple graph, the biological entities such as genes, proteins are represented via nodes, and their interactions are demonstrated by directed or undirected edges depending on the type of biological process. In the basis of the Gaussian graphical model, undirected edges are created among vertices of the system. Additionally, the Gaussian graphical model assumes that the absence of an edge between two nodes in the graph refers to the conditional independence, meaning that given all the other vertices in the system, the two nodes are conditionally independent of each other, causing a zero partial correlation between these corresponding nodes in the variance-covariance matrix (Whittaker, 2001). Though those conditionally independent vertices have zero partial correlation with each other, they can still exhibit a high correlation if they are related with each other only by another node in the system. Therefore, the dependence and the independence between nodes are directly controlled by the inverse of the variance-covariance matrix Θ, which is called the precision matrix. The conditional independence between nodes is represented by zero entries (no partial correlation) in this precision matrix, causing the network exhibiting the direct relation between corresponding vertices (Whittaker, 2001).

3.1.1. Graphical Models via Regression

The main approach in the Gaussian graphical model is to derive the partial correlations from the precision matrix Θ, which can be inferred by a number of alternative techniques, such that GGM can be modeled by different regression functions by regressing each node against all the other nodes in the system. This approach can be used to optimize the maximum likelihood; and it enlarges the solution sets for the high-dimensional data (Whittaker, 2001).

Regressing a selected node (let us choose the last node, p, in the system) against all the other nodes can be formularized by Equation (3.1.).

$$Y^{(p)} = \beta Y^{(-p)} + \epsilon. \tag{3.1.}$$

Here, $Y = (Y^{(-p)}, Y^{(p)})$, where Y is the joint multivariate Gaussian vector, and $Y^{(-p)} = (Y^{(1)}, \dots, Y^{(p-1)})$ implies all the nodes but the last one, p, in the system. Furthermore, β is the regression coefficient, which controls the conditional independence of the nodes. Lastly, ϵ represents the error representing a multivariate normal distribution with mean μ and variance Σ. In order to demonstrate all the nodes by the last one, the mean vector and the variance-covariance matrix should be partitioned. For this purpose, the following illustrations shown in Equation (3.2.) are made (Meinshausen & Bühlmann, 2006).

$$\mu = \begin{pmatrix} \mu_{(-n)} \\ \mu_{(n)} \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} \Sigma_{-n,-n} & \sigma_{-n,n} \\ \sigma_{-n,n}^{t} & \sigma_{n,n} \end{pmatrix}. \tag{3.2.}$$

Here, $\boldsymbol{\mu}_{-\mathbf{p}}$ denotes the mean entries apart from the mean of the pth node, and $\boldsymbol{\mu}_{\mathbf{p}}$ shows the mean of the pth node. Accordingly, $\boldsymbol{\Sigma}_{-\mathbf{p},-\mathbf{p}}$ presents the covariance structure where the pth node is excluded, $\boldsymbol{\sigma}_{\mathbf{p},\mathbf{p}}$ indicates the variance of the pth node and finally, $\boldsymbol{\sigma}_{-\mathbf{p},\mathbf{p}}$ refers to the covariance between the pth node and the remaining nodes. Therefore, μ is a p-dimensional mean vector and Σ describes the $(p\times p)$-dimensional variance-covariance matrix. Additionally, the conditional distribution of the last node, p, on all the other vertices in the system can be exhibited as in Equation (3.3.).

$$Y^{(p)} | \, Y^{(-p)} = y \sim N(\mu_p + \left(y - \mu_{-p}\right)^t \sum\nolimits_{-p,-p}^{-1} \sigma_{-p,p}, \sigma_{p,p} - \sigma_{-p,p}^t \sum\nolimits_{-p,p}^{-1} \sigma_{-p,p}). \tag{3.3.}$$

Let us assume a point j from the given sample $(1, \dots, p)$. If the regression coefficient β in this specified point equals to zero ($\beta_j = 0$), then, in this situation, it can be suggested that the last node p and the node j are conditionally independent given the rest of the vertices in the model.

The regression coefficient β can be demonstrated based on the precision matrix as can be seen below (Wit et al., 2010).

$$\beta = -\theta_{-p,p} / \theta_{p,p}. \tag{3.4.}$$

In Equation (3.4.), similar to the previous explanation, $\theta_{-p,p}$ implies the precision between the pth node and the remaining nodes, and $\theta_{p,p}$ represents the precision of the pth node itself. Based on this formula, it can be said that the regression coefficient determines the structure of the precision matrix.

3.1.2. Strength of the Protein Regulation and Inference of the Gaussian Graphical Model from the Data

Under a known structure where the edges are defined between two corresponding biological entities, the strength of the interaction between two nodes of a graph is directly measured with the precision matrix of a multivariate vector. Considering the random sample $Y_1, Y_2, \dots, Y_n$ with size n of the multivariate vector Y, the joint density function of Y can be written based on observation y_i, $i \epsilon [1, n]$ as illustrated in Equation (3.5.).

$$f(y_i;\, \mu, \Sigma) = (2\pi)^{-n/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(y_i - \mu)^T \Sigma^{-1} (y_i - \mu)\right). \tag{3.5.}$$

In this equation, $(\,.\,)^T$ and $|\,.\,|$ denote the transpose and the determinant of the given term, respectively. Thereby, from Equation (3.5.), the likelihood is presented via $L(\mu, \Sigma) = \prod_{i=1}^{n} f(y_i;\ \mu, \Sigma)$, the log-likelihood can be written via Equation (3.6.).

$$l(\mu, \Sigma) = \log\big(L(\mu, \mathit{\Sigma})\big) = -\tfrac{n}{2} log|\Sigma| - \tfrac{1}{2}\textstyle\sum_{i=1}^{n}(y_i - \mu)^T \Sigma^{-1}(y_i - \mu). \qquad (3.6.)$$

Equation (3.6.) is based on a constant, which is independent of μ and Σ. Since the deal is to illustrate the strength with respect to the precision matrix, the log-likelihood can be displayed by Θ as shown in Equation (3.7.).

$$l(\mu, \Theta) = \frac{n}{2} log|\Theta| - \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^T \Theta (y_i - \mu). \qquad (3.7.)$$

Here, the log-likelihood in terms of the precision matrix can be written as a log-likelihood function by replacing μ with its maximum likelihood estimate $\bar{y}$, as given below (Friedman et al., 2008).

$$l(\Theta) = \tfrac{n}{2} log|\Theta| - \tfrac{n}{2} Trace(S\Theta). \qquad (3.8.)$$

In Equation (3.8.), $Trace(\,.\,)$ implies summation of the diagonal entries of $S\Theta$ and $S = (s_{ij})_{ij}$ is the sample covariance matrix, where s_{ij} is defined as in Equation (3.9.).

$$s_{ij} = \frac{1}{n}\sum_{k=1}^{n}(y_k^{(i)} - \bar{y}^{(i)})(y_k^{(j)} - \bar{y}^{(j)}). \qquad (3.9.)$$

Traditionally, a fully connected graph whose all nodes are connected to each other can be the deal of the defining the system via the inverse of the variance-covariance, precision matrix. In this kind of situations, the maximum likelihood estimate of the covariance matrix is represented by the symbol $\hat{\Sigma} = S$, causing the maximum likelihood estimate of the precision matrix to be denoted by $\widehat{\Theta} = S^{-1}$. In contrast, in the absence of some defined interactions, the aim becomes to find the precision matrix Θ maximizing the log-likelihood ($l(\Theta)$) with respect to the zero constraints already defined by the known structure of the graph. However, there exists a challenge that most of the time, not only the strength of the interactions but also the structure of the network are not known at the beginning, causing the necessity of the inference of the general model with its edges and strength of these edges from the data. For this purpose, five major different approaches are presented in the literature (Friedman et al., 2007; Friedman et al., 2008; Friedman et al., 2015; Meinshausen & Bühlmann, 2006; Tibshirani et al., 2005; Witten & Tibshirani, 2009; Yuan & Lin, 2007; Zou, 2006; Zou & Hastie, 2005). These are

- Maximum likelihood approach,
- Shrinkage approach,
- Lasso-based approach,
- Graphical lasso approach with L_1 penalized likelihood and
- Low-order partial correlations approach.

Maximum Likelihood Approach

Basically, the maximum likelihood approach can be regarded as the most traditional method in the literature and it is used to infer the Gaussian graphical model by maximizing the likelihood of the data. Considering the log-likelihood function of a GGM given by the corresponding equation and the estimate of the precision matrix maximizing this function, partial correlations can be easily obtained. The decision of the partial correlations that are significantly different than zero denoting the existing edges between nodes in the graph is made as the final model selection stage. The theory states when the zero entities are present as the true partial correlation. Accordingly, the partial correlations estimated from the sample covariance matrix are distributed with respect to Equation (3.10.).

$$f(r,k) = (1 - r^2)^{(k-3)/2} \frac{\Gamma\binom{k}{2}}{\sqrt{\pi}(\Gamma\binom{k-1}{2}}. \tag{3.10.}$$

In Equation (3.10.), r shows the partial correlation coefficient in univariate dimension and $\Gamma(.)$ indicates the gamma function. Moreover, the degree of freedom is denoted by $k = n - p - 1$, under the sample size n and the number of variables p.

Additionally, a z transformation applied to the partial correlations can be regarded as an alternative of the Equation (3.10.), which leads to an approximate test based on the normal distribution.

$$z(r) = \frac{1}{2}\sqrt{n-p-1}\ln\left(\frac{1+r}{1-r}\right). \tag{3.11.}$$

In Equation (3.11.), n is the sample size and p is the number of variables.

Another method is named as the likelihood ratio (LR) test and it can be used as denoted in Equation (3.12.) which has an asymptotic χ_1^2 distribution.

$$LR(r) = -n\log(1-r^2). \tag{3.12.}$$

Finally, in order to discard partial correlations by testing the significant difference between entities different than zero of the regression coefficients and the corresponding graphical model, the graphical models are viewed as the regression-based.

In general, the networks can be inferred in terms of the partial correlations, which are significantly different than zero by using one of these defined tests. However, it is not preferred to use the maximum likelihood solution reached from the given method with the deal of the regulatory network inference. The biggest disadvantage can be regarded as a fully connected graph where all the possible nodes are connected to each other, causing the precision matrix to consist of only ones. This situation may result in the absence of the inverse of the covariance matrix (Whittaker, 2001).

Shrinkage Approach

In large-scale biological networks, the main idea is to improve the estimation of the covariance matrix, accordingly, the precision matrix. Considering the estimate of the covariance matrix is defined by Equation (3.13.), λ has a key role in the level of shrinkage.

$$S^* = \lambda T + (1-\lambda)S_u. \tag{3.13.}$$

In Equation (3.13.), $S_u = \frac{n}{n-1}S$ is the unbiased sample covariance, T denotes the low dimensional target having smaller number of parameters and $\lambda \epsilon [0,1]$ implies the shrinkage intensity.

Here, the challenge is to propound an exchange solution between an unbiased estimator (sample covariance) with a large variance and an estimator with a lower variance with a high bias. The importance of λ controlling the shrinkage level arises from its role in the estimate of the precision matrix. In the interval of [0,1] for λ, the zero value results in an equivalence of the estimator with exactly the sample covariance. However, as the value of λ increases to one, the estimator becomes more shrinked. In an attempt to obtain better statistical properties, the implementation of the regularized estimator S^* can be preferred instead of using an unbiased estimator with a large variance or applying an estimator with lower variance, but larger bias. In the literature, the use of a combination of possible targets demonstrated by the equation is suggested since it provides a balance between a simpler target and more complicated targets, eliminating the problems arising from both unequal and off-diagonal covariance.

A second deal here is to identify the optimal shrinkage density λ. In the literature, the choice of this optimal shrinkage density is the minimization of the mean-squared error that is defined by Equation (3.14.).

$$R(\lambda) = E\left[\sum_{i,j}\left(\lambda t_{ij} + (1-\lambda)_{s_{u_{ij}}}-\sigma_{ij}\right)^2\right]. \qquad (3.14.)$$

Considering Equation (3.14.) and any target of T, the solution for the estimation of the density minimization can be easily determined. A function minimizing the mean-square error can be written in terms of the target equation. First, the loss function for the mean square needs to be defined as in Equation (3.15.) in which $E(.)$ is the expectation.

$$R(\lambda) = E\left[\sum_{i,j}\left(\lambda t_{ij} + (1-\lambda)_{s_{u_{ij}}}-\sigma_{ij}\right)^2\right] + E\left[\sum_{i}\left(s_{u_{ii}} - \sigma_{ij}\right)^2\right]. \qquad (3.15.)$$

Second, the equation can be minimized with respect to λ as given in Equation (3.16.).

$$\lambda = \frac{\sum_{i \neq j} var(s_{u_{ij}})}{\sum_{i \neq j} E[s_{u_{ij}}^2]}. \tag{3.16.}$$

According to Equation (3.16.), it is expected that λ would increase as the variance of the sample covariance increases. By replacing the sample mean and variance with their actual values, the real shrinkage intensity value λ can be calculated.

