GRAPHICAL MODELS IN INFERENCE OF BIOLOGICAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

HAJAR FARNOUDKIA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
STATISTICS

JULY 2020

Approval of the thesis:

GRAPHICAL MODELS IN INFERENCE OF BIOLOGICAL NETWORKS

submitted by **Hajar Farnoudkia** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Statistics Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ayşen Dener Akkaya
Head of Department, **Statistics**

Prof. Dr. Vilda Purutçuoğlu
Supervisor, **Department of Statistics, METU**

Examining Committee Members:

Assoc. Prof. Dr. Babek Erdebilli
Industrial Engineering Department, Ankara YBU

Prof. Dr. Vilda Purutçuoğlu
Department of Statistics, METU

Prof. Dr. Barış Sürücü
Department of Statistics, METU

Prof. Dr. Birdal Şenoğlu
Department of Statistics, Ankara University

Prof. Dr. Ömür Uğur
Department of Mathematics, METU

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 work.

Name, Last Name: HAJAR FARNOUDKIA

Signature :

ABSTRACT

GRAPHICAL MODELS IN INFERENCE OF BIOLOGICAL NETWORKS

Farnoudkia, Hajar
Ph.D., Department of Statistics
Supervisor : Prof. Dr. Vilda Purutçuoğlu

July 2020, 70 pages

In recent years, particularly, on the studies about the complex system's diseases, better understanding the biological systems and observing how the system's behaviors, which are affected by the treatment or similar conditions, accelerate with the help of the explanation of these systems via the mathematical modeling. Gaussian Graphical Models (GGM) is a model that describes the relationship between the system's elements via the regression and represents the states of the system via the multivariate Gaussian (normal) distribution. This distribution also explains the structure of biological systems by means of its "conditional independence" feature. Therefore, in the inverse of the covariance matrix of the multivariate normal distribution, the "zero" value implies no functional interaction, and the "non-zero" value stands for the interaction between the proteins in the estimate of the system's structure. In this study, as the novelty, we use the Copula Gaussian Graphical Models (CGGM) in modeling the steady-state activation of the biological networks and make the inference of the model parameters under the Bayesian setting. We suggest the reversible jump Markov chain Monte Carlo (RJMCMC) algorithm to estimate the plausible interactions (conditional dependence) between the systems' elements which are proteins or genes. Several data sets are used to illustrate the out-performance of the proposed RJMCMC in comparison with most of its alternatives. Also, we used some semi-Bayesian RJMCMC method to estimate the autoregressive coefficient matrix where GGM repeated through time. We improved the model by full-Bayesian approach and followingly, by a tuning parameter to increase the accuracy of the estimated matrices. Some simulated data sets are used to show the accuracy of the different proposed methods. Finally, we suggested a method to discover the relationships between variables through copula which is more flexible and it is more appropriate for the non-symmetric or tail dependent cases. We applied the suggested ways in four real data set and we saw that copula can discover the joint density structure in addition to the available relationships in terms of the shape of the joint distribution to see whether it is symmetric or non-symmetric or even tail dependent or not.

Keywords: Gaussian Graphical Models, Reversible jump Markov Chain Monte Carlo Methods, Time series, Copula

ÖZ

BİYOLOJİK AĞLARIN ÇIKARIMINDA GRAFİK MODELLER KULLANIMI

Farnoudkia, Hajar
Doktora, İstatistik Bölümü
Tez Yöneticisi : Prof. Dr. Vilda Purutçuoğlu

Temmuz 2020 , 70 sayfa

Son yıllarda, özellikle, karmaşık sistemin hastalıkları ile ilgili çalışmalar, biyolojik sistemleri daha iyi anlamak ve tedavi veya benzer koşullardan etkilenen sistem davranışları gözlemlemek bu sistemlerin matematiksel modellerle açıklaması sayesinde hızlanmıştır. Gauss Grafiksel Modeller (GGM), regresyon yoluyla sistemin elemanları arasındaki ilişkiyi tanımlayan ve çok değişkenli Gauss (normal) dağılım yoluyla sistemin durumlarını temsil eden bir modeldir. Bu dağılım biyolojik sistemlerin yapısını "koşullu bağımsızlık" özelliği ile de açıklamaktadır. Bu nedenle, çok değişkenli normal dağılımın kovaryans matrisinin tersinde, "sıfır" değeri hiçbir işlevsel etkileşim anlamına gelmez ve "sıfır olmayan" değer, sistemin yapısı tahmininde proteinler arasındaki etkileşimi ifade eder. Bu çalışmada, yenilik olarak, biyolojik ağların kararlı durum aktivasyonunu modellemek için Copula Gaussian Grafiksel Modelleri (CGGM) kullanıyoruz ve Bayesian kurulumu altında model parametrelerinin çıkarımını yapıyoruz. Sistem elemanları olan proteinler veya genlerin arasındaki olası etkileşimi (koşullu bağımlılık) tahmin etmek için tersine atlamalı Markov zinciri Monte