Here, the shrinkage estimator of the covariance matrix can be computed by the target T and the shrinkage intensity value λ; and accordingly, the precision matrix and partial correlations can be obtained. As the final stage, the decision which partial correlations have values significantly different than zero needs to be made. The use of the shrinkage approach requires different statistical tests than the tests used in the maximum likelihood approach in order to estimate the covariance matrix. The partial correlations exhibit different sampling distributions in the use of the shrinkage approach. However, in the literature, it is numerically argued that the sampling distributions eventually exhibit the same form as it is in the maximum likelihood approach under different degrees of freedom, and these distributions can be estimated from the data. The suggested method is argued to be more suited under $n << p$, and it enables to eliminate the partial correlations that the shrinked covariance matrix provides (Wit et al., 2010).

Lasso-based Approach

The biological systems tend to exhibit highly sparse structures. Under high dimensions, several approaches have been proposed for estimating these sparse networks from the data. In order to make the situation more obvious, the number of parameters, which are expected to be estimated can be reduced. The Lasso-based approach is specifically used in the inference of sparse networks. As previously mentioned, it is possible to view the system with respect to the regression models by regressing each node in the graph against all the other vertices. In this approach, the key point is to represent the precision matrix in terms of the matrix of the regression coefficients β. Therefore, the conditional independence among nodes can be represented either by the precision matrix or by the regression coefficients, which eventually corresponds to the same solution. The regression models are also useful for understanding the structure as well as for estimating the partial correlations. Inferring the networks by the regression models also provides well approximations. However, they carry the problem that they do not always result with a symmetric variance-covariance matrix, which is one of the most important requirements in the network inference procedure. In the shrinkage approach, by minimizing the shrinkage intensity, the whole model is forced for the sparsity. However, this approach has the advantage of imposing sparsity to each node in the system. Considering a system having p vertices, for predicting the last node p, the regression model can be constructed as Equation (3.17.).

$$Y^{(p)} = Y^{(-p)}\beta + \varepsilon. \tag{3.17.}$$

In Equation (3.17.), ε denotes the independent and normally distributed error term, and β implies the regression coefficient. Commonly, the regression coefficient β is defined by a least-square criterion. However, in the lasso-based approach, to impose the sparsity to the model, the regression coefficients are limited with respect to a L_1-penalty value λ, which is demonstrated via Equation (3.18.).

$$\left|\left|\beta\right|\right|_1 = \sum_i \left|\beta_{ip}\right| \; < \; \lambda. \tag{3.18.}$$

Accordingly, the regression coefficients are used as a solution for the following representation shown in Equation (3.19.).

$$min_\beta \left[\left|\left|Y^{(p)} - Y^{(-p)}\beta\right|\right|_2^2 + \lambda_p\left|\left|\beta\right|\right|_1\right]. \tag{3.19.}$$

In Equation (3.19.), λ_p denotes the tuning parameter of the representation. Since the larger value of λ causes larger number of zero coefficients in the estimated precision matrix, a sparse network can be regarded as a result of a larger penalty value λ; thus, a complex network is a consequence of a smaller λ.

On the other hand, as previously mentioned, the structure information can also be inferred by regression models, and a zero regression coefficient corresponds to no edge between two corresponding edges, enabling us to identify the connected and separate vertices in requires a final statistical test in order to reveal whether the non-zero elements are significantly different than zero. However, the lasso-based approach has the advantage that it results with exact zero regression coefficients in the precision matrix. Therefore, no further statistical analysis is needed to estimate the model. On the contrary, it comes with a disadvantage that is does not always provide a symmetric result which is when the node i is regressed against all nodes in such a way that it can exhibit a zero regression coefficient with the node j; however, when the node j is regressed against all the other vertices in the model, it may or may not result with a zero regression coefficient with the node i. In the procedure of the network inference, the symmetric results are highly preferred due to the definition of the precision matrix. Therefore, the lasso-based approach is open to argue due to its underlying disadvantage. In the literature, this problem is overcame by one of the two options which are the use of an AND rule assigning no edge if both the two regression coefficient states in Θ are zero; and the use of an OR rule assigning no edge if one of the two regression coefficient in Θ is computed as zero. Since the deal is to infer the biological network, the sparsity is the feature to consider. Hence, it can be clearly expressed that the OR rule results with a sparser network than the AND rule; which is the closer structure to a biological system. One additional situation here arises from imposing each individual node for the sparsity, causing the missing hubs. In biological systems, hubs are mostly faced with, constituting a high sparsity problem in this approach. One suggested approach as a solution for this problem is to create a boundary for the penalty value, which is defined in Equation (3.20.) in terms of the type 1 error α. As the alternative way, the optimal λ is detected from a distribution function as presented in Equation (3.20.).

$$\lambda_i = 2\sqrt{\frac{s_{ii}}{n}}\,\phi^{-1}\left(1-\frac{\alpha}{2p^2}\right). \qquad (3.20.)$$

In Equation (3.20.), ϕ denotes the cumulative distribution function of the standard normal density. This formula controls the penalty value by not allowing it to over α (Tibshirani, 1996; Whittaker, 2001).

Graphical Lasso Approach with L_1-Penalized Likelihood

It is previously mentioned that the biological networks constitute sparse structures. Therefore, imposing sparsity while inferring these networks is one of the most important issues. The Lasso-based approach is propounded as providing this advantage by imposing sparsity to each node in the system; however, it has argued as not always providing a symmetric result. The graphical lasso, also called glasso, approach is a technique satisfying both the sparsity and the symmetric estimation of the precision matrix in the network inference by only assigning the L_1-penalty to the elements of the precision matrix directly, instead of assigning this penalty value to the regression coefficients at each node. Hereby, the optimization can be represented via Equation (3.21.).

$$max_{||\Theta||_1 \leq \rho}[log|\Theta| - Trace(S\Theta)]. \qquad (3.21.)$$

In Equation (3.21.), $||\Theta||_1 = \sum_{i,j}|\theta_{ij}|$ and ρ implies a non-negative tuning parameter. Then, the penalized likelihood optimization can be demonstrated via Equation (3.22.).

$$max_{\Theta}\left[log|\Theta| - Trace(S\Theta) \ - \ \lambda||\Theta||_1\right]. \qquad (3.22.)$$

In Equation (3.22.), λ is the non-negative Lagrange multiplier. The optimal solution of the zero value for λ refers to the maximum likelihood estimation; however, as λ increases, sparser networks are constructed, but the associated likelihood decreases. In some situations such as in high dimensional problems, the number of observations is much lower than the number of variables. In order to solve this challenge and to estimate a more stable precision matrix, the penalized log-likelihood is maximized iteratively at each node by demonstrating the situation with respect to a lasso regression problem. Here, a matrix for different penalty values is created and these penalty values are assigned for different elements of the precision matrix. Then, the following inequality is detected as illustrated in Equation (3.23.).

$$max_{\Theta}\left[log|\Theta| - Trace(S\Theta) - \ ||\Theta * \Lambda||_1\right]. \qquad (3.23.)$$

In Equation (3.23.), $\Lambda = (\lambda_{ij})_{ij}. \lambda_{ij} = \lambda_{ji}$ denotes the matrix consisting of different penalty values for the distinct elements of the precision matrix (Friedman et al., 2014; Friedman et al., 2008).

Low-order Partial Correlations Approach

The large-scale problems in the inference of the Gaussian graphical model can be eliminated by the use of low-order correlations as the estimators of full-order correlations constituting GGMs. For this purpose, rather than calculating the correlation between two variables given the rest in the network, any two variables can be selected given a subset of variables among all the subsets, and the correlation of the two selected nodes can be computed in order to obtain the full-order correlations. Equation (3.24.) demonstrates the propounded approach.

$$\pi_{ij|k} = \frac{\rho_{ij} - \rho_{ik}\rho_{jk}}{\sqrt{(1-\rho_{ik}^2)(1-\rho_{jk}^2)}}. \tag{3.24.}$$

Here, in Equation (3.24.), ρ implies the marginal correlations. This formula expresses that the first order correlation between two variables are calculated by a variable from the rest of the network at each individual time. Here, the correlation between the variables $Y^{(i)}$ and $Y^{(j)}$ is computed with respect to the variable $Y^{(k)}$. In second-order partial correlations, de la Fuente et al. (2004) and Veiga et al. (2007) suggest a decision for eliminating an edge between two nodes in the network; that is, if either one of the following conditions is true, the edge is discarded: The marginal conditions can be zero, and/or at least one of the first-order partial correlations is zero. Magwene & Kim (2004) propound a more restricted suggestion that in order to agree upon the presence of an edge between two corresponding nodes, both the marginal correlations and all the partial correlations obtained from the first-order should refer to the presence of the edge. Magwene & Kim (2004), Wille et al. (2004) and Wille & Bühlmann (2006) propose the use of the first-order partial correlations in which ρ is estimated both by the Pearson coefficient which is the traditional method and by the Spearman rank correlation as the alternative method, as well as considering the second-order partial correlation but no more. Castelo & Roverato (2009) come up with the idea considering up to qth-order partial correlations by applying a Monte Carlo method to provide computational efficiency for all dual combinations of variables in the network.

3.2. Random Forest Algorithm

3.2.1. Basis of the Random Forest Algorithm

Machine learning can be categorized into two subgroups, which are the supervised learning and the unsupervised learning. The supervised learning includes the classification and the regression, while the unsupervised learning includes the clustering. In order to explain the basis of the random forest algorithm, only the supervised learning, classification and regression, is considered. In the classification, the deal is to assign each observation to the correct subgroups whose elements are previously known. This process is called the classification (Breiman, 2001).

Boosting

Boosting is a method discarding the classification problems in which the weighting process of the successive trees depends mainly on the earlier predictors. A weighted vote is selected among all variables for the prediction. In the boosting, a strong ensemble is created in terms of all the other weak classifiers in order to produce a strong classifier committee. This definition leads to the explanation of a weak classifier. A weak classifier can be defined as not providing a better error rate than the random guessing provides. The data are modified a number of times, and to each version of the data is exposed to these weak classifiers sequentially, resulting in a weak classifier chain constituting the purpose of the boosting. Also, Adaboost can be put forward as an adaptive version of boosting not including any random elements, and creating a committee of trees by recalculating the weights of the old ensembles.

Overfitting Problem

In the standard classification trees, the branching process continues by splitting each node with respect to the best groups among all variables. In the selection of the best groups, the branching goes only with the measurements (mainly, generalization error, strength and correlation); however, it also goes with the noises of the measurements, causing the overfitting problem during the calculation.

Bagging

One approach to overcome the overfitting challenge is to choose each tree independently from the previously chosen ones in the forests. The bagging is an algorithm that provides such a solution. Hereby, it constructs each tree independently by selecting a bootstrap sample of the datasets and then it chooses the most voted tree for the prediction. Even though bagging reduces the variance very effectively, it is not totally enough to discard the overfitting problem since it causes bias during the variance reduction. Therefore, the adaptive bagging is suggested as the improved version of the bagging reducing both the variance and the bias effectively. Additionally, it increases the accuracy by improving the estimates of the main measurements (generalization error, strength and correlation) of combined ensembles of trees. Hence, it prevents the effect of the bagging error on the estimates of the measurements, which has a remarkable importance in the inference of biological systems. However, the adaptive bagging algorithm is designed to work well on large datasets, which is not always possible when the data come from biological experiments or personal health records.

Random Forest Algorithm

We have previously mentioned that the random forest accepts the CART methodology in its theoretical basis.

The random forest is a technique selecting the most voted class among the classes consisting of many generated trees. It is also defined as a combination of the tree-structured classifiers in the forest, meaning that it is the classifier of the classifiers. The random forest has superiority on bagging as it constructs each tree with respect to a different bootstrap sample and it applies its own classification as the tree construction method. As opposed to the standard trees, in RF, small subsets of predictors are created and under the random selection, each node is split by using the best one among these randomly chosen predictors. In RFA, as the forest becomes including more trees, an upper boundary for the generalization error (PE^*) is generated in order to prevent the overfitting problem without requiring large datasets. That is the reason why we do not face with such a problem in RFA while other algorithms may result in overfitting challenge.

$$P_{X,Y}\left(P_\Theta(h(X,\Theta) = Y) - max_{j\neq Y}P_\Theta(h(X,\Theta) = j)\right) < 0 \qquad (3.25.)$$

In Equation (3.25.), $max_{j\neq Y}P_\Theta(h(X,\Theta) = j)$ denotes the maximum of all the probability values among all the values of the classifier except for its value on the point Y, while $h(X,\Theta)$ denotes the classifier for the random vector X.

Since the main rule in RFA is to maximize the strength between nodes (in biological systems, they are proteins) having the lowest correlation, the adaptive bagging is a useful basis due to the reason that its well estimates of the main measurements of strength, correlation and generalization error with an additional process of limiting the generalization error by creating an upper boundary increase the accuracy of trees, motifs, modules and the resulting networks.

The strength of each individual tree and the correlation between combinations of these trees determine the generalization error of a random forest. Moreover, the use of the random feature selection to split each node results in an error rate comparable to the others. Furthermore, the generalization error, strength and correlation can be viewed by the internal estimates, and these estimates are used to exhibit the response to the increasing number of features involved in the splitting step. The internal estimates can also be applied to measure the importance of the variable. As it is mentioned, in random forests, small communities (ensembles) are created and these ensembles vote for the most popular class. In order to create these small communities, the most common way is to generate random vectors denoted by "Θ". In order to refer the kth tree in the forest, the random vector representing that specific tree is taken as Θ_k. In a forest, random vectors are $\Theta_1, \Theta_2, \dots, \Theta_{k-1}$, resulting in Θ_k and also resulting in a classifier $h(X, \Theta_k)$, where X is an input vector and k also represents the convergence of trees in the forest. These vectors control the growth of each tree in the small communities. There are ways to generate these random vectors, such as bagging, random split selection, and selecting the training set from a random set of weights. On the other hand, the accuracy of a random forest can be defined in terms of the generalization error, the strength of the individual tree classifiers and the dependence measure correlation between these classifiers. Hence, the generalization error (PE^*) of RFA is controlled by Equation (3.26.).

$$PE^* \leq \frac{\bar{\rho}(1 - s^2)}{s^2}. \qquad (3.26.)$$

In this inequality, the mean value of the correlation between random vectors Θ and Θ' is shown via $\bar{\rho}$, while Θ' is the proposal tree in the next iteration. Here, s denotes the strength of the small communities via $s = E_{X,Y} mr(X, Y)$. E implies the expectation between the random vectors X and Y, whereas, the margin function is indicated by $mr(.)$.

There exist some key points in the random forest algorithm that need to be known how they are generated, in order to construct the trees accurately. For example, the strength and the correlation play key roles in the random forest algorithm as much as the generalization error.

3.2.2. Accuracy Definitions of Random Forest Algorithm

Previously, it is defined that a classifier demonstrating a resulting random vector Θ_k can be exhibited as $h(x, \Theta_k)$. Here, creating an ensemble of classifiers, the representation of this committee will be $h_1(x), h_2(x), \dots, h_K(x)$. By using these given properties, the margin function can be defined as follows:

$$mg(X,Y) = av_k I(h_k(X) = Y) - max_{j \neq Y} av_k I(h_k(X) = j). \qquad (3.27.)$$

In Equation (3.27.), Y, X are the random vectors $(Y = [1 \dots j])$, $I(h_k(X) = ..)$ is the indicator function and av_k is the average of the kth tree.

The upper boundary of the generalization error is mentioned previously. In order to expand it, now, it is defined how it is determined by the margin function. By extending it, the generalization error represented by the margin function can be shown as in Equation (3.28.).

$$PE^* = P_{X,Y}(mg(X,Y) < 0). \tag{3.28.}$$

In Equation (3.28.), the generalization error is represented by PE^*.

As the rule of the random forest algorithm, the number of trees should be increased. In this situation, for all the ensemble (Θ_1, …), the generalization error converges to the formula shown in Equation (3.29.) as previously defined.

$$P_{X,Y}\big(P_\Theta(h(X,\Theta) = Y) - max_{j\neq Y}P_\Theta(h(X,\Theta) = j) < 0\big). \tag{3.29.}$$

We argue that since the rule in the random forest is to maximize the strength and to minimize the correlation, the strength is one of the most important features in the random forest algorithm for the process of the tree construction. So, again by looking at the margin function, it is possible to define the strength of the ensemble of classifiers by Equation (3.30.).

$$s = E_{X,Y}mr(X,Y). \tag{3.30.}$$

Now it is appropriate to connect the generalization error and the strength as in following representation:

$$PE^* \leq var(mr)/s^2. \tag{3.31.}$$

In Equation (3.31.), PE^* is the generalization error and s^2 refers to the square of the strength.

If we bring the correlation into these equations, following representation can be made.

$$var(mr) = \bar{\rho}(E_{\Theta} sd(\Theta))^2 \leq \bar{\rho} E_{\Theta} var(\Theta). \tag{3.32.}$$

In Equation (3.32.), $\bar{\rho}$ is the mean value of the correlation between Θ and Θ', E represents the expectation, sd denotes the standard deviation and var implies the variance.

So, in order to define an upper boundary to the generalization error, all the previously defined formulas are combined and the result is seen in Equation (3.33.).

$$PE^* \leq \bar{\rho}(1 - s^2)/s^2. \tag{3.33.}$$

In Equation (3.33.), s is the strength as defined above.

Here, the ratio $\frac{c}{s^2} = \bar{\rho}/s^2$.

Now, it is explained how RFA works in the use of the random features with a random input selection and with linear combinations of inputs, respectively; and then, these two will be compared to Adaboost.

3.2.3. Ways for Random Forest Algorithm to Work

Random forests using random features

The randomly selected inputs constituting of the forests are collected to grow each tree. The characteristics of this procedure and its comparison with the adaboost are given below:

- Its accuracy is as good as other procedures,
- It is relatively robust to outliers and noise,
- It is faster than bagging and boosting,
- It gives useful estimates of error, strength, correlation and also variable importance and
- It is simple.

We calculate internal estimates in order to decide how many features will be selected in each node. These internal estimates belong to the generalization error, the classifier strength and the dependence, which are also called as the out-of-bag (OOB) estimates.

Various use of out-of-bag estimates can be listed as follows:

- OOB estimates are used as an ingredient in estimates of the generalization error,
- OOB estimates of variance are used to estimate the generalization error for arbitrary classifiers,
- It is proved that OOB estimate is accurate as using a test set of the same size as the training set,
- Using the OOB error estimate removes the need for a set aside the test set.

As mentioned previously, the use of random features can be categorized into two ways: the random input selection, and the linear combinations of inputs. In the random input selection, the keywords are the CART methodology and the Forest-RI, while in the linear combinations of inputs part, the keyword is the Forest-RC. In comparison, Breiman's results (2001) exhibit that the Forest-RI performs better than the adaboost, whereas, the Forest-RC works better than the Forest-RI.

Random forests using the random input selection

The simplest random forest created by random features is selected at random. At each node, it is needed to select a small group of input variables, and these variables then should be split on. The trees are growing by the use of the CART methodology, which is a method to do maximum size but do not prune. In the CART methodology, the main rule is that each node can have only two children. Then each child becomes nodes and has two children. This process continues, recursively. Figure (3.1) exhibits a visual representation of the CART methodology.

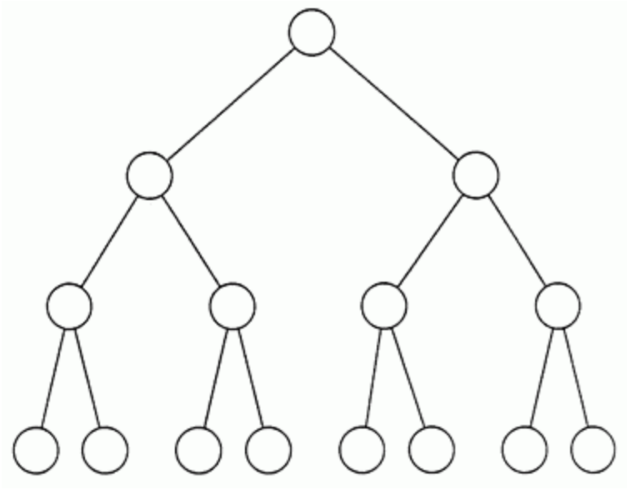

Figure 3.1.: Representation of classification tree construction.

Thus, this procedure is denoted by the Forest-RI, and the size of the group is fixed. For the size of the group, two values are tried, the first one used only one randomly selected variable, and the second one is expected to use the first integer less than $log_2(M+1)$, where M is the number of inputs. In almost all datasets, it is possible to observe that random forest uses 10% of the data as a test set, and the remaining as the training set. For the training set, RFA runs twice, one for F=1 and the other one is for $int(F) < log_2(M+1)$. Breiman's study (2001) argues that, when compared to the Adaboost, this procedure is considerably faster.

Random forests using the linear combinations of inputs

Considering M the number of inputs and F the fraction of M, under the condition that the number of M is only a few and we take F as a fraction of M in order to increase the strength. While increasing the strength, it also causes a higher correlation. Another way to solve this problem is to use a procedure called the Forest-RC. In this procedure, the random linear combinations of a number of input variables are taken in order to define more features. At that point, by specifying L, the features can be generated, leading number of variables to be combined. At each given node, L variables are randomly selected with coefficients, which are uniform random numbers in the interval $[-1, 1]$, and added together. After F linear combinations are generated, a search is performed for the best split. This process is called the Forest-RC.

In order to view the effects of the strength and the correlation on the generalization error, following explanations are done.

The main rule here is that, at each iteration, 10% of the data is splitted off as a test set. Keeping this rule in mind, F (the number of random inputs) is changed in between 1 and

50 at each node, then in order to form a random forest, 100 trees are grown for every 50 F, and terminal values of the set error, the strength and the correlation are recorded. Eighty iterations are performed and at each of these eighty iterations, 10% of the data is splitted as a test set. The results from these eighty iterations are averaged, and in total, 400.000 trees are grown in order to see the effects of the strength and the correlation. As a result, it is observed that both the strength and the correlation exhibit a small, but a steady increase in the applied study. According to the results, random forests with lower generalization errors have higher strength and a lower correlation, making them better. In comparison with the Adaboost, the forest-RC seems better than the forest-RI.

CHAPTER 4

COPULAS

In many biological systems, expression levels of the genes may exhibit non-normal and non-linear behavior with strong tail dependence; and, most of the time, they are not suited for the use of normality-based models. In such cases, copula modeling provides a suitable solution for the construction of the joint distributions. Under the definition of the marginal distributions, the use of copulas can be regarded as a very suitable method for inferring the joint distributions of the non-normal variables (Trivedi & Zimmer, 2007).

4.1. Basic Definition of Copula

The copulas are the joint distributions specified with respect to their parameters and generated in terms of the defined marginal distributions. Thereby, copulas reflect the properties of the joint distributions.

A basic joint distribution of a set of random variables, $(Y_1, \ldots, Y_m)$, can be demonstrated by Equation (4.1.).

$$F(y_1, \ldots, y_m) = P[Y_i \leq y_i; i = 1, \ldots, m]. \tag{4.1.}$$

From this equation, the survival function of $F(y_1, \ldots, y_m)$ can be written for $m = 1, 2, 3$ as it is seen in Equation (4.2) (Trivedi & Zimmer, 2007).

$$\begin{aligned}
\bar{F}(y_1, \ldots, y_m) &= P[Y_i > y_i; i = 1, \ldots, m] = 1 - F(y_1) \\
\bar{F}(y_1, \ldots, y_m) &= 1 - F_1(y_1) - F_2(y_2) + F_1(y_1)F_2(y_2) \\
\bar{F}(y_1, \ldots, y_m) &= 1 - F_1(y_1) - F_2(y_2) - F_3(y_3) + F_{12}(y_1, y_2) + F_{13}(y_1, y_3) \\
&\quad + F_{23}(y_2, y_3) - F(y_1, y_2, y_3).
\end{aligned} \tag{4.2.}$$

For another illustration of the copula definition functions, let us assume X and Y continuous random vectors, and any pair of (X, Y) under a joint cumulative distribution function (cdf), $H(x, y)$, is expressed by Equation (4.3.).

$$H(x,y) = C\{F(x), G(y)\}, \qquad (x, y \in R). \tag{4.3.}$$

In Equation (4.3.), $F(x)$ and $G(y)$ are the marginal distributions and $C: [0,1]^2 \rightarrow [0,1]$ is the copula (Genest & Favre, 2007).