Karlo (RJMCMC) algoritmasını önermekteyiz.. Önerilen RJMCMC'nin alternatiflerinin çoğuna kıyasla yüksek performansını göstermek için çeşitli veri setleri kullanmaktadır. Ayrıca, GGM'nin zaman içinde tekrarladığı otoregresif katsayı matrisini tahmin etmek için bazı yarı Bayes RJMCMC yöntemini kullandık. Modeli tam Bayesci yaklaşımla ve ardından tahmini matrislerin doğruluğunu artırmak için bir ayar parametresi ile geliştirdik. Bazı simüle edilmiş veri setleri, önerilen farklı yöntemlerin doğruluğunu göstermek için kullanıldı. Son olarak, daha genel veriler için özellikle simetrik olmayan veya kuyruğa bağlı durumlar için daha esnek olan kopula yoluyla değişkenler arasındaki ilişkileri keşfetmek için bir yöntem önerdik. Dört gerçek veri setinde önerilen yolları uyguladık ve kopula'nın simetrik veya simetrik olmayan veya hatta kuyruğa bağlı olup olmadığını görmek için birleşik (ortak) dağılımının şekli açısından, mevcut ilişkilere ek olarak birleşik dağılımının yapısını keşfedebileceğini gördük.

Anahtar Kelimeler: Gauss Grafik Modelleri, Tersine atlamalı Markov Zinciri Monte Carlo Yöntemleri, Zaman Serileri, Kopula

To my dearests: My mother and my son Fuad

ACKNOWLEDGMENTS

I would like to thank my supervisor Professor Vilda Purutçuoğlu for her constant support, guidance, and friendship. It was a great honor to work with her for the last five years and our cooperation influenced my academic and world view highly. When I entered the university as a foreign Ph.D. student, it was my first time to be abroad. Prof. Purutçuoğlu was the person who opened the doors for me by hiring me for her project. I never forget her statement about me that was "You are a woman who has a child may need a fortune to show yourself." when I told her that I am not familiar with any of the terms in the project title. I was far from my son for about three years, but I could go to my home country whenever I needed. I should thank TUBITAK grant (114E636) for their supports in the first two years of my educational period and also, I would like to thank Professors Barış Sürücü and Birdal Şenoğlu for their help and presence in every TIK presentation and showing me some ways to solve the problems I might face.

Finally very special thanks to a group of people who've taught me the real meaning of friendship, lately.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xvi

LIST OF ABBREVIATIONS xviii

CHAPTERS

1 INTRODUCTION 1

2 COPULA GAUSSIAN GRAPHICAL MODEL 5

2.1 Some fundamental definitions 5

2.1.1 Graphical Model 6

2.1.1.1 DAG 6

2.1.1.2 Undirected graphs 7

2.1.2 Gaussian Graphical Models 9

2.2 Copula Gaussian graphical models 11

2.3 Reversible Jump Markov Chain Monte Carlo method 14

2.3.1 Resample the latent data 15

2.3.2 Resample the precision matrix 16

2.3.2.1 Resample the graph 16

2.4 RJMCMC Alternatives 17

2.4.1 Birth-Death MCMC 17

2.4.2 Carlin-Chib algorithm 18

2.4.3 Gibbs Sampling 19

2.4.4 Quadratic Approximation for Sparse Inverse Covariance Estimation 21

2.5 Application 22

2.5.1 Ovarian Cancer data 23

2.5.2 The Rochdale data 23

2.5.3 The CellSignal Data 25

2.5.4 The geneExpression data 26

3 TIME SERIES CHAIN GRAPHICAL MODELS 31

3.1 Introduction 31

3.2 Time Series Chain Graphical Model 32

3.2.1 Semi-Bayesian RJMCMC for the Estimation of Parameter Matrices of TSCGM 34

3.2.2 Full Bayesian approach by Reversible Jump Markov Chain Monte Carlo Method in TSCGM 35

3.2.2.1 Fulyl Bayesian RJMCMC without a tuning parameter 35

3.2.2.2 Modified fully-Bayesian RJMCMC with an Adaptable Tuning Parameter 38

3.3 Application 39

3.3.1 Accuracy of Semi-Bayesian RJMCMC for a Simulated Dataset 39

3.3.2 Accuracy of Fully-Bayesian RJMCMC for a Simulated Dataset 40

3.3.3 Accuracy of Fully-Bayesian RJMCMC with Adaptive Tuning Parameter for a Simulated Dataset . . . 41

3.4 Conclusion 42

4 COPULA GRAPHICAL MODELS IN THE CONSTRUCTION OF BIOLOGICAL NETWORKS 45

4.1 Introduction 45

4.2 Copula 45

4.2.1 Types of Copula families 47

4.3 Vine Copula in Inference of Complex Data 51

4.4 Application 54

4.4.1 The "CellSignal" data 54

4.4.2 Data 2 55

4.4.3 The "Rochdale" data 56

4.4.4 23 gene 57

4.4.5 50 genes 58

4.5 Conclusions 60

5 RESULTS AND CONCLUSIONS 61

REFERENCES 63

CURRICULUM VITAE 67

LIST OF TABLES

TABLES

Table 2.1 Definition of the confusion matrix 22

Table 2.2 The Rochdale data 24

Table 2.3 The estimated graph for the Rochale data 24

Table 2.4 The comparison between accuracies of different methods for the Rachdale data 25