4.2. Frechet-Hoeffding Bounds on Copulas

Considering the joint cumulative distribution functions with m variables under univariate marginal distributions $F_1, \ldots, F_m$, it can be clearly said that each of these marginal distributions will be located in the interval $[0, 1]$.

The lower and the upper boundaries for the joint cumulative distribution function can be defined as F_L and F_U by the Frechet-Hoeffding bounds as shown in Equations (4.4.) and (4.5.).

$$F_L(y_1, \ldots, y_m) = \max\left[\sum_{j=1}^{m} F_j - m + 1,\ 0\right] = W, \tag{4.4.}$$

$$F_U(y_1, \ldots, y_m) = \min[F_1, \ldots, F_m] = M. \tag{4.5.}$$

Accordingly,

$$W = \max\left[\sum_{j=1}^{m} F_j - m + 1,\ 0\right] \leq F(y_1, .., y_m) \leq \min[F_1, \ldots, F_m] = M. \tag{4.6.}$$

In Equations (4.4.), (4.5.) and (4.6.), the upper boundary always corresponds to a cumulative distribution function (cdf), and the lower boundary refers to a cdf for $m = 2$; however, for the lower boundary, in order to be a cdf for $m > 2$, there is a need for some other conditions.

Here, margins can be either univariate, bivariate or higher dimensional. In the former type of marginal distributions, the Frechet-Hoeffding boundaries are defined under the m-variate distributions: $\mathcal{F}(F_1, \dots, F_m)$. Otherwise, the boundaries defined under the classes such as $\mathcal{F}(F_{12},\ F_{13})$ and $\mathcal{F}\big(F_{12}, F_{13,}F_{23}\big)$ (Trivedi & Zimmer, 2007).

4.3. Sklar’s Theorem on Copula Functions

According to the Sklar’s theorem, a copula under m-dimension can be represented by a function C from $[0,1]^m$ to $[0,1]$. In order to conclude this statement, the following three conditions need to be satisfied:

1. $C(1, \dots, 1, a_n, 1, \dots, 1) = a_n$ for all values of a_n must be in the interval [0,1] and it is applicable for every $n \leq m$.
2. $C(a_1, \dots, a_m) = 0$ if $a_n = 0$ for any $n \leq m$.
3. C is m-increasing.

The first property states that, among m variables, if the marginal probability of all the $m - 1$ variables is 1, then the joint probability of the m outcomes is the same with the remaining ones.

According to the second property, which, in some situations, is also called the grounded property, if any of the outcomes results in a zero probability; then the joint probability of all outcomes becomes equal to zero.

Finally, the third property stands for the non-negativity of the C-volume of any m-dimensional interval.

The second and the third properties are common in multivariate cumulative distribution functions.

Considering these given definitions, it can be said that an m-dimensional distribution function can represent an m-copula with all its marginal distributions under $U(0,1)$.

A continuous distribution function with m variables $F(y_1, \dots, y_m)$ can view the relationship between distribution functions and the copulas with its univariate margins $F_1(y_1), \dots, F_m(y_m)$ and the inverse functions $F_1^{-1}, \dots, F_m^{-1}$. Afterwards, the following transformations can be done in order to represent marginal distributions and quantile functions in terms of uniformly distributed variables. Let us assume

$$y_1 = F_1^{-1}(u_1) \sim F_1, \dots, y_m = F_m^{-1}(u_m) \sim F_m,$$

in which $u_1, \dots, u_m$ denotes the uniformly distributed variates.

$$\begin{aligned} F(y_1, \dots, y_m) &= F\left(F_1^{-1}(u_1), \dots, F_m^{-1}(u_m)\right) \\ &= P[U_1 \leq u_1, \dots, U_m \leq u_m] \\ &= C(u_1, \dots, u_m). \end{aligned} \qquad (4.7.)$$

Equation (4.7.) represents the copula associated with the distribution function (Trivedi & Zimmer, 2007). Here, $y \sim F$ and $\left(F_1(y_1), \dots, F_m(y_m)\right) \sim C$, $U \sim C$. Then $\left(F_1^{-1}(u_1), \dots, F_m^{-1}(u_m)\right) \sim F$.

4.4. Different Copula Types

4.4.1. Gaussian (Normal) Copula

In the literature, it is suggested that the copula function can be proposed in terms of non-normal distributions with continuous variables for the model selectivity. The following equation demonstrates this situation.

$$\begin{aligned} C(u_1, u_2;\, \theta) = {} & \phi_G(\phi^{-1}(u_1), \phi^{-1}(u_2);\, \theta) \\ & = \int_{-\infty}^{\phi^{-1}(u_1)} \int_{-\infty}^{\phi^{-1}(u_2)} \frac{1}{2\pi(1-\theta^2)^{1/2}} x \left\{ \frac{-(s^2 - 2\theta st + t^2)}{2(1-\theta^2)} \right\} dsdt. \end{aligned} \qquad (4.8.)$$

In Equation (4.8.), ϕ refers to the cumulative distribution function of the standard normal distribution, $\phi_G(u_1, u_2)$ stands for the bivariate ϕ under the correlation parameter θ in the interval (-1,1).

It is analyzed that as the correlation parameter, also named as the dependence parameter, converges to the boundaries of its interval, the value of the Gaussian copula approaches to the Frechet-Hoeffding lower and upper boundaries, respectively. This flexibility of the Gaussian copula enables the dependence ranging from negative values to positive values equally, referring to the equivalent dependence (van Ophem, 1999).

4.4.2. Gumbel Copula

Let us assume that $-logu_j$ is denoted by $\tilde{u}_j$. Then, it is possible to write the Gumbel copula function as Equation (4.9.).

$$C(u_1, u_2;\ \theta) = \exp\left(-\left(\tilde{u}_1^{\theta} + \tilde{u}_2^{\theta}\right)^{\frac{1}{\theta}}\right). \tag{4.9.}$$

Lower and upper boundaries of the Gumbel copula are defined as 1 and ∞, where 1 and ∞ refer to the independence since they are also called the Frechet-Hoeffding bounds, and the interval $[1, \infty)$ defines the level of independence. No negative correlation is enabled in the Gumbel copula as it has a strong right tail but a weak left tail. Moreover, the Gumbel copula is suitable when the data are strongly correlated for high values while are weakly dependent for the lower values (Trivedi & Zimmer, 2007).

4.4.3. Clayton Copula

The defined interval for the Clayton copula is $(0, \infty)$. When the dependence parameter θ becomes closer to infinity, it can be mentioned that it approaches the upper boundary of the Frechet-Hoeffding bounds. However, in contrast to the Gumbel copula, the dependence would never converge to the lower Frechet-Hoeffding bound. The representation of the Clayton copula is given in Equation (4.10.).

$$C(u_1, u_2;\ \theta) = (u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}\ . \tag{4.10.}$$

The Clayton copula has a strong left tail and a weak right tail (Trivedi & Zimmer, 2007).

4.4.4. Frank Copula

The Frank copula has its dependence in the interval $(-\infty, \infty)$, where $-\infty$ and ∞ imply the Frechet-Hoeffding lower and upper bounds, respectively, and 0 denotes the dependence between marginal distributions.

$$C(u_1, u_2;\ \theta) = -\theta^{-1} \log\left\{1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1}\right\}. \quad (4.11.)$$

In the Frank copula, the negative dependence is allowed between margins and it has symmetric tails referring to the dependence. Hence, it is appropriate to infer the models having strong positive and negative dependence between marginal distributions (Trivedi & Zimmer, 2007).

4.5. Measure of the Dependence

Assuming the random pair of continuous variables (X, Y) sampled as $(X_1, Y_1), \ldots, (X_n, Y_n)$, $H(x, y)$ can be suggested as characterizing the behavior of their joint distribution. As stated previously, $F(x)$ and $G(y)$ are the corresponding marginal distributions of X and Y continuous variables providing their individual behavior; while the copula $C(x, y)$ is the copula function identifying their joint behavior of dependence.

Considering the Frechet-Hoeffding bounds, if the copula function C is one of the previously defined W or M, Y can be viewed as a monotone decreasing or increasing deterministic function of X.

$$W(u, v) = \max(0, u + v - 1) \text{ or } M(u, v) = \min(u, v). \quad (4.12.)$$

In the former situation, which is $C = W$, Y can be suggested as a monotonically decreasing function of X and when $C = M$, Y is regarded as a monotonically increasing function of X. Thereby, it can be said that the copula function C is ranged between these defined extremes from W to M, illustrated by Equation (4.13.).

$$W(u, v) \leq C(u, v) \leq M(u, v). \quad (4.13.)$$

Equation (4.13.) is written for all values of u, v in the interval [0,1].

Let us assume the rank pairs (R, S) ranging from 1 to n; denoted by $(R_1, S_1), \ldots, (R_n, S_n)$ in which the rank of X_i from the sample $(X_1, \ldots, X_n)$ is implied by R_i, and S_i refers to the rank of Y_i the continuous variable sample $(Y_1, \ldots, Y_n)$ and the ranks are assigned as the certain values rather than under continuity in order to exclude problems may arising from the zero probability for X and Y.

Redesigning the ranks of both R and S by 1/(n+1), the unit square $[0,1]^2$ is obtained. Accordingly, the empirical copula C_n can be written in terms of these redesigned ranks as shown below in Equation (4.14.).

$$C_n(u,v) = \frac{1}{n}\sum_{i=1}^{n} 1\left(\frac{R_i}{n+1} \leq u, \frac{S_i}{n+1} \leq v\right). \tag{4.14.}$$

Thereby, C_n can be regarded as a rank-based estimator of normally distributed C for any given (u,v) pair.

Among the rank-based estimators, Spearman's Rho and Kendall's Tau can be propounded as the most common two methods for measuring the dependence (Genest & Favre, 2007).

4.5.1 Spearman's Rho

The basis of the Spearman's Rho method calculates the correlation either between the rank pairs (R_i, S_i) of C or between $(\frac{R_i}{n+1}, \frac{S_i}{n+1})$ of C_n. Therefore, the equation of the Sperman's Rho can be written as in Equation (4.15.).

$$\rho_n = \frac{\sum_{i=1}^{n}(R_i-\bar{R})(S_i-\bar{S})}{\sqrt{\sum_{i=1}^{n}(R_i-\bar{R})^2 \sum_{i=1}^{n}(S_i-\bar{S})^2}} \in [-1,1]. \tag{4.15.}$$

In Equation (4.15.), $\bar{R}$ and $\bar{S}$ stand for the averages of ranks computed via $\frac{1}{n}\sum_{i=1}^{n} R_i$ and $\frac{1}{n}\sum_{i=1}^{n} S_i$, respectively (Genest & Favre, 2007).

4.5.2. Kendall's Tau

The basis of this approach is to compute the dependence in terms of the number of concordant pairs of (X_i, Y_i) and (X_j, Y_j) represented by P_n and the number of discordant pairs denoted by Q_n; and it is illustrated by Equation (4.16.).

$$\tau_n = \frac{P_n - Q_n}{\binom{n}{2}} = \frac{4}{n(n-1)} P_n - 1. \tag{4.16.}$$

In Equation (4.16.) a pair is called as concordant when $(X_i - X_j)(Y_i - Y_j) > 0$; otherwise, a pair is referred to as discordant when $(X_i - X_j)(Y_i - Y_j) < 0$. Here, it is obvious that these intervals do not include zero, which occurs under the zero probability when X and Y are considered as continuous variables. In order to make this situation more clear, a number of transformations can be suggested; however, it is not included here since in this study, the Spearman's Rho is chosen as the measure of dependence (Genest & Favre, 2007).