Table 2.5 The comparison between accuracy of different methods for the CellSignal dataset 25

Table 2.6 The comparison between accuracy of different methods for the CellSignal dataset 29

Table 3.1 Results of accuracy (Acc) and sensitivity (Sens) measures for the simulated dataset under different sample size (n), number of genes (p) and number of time points (T). 40

Table 3.2 Results of specificity for the fully-Bayesian method with a likelihood based method under different sample sizes (n), number of genes (p) and number of time points (T). 41

Table 3.3 Results of Matthew correlation coefficient (MCC) of the modified RJMCMC method with adaptable tuning parameter (MCC$_A$), without tuning parameter (MCC$_B$) and also the likelihood based method (MCC$_L$). . . 42

Table 4.1 The denotation and properties of the bivariate Elliptical families . . 48

Table 4.2 the notation and the properties of the bivariate archimedean families 50

Table 4.3 The most important functions' names and descriptions according to the VineCopula package in R [5]. 53

Table 4.4 The upper triangular of the estimated adjacency matrix of the CellSignal data with the copula families via numbers in the "VineCopula" package. 55

Table 4.5 The upper triangular of the estimated adjacency matrix of Data 2 with the copula families via numbers in the "VineCopula" package 56

Table 4.6 Four positive edges, the best appropriate pair copula and associated Kendall's τ for data with 23 genes. 58

Table 4.7 Two positive edges, the best appropriate pair copula and the associated Kendall's τ for data with 50 genes. 59

LIST OF FIGURES

FIGURES

Figure 2.1 A directed acyclic graphical model with five variables. 6

Figure 2.2 : A and B are marginally dependent (Top graph) and conditionally independent (Bottom graph). 7

Figure 2.3 A and B are marginally dependent (Left graph) and conditionally independent (Right graph). 7

Figure 2.4 A and B are marginally independent (Left graph) conditionally dependent (Right graph). 8

Figure 2.5 An example of an undirected graph with six variables. 8

Figure 2.6 A representation of a marginal independence that an undirected graph cannot describe. 8

Figure 2.7 A representation of a conditional independence that a directed graph cannot describe. 9

Figure 2.8 The true undirected network of the CellSignal data 26

Figure 2.9 Biologically validated links of the gene expression data according to the study of [4] 28

Figure 4.1 Examples of a four-dimensional C-Vine (left panel with order $\{2, 4, 1, 3\}$) and a D-vine structure (with order$\{2, 4, 3, 1\}$) in the right panel) 47

Figure 4.2 CDF and PDF of a bivariate student's t copula 49

Figure 4.3 PDF and the scatter plot of the simulated Gaussian copula for $\rho = 0.8$ (left) and $\rho = 0.2$ (right) 50

Figure 4.4 Gumbel, Clayton and Frank copulas, respectively, from left to right with parameters corresponding to Kendall's τ values of 0.5. 51

Figure 4.5 Samples from the Clayton copulas rotated with Kendall's $\tau = 0.5$. 51

Figure 4.6 Proposed data analysis and model building work by some functions in package **VineCopula** in **R** 52

Figure 4.7 The quasi tree structure of 23 genes obtained from the STRING database. 58

Figure 4.8 The quasi tree structure of 50 genes obtained from the STRING database 59

LIST OF ABBREVIATIONS

GGM	Gaussian Graphical Models
CGGM	Copula GGM
RJMCMC	Reversible Jump Markov Chain Monte Carlo Method
TSCGM	Time Series Chain Graphical models
DAG	Directed Acyclic Graphical models
MCC	Matthew's correlation coefficient
PDF	Probability Distribution Function
CDF	Cumulative Distribution Function
R-Vine	Regular Vine
C-Vine	Canonical Vine
D-Vine	Drawable Vine
VAR	Vector autoregression

CHAPTER 1

INTRODUCTION

Describing the biological network in term of the mathematical models has been increasing gradually in recent years as mathematical models can explain the model easier and catch the details better than other models. Furthermore, to make the models easy to understand to anyone, the graphical representation is used especially in the studies about the complex system's diseases. This study's purpose is to find the relationship between each variable which is the gene in the biological data sets. To model the network, the linear regression model is used in a way that each gene can be described as a linear combination of all other remaining relevant genes. So, the model parameters which are the regression model coefficients should be estimated by some mathematical or statistical ways. Then, it is seen that to infer the main parameters, the precision matrix is needed to be estimated when the data come from the normal distribution. In the case of non-Gaussian data or categorical data and etc., the copula approach is used to make the data normally distributed with the same structure. It is like the standardization or scaling the data. Once the main assumptions are provided, RJMCMC is suggested to estimate the precision matrix by using a Bayesian approach in three steps suggested by [11]. The application section of Chapter 2 includes some real data sets to illustrate the outperformance of the proposed method with its alternatives by using some accuracy measures.

The time series chain graphical model is a generalized form of Gaussian graphical models, and is gained by repeating the GGM through time. The mathematical description of the model, here, is a VAR(1) (vector of autoregression model with lag 1) model which includes the autoregressive coefficient matrix apart from the precision matrix. The mathematical description of this model is presented in Chapter 3.

Hereby, in this thesis, We proposed a semi-Bayesian and full-Bayesian RJMCMC to estimate the parameters of the underlying models. By applying to some simulated data, we are able to compute the accuracy of our proposed methods and compare them with the penalized likelihood method proposed by [1] in the application section of that chapter.

Until here, the main assumption has been the normality of the data so that we can use the exclusive properties of the normal distribution while estimating the parameters. In Chapter 4, we suggest a copula approach to discover the relationship between two genes (variable) by defining the most appropriate joint distribution according to Sklar's theorem [32] without using the normality constants in our calculations. Thereby, in order to diminish the complexity of the model, we propose the vine copula which decomposes the joint density function into bivariate densities. So with the help of this decomposition, each relationship can be investigated independently from all other pairs in a specific structure without any elimination of pair or particular assumption about distributions. Similarly, in Chapter 4, we present all of the underlying details and show the accuracy of the copula in inference of the networks. These results are also compared with proposed RJMCMC in Chapter 2 by using two special kinds of vine copula. Furthermore, in Chapter 4, we use two real data sets in order to illustrate the general kind of vine copula's accuracy.

Accordingly, we can summarize the aim of this thesis as below:

- In order to construct realistic and complex biological network models, we suggest the application of the Copula Gaussian Graphical model whose inference is conducted by the reversible jump MCMC algorithm. By this way, we can take into account the high correlation between system's elements, which are genes and proteins, their nonlinear and sparse relationships in the mathematical description of the systems. Then, we propose alternatives Bayesian algorithm besides RJMCMC in order to gain the computational efficiency without losing accuracy. For this purpose, we suggest the Gibbs sampling and compare out results with the Birth-and-Death method and QUIC (quadratic approximation for sparse inverse covariance estimation) approach.
- Later, we extend our model by including the time effect and suggest a fully

Bayesian, and semi-Bayesian approach for this new complex model, called time series chain graphical model, that is based on the normality assumption of the variables.

- Finally, we propose the vine copula approach for the construction of the underlying complex biological model in order to relax the normality assumption of the variables. For this purpose, we apply C-Vine, D-Vine and R-Vine copulas, generate a special plan to construct the model and estimate the model parameters step by step. At the end, we compare all these modeling approaches in terms of accuracy and computational time.

As a result, in the light of this thesis study, we aim to better understand the actual complexity of the biological networks and describe the true activation of the systems. We consider that the suggested approaches in this thesis can be also applicable in other networks in different fields, too.

CHAPTER 2

COPULA GAUSSIAN GRAPHICAL MODEL

Copula Gaussian Graphical model is used to show the relationship between variables graphically. In the biological data, the variables are genes and their relationship is denoted in terms of the conditional dependence. Among defined distributions, only in Gaussian, being correlated implies the independence. Therefore, we are interested in to work with the Gaussian distribution. The first section of this chapter is allocated to some definitions. In the second section, Copula Gaussian graphical model (CGGM) is defined with details. Furthermore, the model parameter is described to be estimated by the Reversible Jump Markov Chain Monte Carlo (RJMCMC) approach. RJMCMC was firstly introduced by Green [13] in the cases when the parameter dimension is not fixed. The third section consists some useful details about RJMCMC in the estimation of the precision matrix in three steps which were suggested by Dobra and Lenkoski [11] by using a Bayesian approach. Accordingly, in the third section, some powerful alternatives of RJMCMC are defined that were used in the application part, too.

Finally, in the application part, we use four real data sets to compare RJMCMC with its alternatives in terms of some accuracy measures like F_1-score and Mathew Correlation Coefficient (MCC).

2.1 Some fundamental definitions

Firstly, the Graphical model which is the base of CGGM is explained by its types in order to know what it shows or what is needed to be estimated and also to have a general view about their pros and cons. Then, the Gaussian graphical model is defined with its structure and the mathematical model with details to identify the parameter(s). Finally, Copula Gaussian graphical model is explained to show how it acts when the normality assumption does not hold for the data.

2.1.1 Graphical Model

Graphical models are used to make a better understanding of the models and also to observe the behavior of the system which is mainly divided by two categories:

- Directed Acyclic Graphs (DAG)
- Undirected Graphs

Before defining their mathematical structures, it is necessary to define child and parents relationships in graphs. When there is a direction in a graph from A to B which is shown as $A \rightarrow B$, A is called as the B's parent and B is named the A's child.