CHAPTER 5

APPLICATION AND RESULTS

5.1. Accuracy Measures

In our study, in order to evaluate and compare the results of both algorithms, firstly, the true positive (TP), true negative (TN), false positive (FP) and the false negative (FN) numbers are calculated, and based on these values, the precision, recall, F-measure, false positive rate (FPR), false discovery rate (FDR) and the Matthew's correlation coefficient (MCC) values are computed. Here, TP implies that an interaction found between two proteins as a result of the algorithm actually exists. TN denotes that an absence of an interaction between two proteins is not suggested as present at the end of the analysis. FP stands for the situation that the algorithm finds an interaction between two nodes where the interaction does not actually exist, and finally, FN illustrates that the algorithm misses an interaction that is present between two vertices in the real situation (Ayyıldız, 2013).

Table 5.1.: TP, TN, FP and FN values in terms of actual and predicted situations.

		Actual Situation	
		True	False
Predicted Situation	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

Among the accuracy measures, the precision is represented in terms of TP and FP values in the following equation.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}. \tag{5.1}$$

In the evaluation, higher values of the precision refer to better results in the inference of systems.

Similarly, recall, F-measure and specificity values can be computed as below.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}. \tag{5.2}$$

The recall value is also called as the True Positive Rate (TPR) of the analysis; therefore, higher values of TPR are preferred for correctly inferred networks.

$$\text{F} - \text{Measure} = 2\times\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5.3}$$

The F-Measure is one of the most accurate measures since it controls both the precision and recall. However, in some situations, zero TP causes zero recall and zero precision, resulting in an undefined F-measure value. When it is applicable, higher F-measure is preferred for the evaluation.

The false positive rate (FPR) can be computed as $1 - Specificity$ and the smaller value of FPR is preferred for performing accurate evaluations.

$$\text{FPR} = \frac{\text{FP}}{\text{TN+FP}}. \tag{5.5}$$

Similarly, the false discovery rate (FDR) can be written as $1 - Precision$ and it refers to the mislabeled positive ratio among all elements labeled as positive.

$$\text{FDR} = \frac{\text{FP}}{\text{TP+FP}}. \tag{5.6}$$

Finally, the balanced measure ranges from -1 to 1 and refers to the fully misclassification of the elements on the exact ratio of -1 and fully correct classification on 1 is represented by the Matthew's Correlation Coefficient (MCC). MCC is demonstrated by the following formula.

$$\text{MCC} = \frac{\text{TPxTN}-\text{FPxFN}}{\sqrt{(\text{TP+FP})(\text{TP+FN})(\text{TN+FP})(\text{TN+FN})}}. \tag{5.8}$$

5.2. Description of the Simulated Data

In this study, in order to perform our suggested algorithms on the data under the normality assumption, we use the *huge* package in the R programing language and generate scale-free data under different dimensions. The scale-freeness is one of the main topological features of the biological systems and it is related with the connectivity of the nodes (Barabási et al., 2011). The scale-free system implies that the network is very sparse and there are few nodes, called hubs, having high connections with other nodes and many nodes having just few links with other nodes.

5.2.1. Aim of Using Copulas

Dealing with the biological data, it is not always possible to face with the normality assumption. For example, the gene expression levels can exhibit a wide range of distributional varieties. Thus, the algorithms used in the description of biological networks should provide equal performances as much as possible on any kind of data in order to avoid the problems arising from the distributional characteristics of the data. Unfortunately, current techniques perform the analyses with respect to their specific basis without regarding features of the data. On the other hand, since we suggest a non-parametric model for overcoming these problems, we need to detect whether our proposed non-parametric algorithm (RFA) provides equal outputs on the data far away from the normality assumption as it provides on the normally distributed data. Furthermore, Gumbel, Frank and Clayton copulas, which are dependent on a single parameter, rather than a variance-covariance matrix denoting all pairwise correlations, do not have explicit solutions in high dimensional datasets (Wawrzyniak, 2006). Therefore, we apply the Gaussian Copula as it has explicit form for all dimensions and can describe positive defined covariance matrix. Hence, the Gaussian copula is performed with respect to exponential, student-t and log-normal marginal distributions in order to reveal the performance of the suggested algorithm under different joint distributions, and to compare the results of the GGM and RFA outputs under various measures of accuracies.

5.3. Application via GGM

In inference of the Gaussian graphical model (GGM) via the graphical lasso (glasso) from the biological data, the algorithm requires a penalty value, λ as stated in previous chapters. The choice of λ has a remarkable importance since it is directly responsible for the structure of the network. The lower value of λ causes a very complex system while higher value of the penalty results in very sparse networks. There exist a couple of criteria such as AIC (Akaike, 1973), BIC (Kass & Wasserman, 1995; Schwarz, 1978), RIC (Donoho & Johnstone, 1994; Foster & George, 1994) and StARS (Liu et al., 2010) in order to compute the optimal penalty value. In our analyses with glasso, we calculate the optimal λ according to the RIC criterion and compute our estimated precision via the glasso package in R (Friedman et al., 2015). Then, in order to convert the estimated precision to the adjacency matrix, we use 0.10 as the threshold value in all analyses based on GGM due to its suitability in biological systems. Later, in the evaluation, we apply the multivariate normal distribution and non-normal distributions generated via distinct copulas. Furthermore, to detect any change in the accuracy, we run the model under 10, 20, 50 and 100 number of genes/proteins, i.e., (10x10), (20x20), (50x50) and (100x100)-dimensional Θ, respectively. In all the calculations, we set the number of observations per gene/protein to 20, which is particularly reasonable for high dimensional systems. Moreover, in comparison of both estimated systems, we use a predefined structure of the population precision. This structure is directly obtained from the multivariate normal distribution via the huge package, whereas for non-normal data, we produce them originally as the following way.

— We generate a binary structure (0 and 1) of the network under scale-freeness. Then we defined our independent margins from fully exponential (with rate 5), fully student-t (with degree of freedom 5) and fully log-normal (with mean 10 and standard deviation 2) distributions. Afterward, we bind these margins and correlation structure by the Gaussian copula. Finally, we compute the recall, precision, F-measure, false positive rate (FPR), false discovery rate (FDR) and the Matthew's correlation coefficient (MCC) based on 1000 Monte Carlo runs from both approaches in order to observe the changes in accuracy.

5.4. Application via RFA

Since RFA is a non-parametric approach which does not deal with the variance-covariance matrices and the threshold values, the basis of RFA depends on whether there is an interaction between nodes or not (Breiman, 2001). Therefore, a number of iterations are performed depending on the size of the data and nodes, i.e., proteins, are bound to each other iteratively by maximizing the strength while minimizing the correlation. In this way, after the two closest proteins are bound to each other with respect to their instance values from the confusion matrix provided by RFA as an output at the end of each iteration, this process can continue either by completely binding two new proteins to each other, or by binding an individual protein to the previously bound nodes. The underlying iterations continue until there are no related proteins left to bind. Accordingly, in the resulting matrix, all the proteins can be bound to each other as well as there can be individual proteins which are not interacting any of the other proteins in the system. The latter system is called as a sparse network, which is highly common when the deal is the biological data.

In the application of RFA in R by using the package 'randomForest' (Breiman & Cutler, 2012), first, we create a symmetric and empty adjacency matrix whose row and column names are the protein labels. When RFA starts to proceed, it creates a confusion matrix consisting of instance values at the end of each run. Since our purpose is to have the maximum strength between two nodes with very few correlation, we accept the instance values in the confusion matrix as equivalent to the strength; and at each run, we aim to select the nodes having the maximum strength. After we choose the proteins from the confusion, we go back to the original data consisting of observations, which belong to the proteins, and then label them with a common name. Afterward, the new form of the data becomes exposed for another random forest run, and again, the couples having the maximum strength are chosen from the confusion and labeled together in the original observation data. This process continues until there is no remaining protein to bind. During this iterative process, at the end of each run, we record the protein names before labeling them together, jump into the adjacency matrix that we create at the beginning of the procedure and write "1" to the correct cells that they refer to. Constructing the network, we do not only cope with the protein pairs, but also with the motifs and the modules. This occurrence arises from the situation that, in a specific run, a protein shares the maximum strength with a previously bound couple, goes and binds to them and these new complex proteins constitute a structure together. In such cases, in order to find out from which protein in the couple (or motif, or module) the single one binds to, we check the first confusion matrix in which all the proteins stay single and search for the maximum strength among all possible combinations of those proteins. Finally, in the adjacency matrix, the cell having the labels of the chosen proteins is filled with “1”. When the procedure ends due to no protein to bind the empty cells of the adjacency matrix are filled with “0”s. In the end, the adjacency matrix becomes ready to be compared with the true precision matrix.

5.5. Outputs of the Algorithms

Analyses under the Normality Assumption

In order to evaluate and compare the performances of both algorithms in terms of their accuracies, multivariate normally distributed data are generated under the dimensions 10, 20, 50 and 100 proteins consisting of 20 observations per protein under scale-free structure. Table 5.2 indicates the outputs of the analyses from the simulated data.

Table 5.2.: GGM and RFA results based on 1000 Monte Carlo runs under multivariate normally distributed data; and 10, 20, 50 and 100 number of nodes in the system, respectively.

GGM and RFA Analyses under Normality Assumption									
	GGM					**RFA**			
	Perfection Level	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**
Precision	1	0.5207	1.0000	0.5000	0.2414	0.5832	0.5715	0.5936	0.6313
Recall	1	0.3926	0.3448	0.0017	0.0003	0.3741	0.3573	0.3429	0.3389
F-Measure	1	0.4476	0.5128	0.0034	0.0007	0.4558	0.4397	0.4347	0.4410
FPR	0	0.1406	0.0000	0.0001	0.0000	0.1039	0.0454	0.0148	0.0061
FDR	0	0.4793	0.0000	0.5000	0.7586	0.4168	0.4285	0.4064	0.3688
MCC	1	0.2773	0.5571	0.0265	0.0079	0.3160	0.3824	0.4261	0.4510

Considering the outputs of both algorithms under the multivariate normality, even though the results are highly close to each other in all accuracy measures, it can be observed that RFA provides relatively higher accuracy than GGM. Specifically, the precision outputs demonstrate that, especially in 100 dimensions, RFA provides a much better result than GGM. The true positive rate, demonstrated via the recall value, exhibits a remarkable difference between GGM and RFA, especially, in 50 and 100 dimensions. From the perspective of the false discovery rate (FDR), it is obvious that RFA has better results, specifically, in 100 dimensions. MCC results show that, in the dimensions 50 and 100, RFA provides much better results than GGM.

The basis of GGM is the Normal distribution; therefore, it is expected for GGM to perform well on multivariate normal data. However, even under normality, the nonparametric approach RFA provides much better results, in particular, the dimension of the networks increases.

Analyses under the Non-normality

Biological data mostly are not normally distributed. Mistaken normality assumption may cause problems in inference of biological networks. Therefore, the choice of the algorithm that is the most suited one to the data has a significant importance in the model construction. In order to test how both algorithms perform when the normality assumption is not satisfied, data are generated from exponential, student-t and log-normal marginal distributions under dimensions 10, 20, 50 and 100 proteins by using the Gaussian copula. Tables 5.3 and 5.4 demonstrate the results of analyses.

Table 5.3.: GGM and RFA results based on 1000 Monte Carlo runs under non-normally distributed data whose margins are fully student-t with degrees of freedom 5; and whose number of nodes are 10, 20, 50 and 100. NC refers to the not-computable value.

Margins are Student-t									
	GGM					**RFA**			
	Perfection Level	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**
Precision	1	0.0998	0.0000	0.0000	0.0000	0.0069	0.0166	0.0000	0.0000
Recall	1	0.0618	0.0000	0.0000	0.0000	0.0061	0.0154	0.0000	0.0000
F-Measure	1	0.0763	NC	0.0000	NC	0.0065	0.0160	NC	NC
FPR	0	0.1223	0.0552	0.0208	0.0102	0.0359	0.0955	0.0216	0.0104
FDR	0	0.9002	1.0000	1.0000	1.0000	0.9931	0.9834	1.0000	1.0000
MCC	1	-0.0739	-0.0743	-0.0288	-0.0143	-0.0316	-0.0830	-0.0294	-0.0144

From Table 5.3., it is seen that when the margins come from the student-t distribution with the degree of freedom 5, the precision and the recall values are slightly higher in GGM than RFA under 10-dimensional systems. While under 20-dimensional systems, RFA provides slightly higher results in terms of the same accuracy measures. On the other hand, in 50 and 100-dimensional systems, both GGM and RFA provide exactly the same outputs.

Under fully exponential marginal distributions with rate 5 as presented in Table 5.4., the results of RFA and GGM are very close and there is no any particular advantage in modeling the system either RFA or GGM.