2.1.1.1 DAG

In Figure 2.1 you can see a directed acyclic graph. As seen in the figure, every

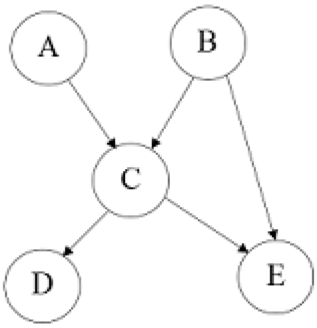


Figure 2.1: A directed acyclic graphical model with five variables.

node is independent of other nodes given its parents, children, and parents of children. So according to Figure 2.1, the joint distribution function can be written as

$P(A, B, C.D.E) = P(A)P(B)P(C|A, B)P(D|C)P(E|B, C)$.

Hence, in the coming sections of this study that we will see, the conditional dependence is a crucial issue as it can discover the regression coefficient defined as the model of the biological data. But in graphical model, there are two definitions for independence: marginally and conditionally independence. To see the difference between them, we suggest their graphical representations for three variables which is the simplest shown in Figures 2.2, 2.3 and 2.4.

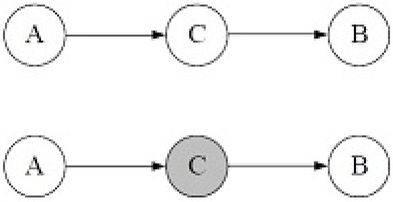


Figure 2.2: : A and B are marginally dependent (Top graph) and conditionally independent (Bottom graph).

2.1.1.2 Undirected graphs

As it is clear from its name, there is no direction between variables in this kind of graph. Hence, the main goal is to find a relationship between variables regardless of their directions. In the Gaussian Graphical model, the conditional dependencies are defined as their relationships under an undirected graph. Figures 2.5 and 2.7 illustrates some simple examples of the undirected graphs In this kind of graphs, the main assumption is that every node is independent of all other nodes given its neighbors. Accordingly, in the network represented in Figure 2.5, the joint density function can be decomposed as $f(A, B, C, D, E, F) = f_1(A, C)f_2(B, C)f_3(C, D, E)f_4(D, E, F)$

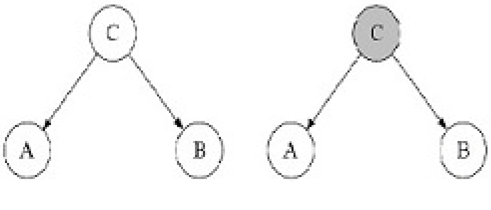


Figure 2.3: A and B are marginally dependent (Left graph) and conditionally independent (Right graph).

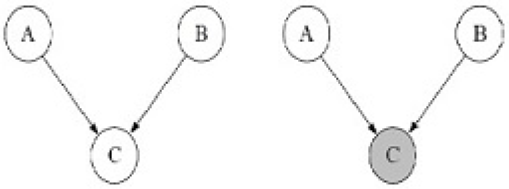


Figure 2.4: A and B are marginally independent (Left graph) conditionally dependent (Right graph).

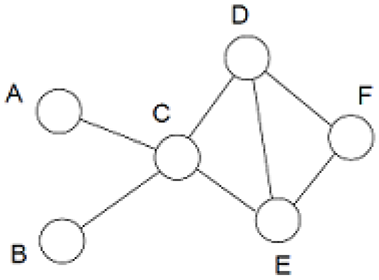


Figure 2.5: An example of an undirected graph with six variables.

where $f_1, ..., f_4$ are multivariate joint density functions with a lower dimension. By this way, the new density function becomes less complicated. Therefore, the advantage of a directed and undirected graph can be described as the separation of the complex structures of networks into small sub-network represented in Figure 2.6 and 2.7 in order.

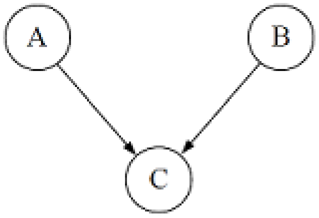


Figure 2.6: A representation of a marginal independence that an undirected graph cannot describe.

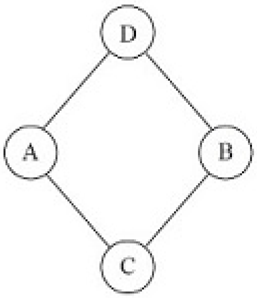


Figure 2.7: A representation of a conditional independence that a directed graph cannot describe.