Table 5.4.: GGM and RFA results based on 1000 Monte Carlo runs under non-normally distributed data whose margins are fully exponential with rate 5; and whose number of nodes are 10, 20, 50 and 100. NC refers to the not-computable value.

Margins are Exponential									
	GGM					**RFA**			
	Perfection Level	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**
Precision	1	0.0000	0.0000	0.0000	0.0000	0.0183	0.0127	0.0016	0.0041
Recall	1	0.0000	0.0000	0.0000	0.0000	0.0151	0.0115	0.0009	0.0036
F-Measure	1	NC	NC	NC	NC	0.0166	0.0121	0.0011	0.0038
FPR	0	0.1220	0.0552	0.0208	0.0102	0.1778	0.0938	0.0216	0.0174
FDR	0	1.0000	1.0000	1.0000	1.0000	0.9817	0.9873	0.9984	0.9959
MCC	1	-0.1562	-0.0743	-0.0289	-0.0143	-0.1757	-0.0860	-0.0282	-0.0149

On the other side, as seen in Table 5.5., under the log-normal margins' data, the performance of RFA is better than or at least equal to GGM based on all measures except the false positive rate (FPR) which exhibits better results in GGM than RFA. Furthermore, under this condition, we observe that GGM cannot calculate most of the scores due to its zero true positive estimates.

Finally, in the assessment of the mixture distributions whose margins come from exponential and log-normal distributions together with normal densities as presented in Table 5.6. and Table 5.7., it is found that RFA significantly performs better than GGM in all measures except false positive rate and, similar to previous analyses, GGM cannot be computed in most of these scores even under lower dimensional systems.

Table 5.5.: GGM and RFA results based on 1000 Monte Carlo runs under non-normally distributed data whose margins are fully log-normal with mean 10 and standard deviation 2; and whose number of nodes are 10, 20, 50 and 100. NC refers to the not-computable value.

Margins are Log-normal									
	GGM					**RFA**			
	Perfection Level	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**
Precision	1	NC	NC	NC	NC	0.0000	0.0000	0.0000	0.0000
Recall	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
F-Measure	1	NC	NC	NC	NC	NC	NC	NC	NC
FPR	0	0.0000	0.0000	0.0000	0.0000	0.1463	0.0939	0.0275	0.0120
FDR	0	NC	NC	NC	NC	1.0000	1.0000	1.0000	1.0000
MCC	1	NC	NC	NC	NC	-0.1730	-0.0988	-0.0333	-0.0155

Table 5.6.: GGM and RFA results based on 1000 Monte Carlo runs under non-normally distributed data whose margins are half log-normal with mean 10 and standard deviation 2 and, half normal with mean 10 and standard deviation 2; and whose number of nodes are 10, 20, 50 and 100. NC refers to the not-computable value.

Margins are Half Normal and Half Log-normal									
	GGM					**RFA**			
	Perfection Level	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**
Precision	1	NC	NC	NC	NC	0.2000	0.0385	0.0000	0.0000
Recall	1	0.0000	0.0000	0.0000	0.0000	0.2222	0.0526	0.0000	0.0000
F-Measure	1	NC	NC	NC	NC	0.2105	0.0444	NC	NC
FPR	0	0.0000	0.0000	0.0000	0.0000	0.1951	0.1381	0.0550	0.0143
FDR	0	NC	NC	NC	NC	0.8000	0.9615	1.0000	1.0000
MCC	1	NC	NC	NC	NC	0.0260	-0.0745	-0.0477	-0.0169

Table 5.7.: GGM and RFA results based on 1000 Monte Carlo runs under non-normally distributed data whose margins are half exponential with rate 5, and half normal with mean 10 and standard deviation 2; and whose number of nodes are 10, 20, 50 and 100. NC refers to the not-computable value.

Margins are Half Normal and Half Exponential									
	GGM					**RFA**			
	Perfection Level	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**	**10 Proteins**	**20 Proteins**	**50 Proteins**	**100 Proteins**
Precision	1	NC	NC	NC	NC	0.1000	0.1053	0.0000	0.0248
Recall	1	0.0000	0.0000	0.0000	0.0000	0.1111	0.1053	0.0000	0.0303
F-Measure	1	NC	NC	NC	NC	0.1053	0.1053	NC	0.0272
FPR	0	0.0000	0.0000	0.0000	0.0000	0.2195	0.0939	0.0375	0.0241
FDR	0	NC	NC	NC	NC	0.9000	0.8947	1.0000	0.9752
MCC	1	NC	NC	NC	NC	-0.1041	0.0113	-0.0390	0.0056

5.6. Description of Real Data

Even though the simulated data provide highly close representations of the real biological situations, it is still necessary to test the algorithms on the real observations to assess the real life problems.

In this study, we use two real datasets. The first one belongs to JAK-STAT pathway controlling the mammalian immune system and consisting of 38 proteins (Ayyıldız, 2013). On the other side, the second one is the cell signaling data consisting of 11 proteins (Sachs et al., 2005).

JAK-STAT Pathway

JAK-STAT is a ligand-specific pathway that controls a biological process, gene expression, by enabling transcriptional regulation without a need for second messengers. In this pathway, information provided by the extracellular polypeptide signals is transferred directly to the target gene promoters in the nucleus by the intervention of transmembrane receptors (Aaronson & Horvath, 2002).

In fact, due to the lack of real biological data, the data generated by the Gillespie algorithm is used in this study. The Gillespie algorithm (Gillespie, 1977) is one of the most common and practical exact stochastic simulation algorithms for the simulation of biological networks. In the simulation, all the initial numbers of molecules are set to 100 for all 38 proteins and the reaction rate constants are equal to the values given in the study of Maiwald et al., 2010, which are biologically validated. The list of totally 38 proteins is presented in the Table 5.8 and the simple representation of the true JAK-STAT system is shown in Figure 5.2.

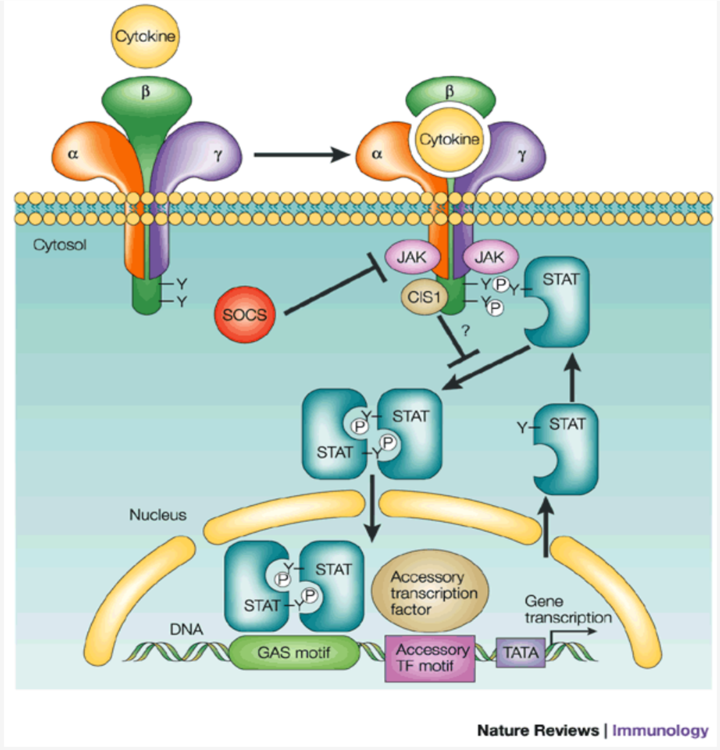

Figure 5.1.: Simple representation of the JAK-STAT pathway from Nature Reviews (Leonard, 2001).

Outputs of both algorithms of the JAK-STAT pathway can be seen in Table 5.9.

Considering these results and excluding the F-Measure values from the evaluation measures because of the undefined number obtained from the GGM analysis, it can be clearly observed that RFA provides better results than GGM with higher true positive rate denoted via the recall value.

Table 5.8.: List of proteins in the JAK-STAT pathway whose description is based on the study of (Maiwald et al., 2010).

Molecule Symbol	**Molecule Name**	**Molecule Symbol**	**Molecule Name**
P1	Receptor IFNAR1	P20	IRF9n
P2	TYK	P21	Free TFBS
P3	Receptor Tyk Complex	P22	Occupied TFBS
P4	Receptor IFNAR2	P23	mRNAn
P5	JAK	P24	mRNAc
P6	Receptor Jak Complex	P25	SOCS
P7	IFN_Free	P26	Stat2n_IRF9
P8	IFNAR Dimer	P27	STAT2n
P9	Active Receptor Complex_Stat2c	P28	CP
P10	STAT2c_IRF9	P29	ISGF-3c_CP
P11	Active Receptor Complex_STAT2c	P30	Stat1c*_Stat2c*_CP
P12	IRF9c	P31	NP
P13	STAT2c	P32	Stat1n*_Stat2n*_NP
P14	STAT1c	P33	ISGF-3n_NP
P15	Active Receptor Complex_STAT2c_STAT1c	P34	Occupied TFBS_NP
P16	STAT1c*_STAT2c*	P35	PIAS
P17	ISGF-3c	P36	PIAS_ISGF-3n
P18	ISGF-3n	P37	STAT1n
P19	STAT1n*_STAT2n*	P38	IFN_influx

Table 5.9.: GGM and RFA results on the JAK-STAT pathway.

JAK-STAT Pathway			
	Perfection Level	**GGM**	**RFA**
Precision	1	NC	0.1074
Recall	1	0.0000	0.0611
F-Measure	1	NC	0.0779
FPR	0	0.0000	0.0502
FDR	0	NC	0.8926
MCC	1	NC	0.0141

Cell Signaling Pathway

The cell signaling pathway consists of 11 proteins to describe the phosphorylation of the molecules listed in Table 5.10 under various experimental conditions of the human primary naive CD4T2 cells measured from 11672 red blood cells. The visual representation of this true cell signaling system can be seen in Figure 5.2.

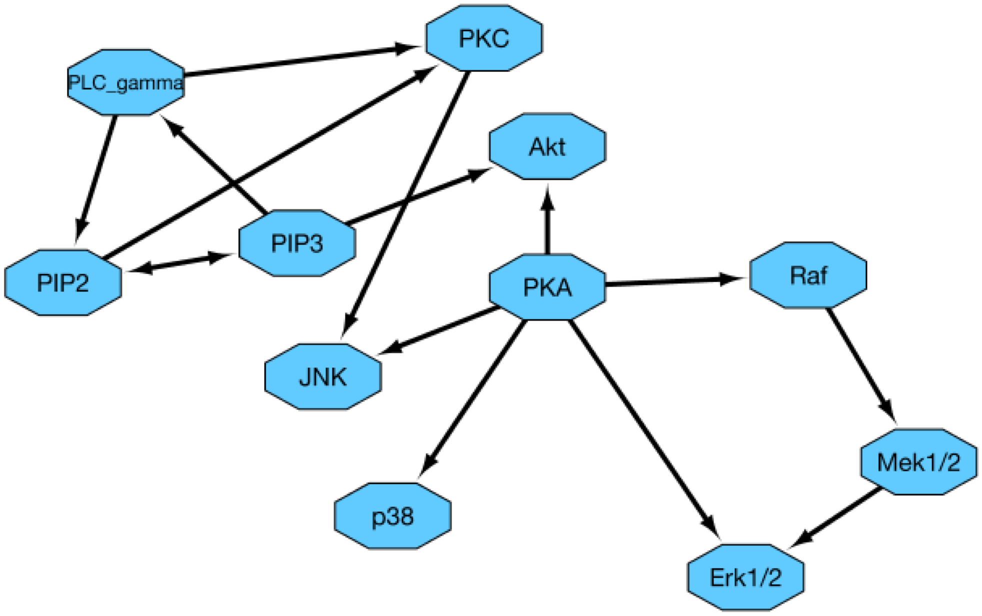

Figure 5.2.: Simple illustration of the true cell signaling pathway.

In Figure 5.2, the molecules are represented by octagons, and the interactions among these molecules are denoted by arrows, where the size and the shape of the nodes are drawn as the same for each molecule, not indicating any difference in terms of the biological role of the molecules.

Table 5.10. List of species in the cell signaling pathway whose description is based on the study of Sachs et al., 2005.