2.1.2 Gaussian Graphical Models

Gaussian Graphical Model (GGM) is used to represent a model structure in a graphical way. The main assumption is the normality of the data, so that, it is called a Gaussian graphical model. In the future section of this chapter, it will be discussed by details with its alternatives, as well. Hence, to visualize the model, suppose that we have a data matrix with p variables and n samples and we are interested in obtaining the relationship between the variables. In this kind of network which is common in social surveys and biological fields, each variable is shown by a node in a graph and the conditional dependence between two nodes is shown by an undirected edge connecting those corresponding nodes. So a graphical model can be represented by $G = (V, E)$ where $V = 1, 2, ..., p$ and E is the set of available edges. Furthermore, here, undirected edge means that if $(i, j) \in E$ is equivalent to $(j, i) \in E$, Y_i and Y_j are taken as dependent given the remaining variables and denoted by

$$Y_i \not\perp\!\!\!\perp Y_j \mid Y_{V\{i,j\}},$$

where $V = \{1, 2, .., p\}$. This is called the pairwise Markov property.

The pairwise term comes from being investigated two by two and the Markov's statement comes from the property that the other remaining variable is taking account but not directly.

Accordingly, if we turn back the description of the model, we can assume that the data vector Y follows a p-dimensional multivariate normal distribution $N_p(0, \Theta^{-1})$ where Θ is the inverse of the covariance matrix K. With n samples, the likelihood function can be written proportional to

$$p(y^{1:n} \mid \Theta) \propto det(\Theta)^{n/2} exp\left\{-\frac{1}{2}tr(\Theta^T U)\right\},$$

where U is the trace of $Y'Y$ matrix, det(.) and tr(.) denote the determinant and trace of the matrix, respectively. Furthermore, $(.)^T$ shows the transpose of the matrix. So, a graphical model with V nodes and E edges, i.e., (V, E) from $N_p(0, \Theta^{-1})$ is called the Gaussian graphical model. To better understand the structure of the data, we can assume a $(n \times p)$-dimensional matrix like

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdot & \cdot & y_{1p} \\ y_{21} & y_{22} & \cdot & \cdot & y_{2p} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ y_{n1} & y_{n2} & \cdot & \cdot & y_{np} \end{bmatrix}$$

when the data are supposed to have a multivariate normal distribution with a mean vector $\mu = \{\mu_1, \mu_2, ..., \mu_p\}$ and the inverse covariance matrix which is called the precision matrix as

$$\Theta = \begin{bmatrix} \theta_{11} & \theta_{12} & \cdot & \cdot & \theta_{1p} \\ \theta_{21} & \theta_{22} & \cdot & \cdot & \theta_{2p} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \theta_{p1} & \theta_{n2} & \cdot & \cdot & \theta_{pp} \end{bmatrix}$$

In this description of the precision matrix Θ, Θ is a symmetric matrix meaning that $\theta_{ij} = \theta_{ji}$ and a positive definite matrix because of being the inverse of the covariance matrix [9]. So according to the Cholesky decomposition [10], this matrix can be decomposed into two upper and lower triangular matrices as $\Theta = \varphi\varphi^T$.

In the Gaussian graphical model (GGM), the nodes, also called states, are described by a multivariate normal distribution as shown above with a p-dimensional mean vector $\mu = (\mu_1, \mu_2, ..., \mu_p)$ and a $(p \times p)$-dimensional covariance matrix Θ for totally p nodes. Therefore, indeed, the precision matrix Θ is the expression to represent the conditional dependence between nodes in a way that the significantly large values point a highly possible dependency between the two related nodes given the remaining ones in the network that cannot be realized by the covariance matrix. So an exclusive property of the Gaussian distribution is that the covariance value determines the

correlation implying the dependent structure between variables. At the same time, its inverse realizes the conditional dependence structure.
Thereby, the mathematical description of the graphical model is denoted as below.

$$Y_p = \beta Y_{-p} + \varepsilon \tag{2.1}$$

where Y_p stands for the state of the pth node and Y_{-p} shows the states of all other nodes except the pth node. β is a vector of the regression coefficient associated to Y_{-p} and ε refers to the p-dimensional vector for the random error. Accordingly, the distribution of Y is denoted as

$$f(y|\mu, \Theta) = (2\pi)^{-\frac{n}{2}} \det(\Theta)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(y-\mu)^T \Theta (y-\mu)\right\} \tag{2.2}$$

Hence, in inference of this model, β has a direct relation with Θ via $\beta = \frac{\Theta_{-pp}}{\Theta_{pp}}$ in which Θ_{-pp} is the $((p-1) \times p)$-dimensional submatrix of Θ when the associated term of the pth node is discarded. Thus, the knowledge of β implies the knowledge of Θ, resulting in the information about the conditional dependency between the related nodes. Briefly, the "zero" elements of Θ implies non-significant or "zero" β. So, after now, to estimate the β values, it is enough to estimate the precision matrix.

2.2 Copula Gaussian graphical models

If the normality assumption does not hold for the data matrix, the copula can solve the problem by combining the data in a way that their joint distribution is Gaussian with the same covariance matrix. For binary and ordinal categorical data, [24] introduced a continuous latent variable Z by defining some increasing thresholds $\tau_\nu = (\tau_{\nu,0}, ..., \tau_{\nu,\omega_\nu})$. So,

$$y_v^j = \sum_{l=1}^{\omega_v} l \times 1_{\tau_{v,l-1} < z_v^j \leq \tau_{v,l}}. \tag{2.3}$$

The relationship between Y_{ij} and Z_{ij} satisfies the constraint below.

$$y_{ij} < y_{ik} \Rightarrow z_{ij} < z_{ik} \text{ and } z_{ij} < z_{ik} \Rightarrow y_{ij} \leq y_{ik}$$

Then, by defining the interaction of the correlation matrix in terms of Θ as

$$\gamma_{ij}(\theta) = \frac{\theta_{ij}^{-1}}{\sqrt{\theta_{ii}^{-1}\theta_{jj}^{-1}}} \tag{2.4}$$

and $Z_v \sim N_p(0, \Theta^{-1})$, we can get a one-to-one correspondence with observed data via

$$Y_i = F^{-1}\left(\Phi(\frac{Z_i}{\sqrt{\theta_{ii}^{-1}}})\right). \tag{2.5}$$

In Equation 2.5 , θ_{ii} and θ_{jj} indicate the diagonal entries of the ith and the jth node, in order. Accordingly, θ_{ij} means the precision value between the ith and the jth node. On the other hand, in Equation 2.5, F^{-1} and Φ stand for the inverse of the cumulative distribution function (CDF) and CDF of the normal distribution, respectively. Hence, by denoting $C(u_1, ..., u_p|\gamma)$ as the Gaussian copula with $(p \times p)$-dimensional correlation matrix for the p random sample from the standard uniform distribution, we have

$$p(Y_1 < y_1, ..., Y_p < y_p) = C(F_1(y_1), ..., F_p(y_p)|\gamma(\Theta)). \tag{2.6}$$

Roughly speaking, by the thresholds, the old value is going to be projected to the normal distribution by ordering them and choosing the best normally distributed value that fits according to its position in the density function.

In this study, we decompose the multivariate normal distribution of the states via the Gaussian copula model with the normal marginal distributions. This new probability distribution function is used in the calculation of the likelihood within the Bayesian framework which will be explained by details later in this chapter. As discussed before in Section 2.1.2, the base of the model is a linear regression model in which to estimate the coefficient parameter β the precision matrix is needed to be estimated according to the knowledge about the data.In order to infer the precision matrix, several methods are suggested in the literature, such as the maximum likelihood (MLE) biased estimator [33], penalized MLE [6], least absolute selection and shrinkage Operator (LASSO) regression [36] and other parametric or robust estimation methods [25]. But in some cases using those methods could be difficult due to the larger number of parameters (the elements of the precision matrix is equal to $\frac{d(d-1)}{2}$ where p is the number of variables (columns) in data set) towards a smaller number of samples or in a case of sparsity which is very common in biology is the case. If we define a matrix with some values regarding the relationship between every two variables, the sparsity rate is the ratio of zero values in the upper triangular part of the matrix to the number of the parameter in the precision matrix. In most of the biological data set, in spite of a large number of variables (genes), only some of them have some relationship which means that we deal with a sparse adjacency (an upper triangular matrix with one and zero elements) matrix. Furthermore, in some cases, the number of samples is less than the number of parameters. So, the Markov Chain Monte Carlo method can be applied to these kinds of cases which start with a random matrix and in each step and then it becomes closer to the true (target) matrix iteratively. The well-known property of the Markov chain model is that in each step or iteration is independent on all other previous iteration except the last one.That is, only the previous step will be accounted. Thus, the Monte Carlo method means by taking the mean of random samples and with the help of the law of large numbers the estimation converges to the real value of the parameter. By taking into account of the explanation about the Markov Chain and Monte Carlo method, we have a general image of the MCMC process.

Accordingly, to estimate the precision matrix by the MCMC method, we cannot use it directly, as MCMC is appropriate for the model with a fixed number of parameters. But in the precision matrix, the dimension depends on the non-zero elements of the upper-triangular part of the symmetric matrix. So RJMCMC is suggested by [13] which deals with changing dimensions in each iteration. Therefore in an article published by Dobra and Lenkoski [11] by updating the matrix and the graph in the separate steps by using some latent variables, an accurate estimation is achieved for some economical data set.

In the following section, the proposed RJMCMC [11] method is explained. Followingly, some powerful alternative of RJMCMC will be listed. Finally, in the application section, we applied RJMCMC to some biological data sets as well as economical data to measure its accuracy according to the true graph of the data and compare the accuracy of RJMCMC with some of its explained alternatives.