Measured Molecule	**Antibody Specificity**
Raf	Phosphorylation at S259
Erk1 and Erk2	Phosphorylation at T202 and Y204
P38	Phosphorylation at T180 and Y182
jnk	Phosphorylation at T183 and Y182
AKT	Phosphorylation at S473
Mek1 and Mek2	Phosphorylation at S217 and S221
PKA substrates	Detects proteins and peptides containing a phospho-Ser/Thr residue with arginine at the -3 position
PKC	Detects phosphorylated PKC-α, -βI, -βII, -δ, -ϵ, -η and -θ isoforms only at C-terminal residue homologous to S660 of PKC- βII
PLC-γ	Phosphorylation at Y783
PIP$_2$	Detects PIP$_2$
PIP$_3$	Detects PIP$_3$

Table 5.11.: Cell Signaling results from both GGM and RFA.

Cell Signaling Pathway			
	Perfection Level	**GGM**	**RFA**
Precision	1	NC	1.0000
Recall	1	0.0000	0.0444
F-Measure	1	NC	0.0851
FPR	0	0.0000	0.0000
FDR	0	NC	0.0000
MCC	1	NC	0.1685

Table 5.11 indicates the results of GGM and RFA in this cell signaling pathway. From the outputs, it is seen that RFA finds more accurate measures while GGM cannot calculate most of the measures.

Human Gene Expression Pathway

The human gene expression pathway consists of 100 proteins in which each molecule constitutes 60 observations. Different from the Cell Signaling pathway, here, true precision of the gene expression pathway is unknown, causing us not to compute the accuracy measures for the comparison of GGM and RFA outputs. Therefore, for this human gene expression dataset, we run both GGM and RFA by excluding the calculation of the accuracy measures based on TP, TN, FP and FN, and instead, we only record the interactions that are found in the resulting precision matrices.

The results exhibit that GGM poorly performs in this dataset by not catching even one interaction between molecules, whereas, RFA can detect the new interactions as well as capturing validated interactions validated from the databases STRING and GeneMANIA.

Table 5.12 illustrates the interactions between proteins that are detected via RFA; and among these interactions, the interactions between HMOX1 and IL8, RPS4Y1 and EIF1AY, and between DDX3Y and KDM5D are validated based on the String Database.

Table 5.12.: The list of interactions between molecules that are recorded as final results of RFA from the human gene expression data.

Interactions	**Pair of Molecule Names**	
	Molecule 1	**Molecule 2**
Biologically Validated Interactions	HMOX1	IL8**
	RPS4Y1	EIF1AY**
	DDX3Y	KDM5D**
	TNFRSF19	LEPREL1*
New Interactions		
Close Localization based on STRING DB	EPS8	STEAP1
	G0S2	IL8
	ABCC6	KDM5D
	MOXD1	LEPREL1
Others based on STRING DB	EPS8	RGS13
	TCEAL2	F13A1
	STEAP1	HLA-A

** Validated interactions based on both STRING-DB and GeneMANIA-DB *Validated interactions based on GeneMANIA-DB

In order to exhibit the true situation of PPIs among selected proteins via RFA, we convert the probe IDs into protein IDs. Figure 5.3 displays the true situation of the proteins and their true interactions based on String Database. Considering the localization of the proteins in Figure 5.3, especially EPS8 and STEAP1, and also G0S2 and IL8, ABCC6 and KDM5D, and MOXD1 and LEPREL1 seem close to each other, which can be considered as possible new biological pathways.

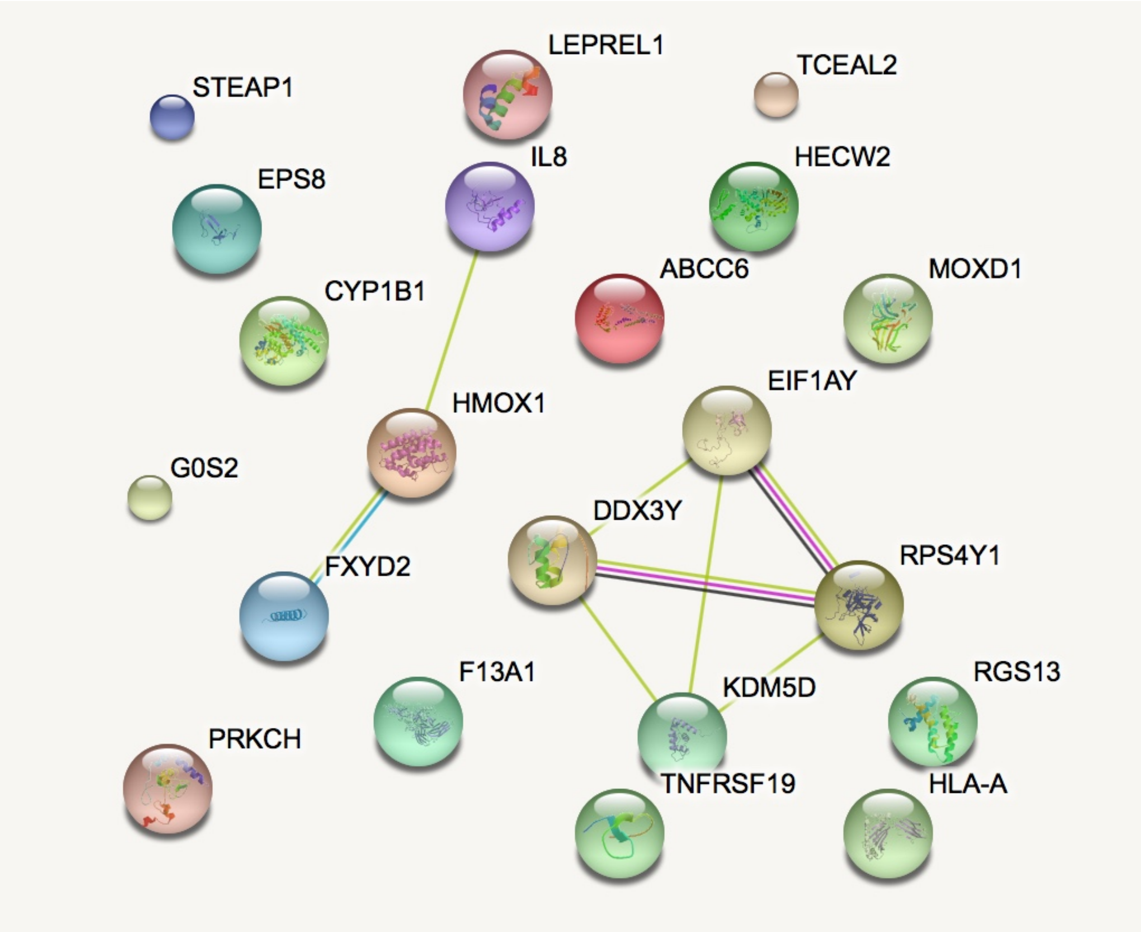

Figure 5.3. The true representation of the selected proteins and their interactions based on the STRING Database at the end of the RFA analysis. The smaller nodes correspond to the proteins whose 3D structures are known, while the 3D structures of the bigger nodes are known.

Afterwards, we display the true gene interactions among these captured nodes via RFA from the GeneMANIA database. Figure 5.4 exhibits the true gene interactions with co-expression and Figure 5.5 illustrates the true gene interactions with co-localization and genetic interactions, in which blue edges refer to the co-localization of the genes, while green edges among genes indicate the genetic interactions.

Similar to the STRING DB results, some of the nodes are observed close to each other including EPS8 and STEAP1, ABCC6 and KDM5D, which were highly close in

STRING-DB results, STEAP1 and HLA-A, and DDX3Y and KDM5D based on GeneMANIA.

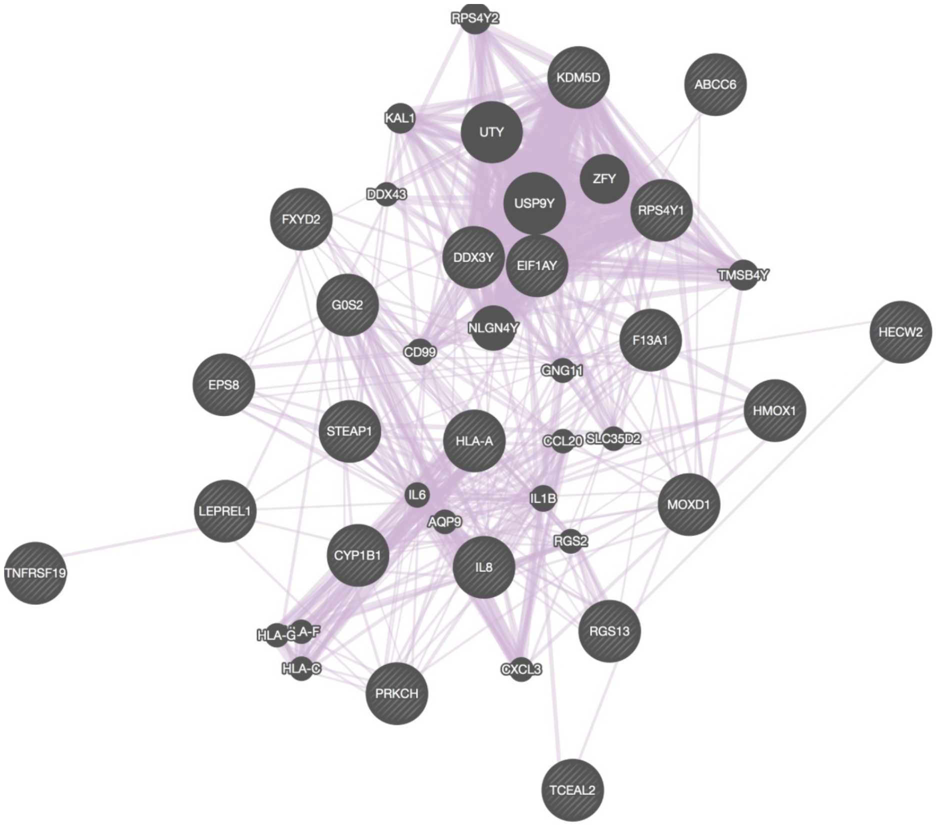

Figure 5.4.: The true gene interactions with co-expression of the selected genes based on GeneMANIA Database. The gray nodes with white lines in refer to the selected genes via RFA, while the fully gray nodes without white lines in correspond to the added genes by the GeneMANIA database.

Considering these gene interactions with co-expression based on the GeneMANIA Database represented in Figure 5.4, it can be clearly said that RFA is strong enough to capture all these interactions from the data with no false positives in terms of co-expressions.

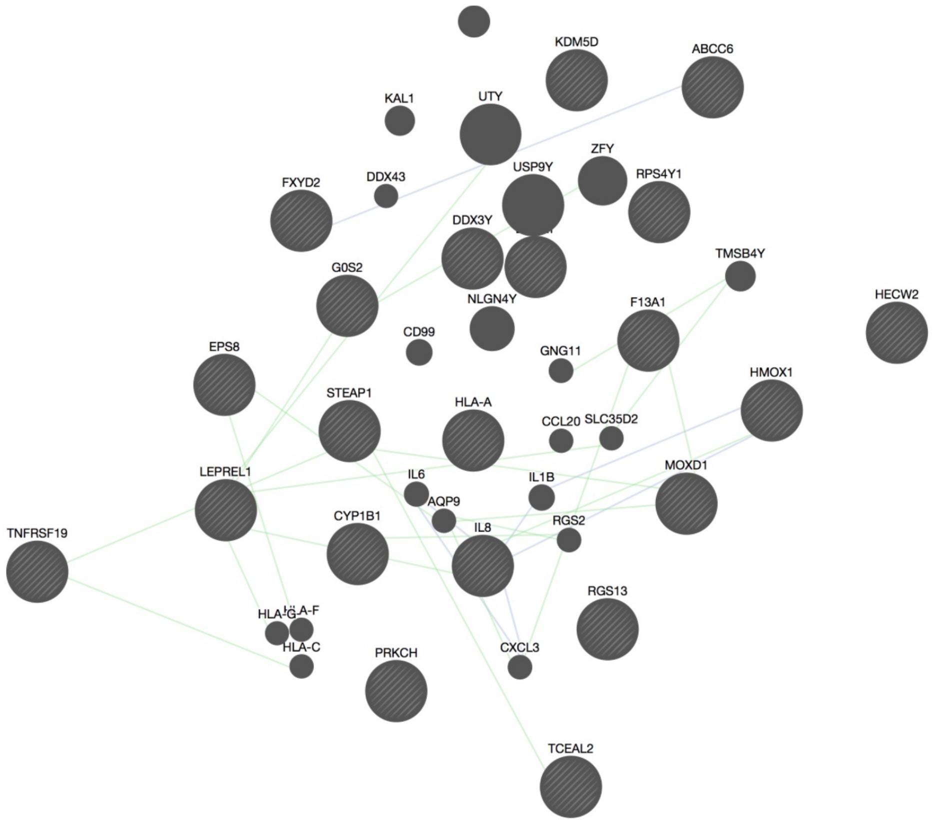

Figure 5.5.: The true gene interactions with co-expression of the selected genes based on GeneMANIA Database. The gray nodes without white lines in refer to the selected genes via RFA, while the fully gray nodes without white lines in correspond to the added genes by the GeneMANIA database.

Considering the true network based on the GeneMANIA Database in terms of co-localization and genetic interactions of the detected genes via RFA, it can be seen that a green edge, a genetic interaction, is validated between TNFRSF19 and LEPREL1, which is signed with “*” in Table 5.12.

CHAPTER 6

OUTLOOK, DISCUSSION AND CONCLUSION

Outlook

The first part of this thesis gives a brief introduction about the importance of biological network inference studies and how it becomes one of the most popular and promising research area in the bioinformatics.

The second part consists of two independent subsections; one is the basic explanation of network, types of biological networks, and different approaches to infer the networks from the biological data. Among the alternatives, the differential equation is performed as the random forest algorithm. The second subsection is the CART methodology, which takes place into this thesis since it constitutes the basis of the random forest algorithm.

In the third part, the detailed explanations of both algorithms are presented with their mathematical details. First, GGM is explained by providing a couple of approaches to infer GGM from the model. Among alternatives, the lasso based graphical approach with the penalized likelihood is chosen since it is the most suitable one due to its tendency to construct sparse models. Second, RFA is examined in terms of its basis, advantages and accuracy.

In the fourth chapter, the basics of copulas and the description of different copula types are presented by implying their advantages and disadvantages and finally, the fifth chapter indicates the application steps and results. Here, the accuracy measures are illustrated via tabulated results of both GGM and RFA.

Discussion

In this study, it is aimed to infer the biological systems by applying both parametric and non-parametric algorithms and to evaluate the performances of the algorithms in terms of their accuracies in order to examine the biological network construction and the inference process from a different perspective.

The Gaussian graphical model is one of the commonly used approaches due to its popular basis, Gaussian distribution. However, biological data most of the time are not normally distributed, causing problems during the network construction. Since the correct and accurate inference of the biological systems carries a huge importance in exploring disease pathways and many other biological incidents, the choice of the algorithm is responsible for the flow of this interdisciplinary procedure.

Since the basis of GGM is the Gaussian distribution, it is expected for GGM to perform most powerful and provide remarkably better results than any other non-parametric algorithm on the multivariate normally distributed data. However, the results exhibit that the suggested non-parametric algorithm, random forest, provides much better results than GGM in terms of all accuracy measures. On the other hand, with respect to its basic distribution, GGM is also expected to exhibit lower performance on the non-normally distributed data than RFA. Considering the results obtained from the analyses of both algorithms on non-normally distributed data whose marginal distributions are exponential, student-t and log-normal, which are generated by copula, GGM and RFA provide similar results, but still we observe improvement in the estimates of RFA regarding the output of GGM. One reason of this situation may arise from the use of the Gaussian copula. Even though the margins are chosen from exponential, student-t and log-normal distributions, still the basis of the Gaussian copula may cause some normality signs; therefore, it may provide an equal condition for both algorithms.

In the real data simulations, RFA again provides much better results than GGM in all applicable accuracy measures. Additionally, since RFA is a nonparametric approach, it does not require the definition of any external value, such as thresholds used for rendering the precision matrices binary or penalty values directly controlling the structure of the network, i.e., sparsity level, can be regarded as one of the important issues in the model inference.

Considering all these outputs and the advantages, it is obvious that RFA provides at least the same results, but mostly, it provides better accuracy measures than GGM. Therefore, it can be confidentially suggested as a strong and promising alternative of GGM.

Conclusion and Future Work

Modeling of protein-protein interactions is one of the most important issues in understanding disease pathways, thereby, personalized medicine. Constructing these interaction networks accurately enables biologists and physicians to make the best decision for their research or their patients. In this study, we have demonstrated that nonparametric models are easy and highly strong alternatives for inferring biological networks. Both the generated data and the real pathway have exhibited that without requiring external values, interactions can be modeled accurately by the nonparametric models.

For the future studies, in order to overcome the normality assumption in the definition of the data used to evaluate the performance of both algorithms under the non-normality assumption, instead of using the Gaussian copula, the description may be extended by D-vine (Schirmacher & Schirmacher, 2008) copula. Additionally, some other nonparametric models, such as CART (Timofeev, 2004) and Hidden markov model

(Yoon, 2009) may be applied as alternatives to the Gaussian graphical models in an attempt to infer the biological systems. Finally, the performance of GGM can be improved by suggesting another model selection criteria in place of STARS and RIC, such as AIC (Akaike, 1973) and ICOMP for the selection of the optimal penalty value λ.

REFERENCES

Aaronson, D., & Horvath, C. (2002). Roadmap for those who dont know JAK-STAT. *Science*, *296*, 1653–5. http://doi.org/10.1126/science.1071545.

Akaike, H. (1973). Information theory and an extension of the maximum likehood principle. In *Second International Symposium on Information Theory* (pp. 267–281). http://doi.org/10.2307/2938260.

Ayyıldız, E. (2013). *Gaussian graphical models in estimation of biological systems*. Master's Thesis, Middle East Technical University.

Barabási, A.-L., Gulbahce, N., & Loscalzo, J. (2011). Network medicine: a network-based approach to human disease. *Nature Reviews. Genetics*, *12*(1), 56–68. http://doi.org/10.1038/nrg2918.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. http://doi.org/10.1023/A:1010933404324.

Breiman, L., & Cutler, A. (2012). Breiman and Cutler’s random forests for classification and regression. *Package “randomForest,”* 29. http://doi.org/10.5244/C.22.54

Castelo, R., & Roverato, A. (2009). Reverse engineering molecular regulatory networks from microarray data with qp-graphs. *J Comput Biol*, *16*(2), 213–227. http://doi.org/10.1089/cmb.2008.08TT.

de la Fuente, A., Bing, N., Hoeschele, I., & Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, *20*(18), 3565–3574. http://doi.org/10.1093/bioinformatics/bth445\rbth445.

Donoho, D. L., & Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, *81*(3), 425–455. http://doi.org/10.1093/biomet/81.3.425.

Foster, D. P., & George, E. I. (1994). The risk inflation criterion for multiple regression. *The Annals of Statistics*, *22*(4), 1947–1975. http://doi.org/10.1214/aos/1176325766.

Friedman, A. J., Hastie, T., & Tibshirani, R. (2014). glasso: graphical lasso-estimation of Gaussian graphical models (R package).

Friedman, J., Hastie, T., Höfling, H., & Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, *1*(2), 302–332. http://doi.org/10.1214/07-AOAS131.

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics (Oxford, England)*, *9*(3), 432–41.

http://doi.org/10.1093/biostatistics/kxm045.

Friedman, J., Hastie, T., & Tibshirani, R. (2015). Package "glasso" title graphical lasso-estimation of Gaussian graphical models. Retrieved from http://www-stat.stanford.edu/~tibs/glasso.

Genest, C. & Favre, A. C. (2007). Everything you always wanted to know about copula but were afraid to ask. *Journal of Hydrologic Engineering, 12(4),* 347-368.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, *81*(25), 2340–2361. http://doi.org/10.1021/j100540a008.

Kass, R. E., & Wasserman, L. (1995). A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the American Statistical Association*. http://doi.org/10.2307/2291327.

Leonard, W. J. (2001). Cytokines and immunodeficiency diseases. *Nature Reviews. Immunology*, *1*(3), 200–208. http://doi.org/10.1038/35105066.

Lewis, R. J., Ph, D., & Street, W. C. (2000). An introduction to classification and regression tree (CART) analysis. *2000 Annual Meeting of the Society for Academic Emergency Medicine*, (310), 14p. http://doi.org/10.1.1.95.4103.

Liu, H., Roeder, K., & Wasserman, L. (2010). Stability approach to regularization selection (StARS) for high dimensional graphical models. *Advances in Neural Information Processing Systems*, *24*(2), 1432–1440. http://doi.org/papers3://publication/uuid/F1CE0C72-5199-4FC6-829C-B76A36C5ED28.

Magwene, P. M., & Kim, J. (2004). Estimating genomic coexpression networks using first-order conditional independence. *Genome Biology*, *5*(12), R100. http://doi.org/10.1186/gb-2004-5-12-r100.

Maiwald, T., Schneider, A., Busch, H., Sahle, S., Gretz, N., Weiss, T. S., … Klingmüller, U. (2010). Combining theoretical analysis and experimental data generation reveals IRF9 as a crucial factor for accelerating interferon-α induced early antiviral signalling. *FEBS Journal*, *277*(22), 4741–4754. http://doi.org/10.1111/j.1742-4658.2010.07880.x.

Meinshausen, N., Bühlmann, P., & Zürich, E. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, *34*(3), 1436–1462. http://doi.org/10.1214/009053606000000281.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., & Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*

(New York, N.Y.), *308*(5721), 523–9. http://doi.org/10.1126/science.1105809.

Schirmacher, D., & Schirmacher, E. (2008). Multivariate dependence modeling using pair-copulas. In *2008 ERM Symposium* (pp. 1–52). Retrieved from http://www.ermsymposium.org/2008/pdf/papers/Schirmacher.pdf%5Cnhttp://www.ermsymposium.org/2008/index.php.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464. http://doi.org/10.1214/aos/1176344136.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, *58*(1), 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, *67*(1), 91–108. http://doi.org/10.1111/j.1467-9868.2005.00490.x.

Timofeev, R. (2004). Classification and Regression Trees (CART) Theory and Applications. *Journal of neurosurgery*. http://doi.org/10.3171/jns.1995.82.5.0764.

Trivedi, P. K., & Zimmer, D. M. (2007). Copula modeling: An introduction for practitioners. *Foundations and Trends® in Econometrics*, *1*(1), 1–111. http://doi.org/10.1561/0800000005.

van Ophem, H. (1999). A general method to estimate correlated discrete random variables. *Econometric Theory*, *15*(2), 228–237. http://doi.org/10.1017/S0266466699152058.

Veiga, P., Bulbarela-Sampieri, C., Furlan, S., Maisons, A., Chapot-Chartier, M. P., Erkelenz, M., Kulakauskas, S. (2007). SpxB regulates *O*-acetylation-dependent resistance of *Lactococcus lactis* peptidoglycan to hydrolysis. *The Journal of Biological Chemistry*, *282*(27), 19342–19354. http://doi.org/10.1074/jbc.M611308200.

Wawrzyniak, M. M. (2006). *Dependence concepts*. Delft Institute of Applied Mathematics, Delft University of Technology.

Whittaker, J. (2001). *Graphical models in applied multivariate statistics*. John Wiley and Sons. Retrieved from http://dl.acm.org/citation.cfm?id=1593402.

Wille, A., & Bühlmann, P. (2006). Low-order conditional independence graphs for inferring genetic networks. *Statistical Applications in Genetics and Molecular Biology*, *5*(1). http://doi.org/10.1007/11671404_1.

Wille, A., Zimmermann, P., Vranova, E., Furholz, A., Laule, O., Bleuler, S., …

Bühlmann, P. (2004). Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana. *Genome Biology*, *5*(11). http://doi.org/10.1186/gb-2004-5-11-r92.

Wit, E., Vinciotti, V., Purutçuoğlu, V. (2010). Statistics for biological networks: short course notes. Florianopolis, Brazil: 25th International Biometric Conference (IBC).

Witten, D. M., & Tibshirani, R. (2009). Covariance-regularized regression and classification for high-dimensional problems.

Yoon, B. J. (2009). Hidden Markov models and their applications in biological sequence analysis. *Current Genomics*, *10*, 402–415. http://doi.org/10.2174/138920209789177575.

Yuan, M., & Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, *94*(1), 19–35. http://doi.org/10.1093/biomet/asm018.

Zou, H. (2006). The Adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, *101*(476), 1418–1429. http://doi.org/10.1198/016214506000000735.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Statist. Soc. B*, *67*(2), 301–320.