2.3 Reversible Jump Markov Chain Monte Carlo method

A Bayesian approach is used for a better estimation of the precision matrix as an estimation done by Bayesian is more robust because it is a linear combination of the estimated parameter based on the prior distribution and MLE. As pointed beforehand, the precision matrix is the inverse of the covariance matrix. On the other hand, for one-parameter model, if $Y \sim N(\mu, \sigma^2)$ then $\frac{(v-1)s^2}{\sigma^2}$ is Chi-squared distributed via $\chi^2_{(n-1)}$ where n is the number of samples and s^2 denotes the sample covariance. So the inverse of σ^2 has some chi-square distribution. The generalized version of the multivariate χ^2 is G-Wishart distribution which is used as a good prior distribution for the precision matrix. The G-Wishart distribution with parameters D and δ knowing the graph(G) is in the form of $p(\Theta|G) = \frac{1}{I_G(\delta,D)} \det \Theta^{(\delta-2)/2} \exp\left\{-\frac{1}{2}\text{tr}(\Theta^T D)\right\}$ with some normalization constant $I_G(\delta, D)$ which is not straightforward to obtain when the graph is not a full graph (all of the elements are non-zero). It that circumstance, another MCMC approach is used to obtain it through an algorithm suggested by Lenkoski [18].

Another interesting property of the G-Wishart distribution is that it is conjugate with a normal distribution such that if $\Theta \sim Wishart(\delta, D)$ and $y = (y_1, ..., y_n) \sim MVN(0, 1)$, then the posterior distribution of Θ for the given y is $(\Theta|y) \sim Wishart(n+\delta, D + U)$.

Another fact about the G-Wishart distribution is that the precision matrix can be decomposed by the Cholesky decomposition method into two normally distributed matrices $\Theta = \varphi^T \varphi$ where φ is a upper triangle matrix which was discussed in Section 2. Each zero element in φ implies a zero element in Θ for the corresponding variables. Hence, the main goal in the estimation of the precision matrix is to discover if there is a conditional dependence between every two variables or not. Furthermore, the goal can change the process in order to see the zero and non-zero elements in the final or estimated graph. Therefore, instead of working with the G-Wishart distribution with its normalization constant and other difficulties, in some steps of the RJCMCM, it is preferred to use normal distribution by applying the Cholesky decomposition.

The main idea about the relationship between Gaussian and G-Wishart distribution comes from the one-parameter version of normal and Chi-square in such a way that if $Y \sim N(0, 1)$ then $Y^2 \sim \chi^2_{(n-1)}$ where n is the length of the Y vector. So, the way of how χ^2 can be rooted in normally distributed variables shows us the Cholesky decomposition of the G-Wishart distribution into two normally distributed matrices. Another method that was used in the proposed algorithm by Dobra and Lenkoski [11] in three steps, is the Metropolis Hasting algorithm [14]. In this algorithm to do any change in the value of a matrix, a ratio is calculated and if the ratio is bigger than any random numbers between zero and one, then that move is done. Otherwise, there happens no change. To better understand the Metropolis Hasting algorithm, by defining $\pi(x)$ and $\pi(y)$ as the target density of data and q as the conditional density, also called the proposal kernel or the candidate kernel, the algorithm can be also written in the following way

- Calculate the ratio $p = \frac{\pi(y)q(x|y)}{\pi(x)q(y|x)}$.
- Generate a random number ω in the interval $(0, 1)$
- If $p > \omega$, then x will be changed to y.
- Otherwise, x remains without change.

Hence, we can describe the three-steps RJCMC algorithm of Dobra and Lenkoski [11] as below.

2.3.1 Resample the latent data

In this step, we transform the original variables to some latent variables, called as Z. Thus, Z is a $(n \times p)$-dimensional matrix and for each column, which is related to each node, we calculate its minimum L and its maximum U as the vectors of p elements based on the original variables.

In this calculation, by using the θ matrix and L as well as U vectors, we generate another Z_is from a truncated normal in the L_i and U_i distributions to control the fluctuations in each iteration in a way that

$$Z_i \mid Z_{V\setminus i} \sim N(\mu_i, \sigma_i^2)$$

where $Z_i \mid Z_{V\setminus i}$ denotes the ith variable conditioned on all other variables and $\mu_i = -\sum_{y\in bd(i)} \frac{\theta_{i,y}}{\theta_{i,i}} z_{y,j}$ for $bd(i) = \{y \in (1,\ldots,p) : (i,y) \in E\}$ when $E = \{(i,y)|\theta_{i,y} \neq 0, i \neq y\}$ and $\sigma_i^2 = \frac{1}{\theta_{i,i}}$. In the next step of the algorithm, the underlying $z_{i,j}$s are used.

2.3.2 Resample the precision matrix

The inverse of the covariance matrix is computed by using the latent variables from step 1. Firstly, the precision matrix Θ is decomposed by the Cholesky decomposition method. Then, for each element of φ, the following calculations are repeated:

(i) For non-zero diagonal elements Metropolis-Hasting update of φ is done by sampling a γ from a normal distribution truncated below at zero (because for diagonal elements we cannot use any negative value) with a mean $\varphi_{i,i}$ and a variance σ_p^2. Then, γ is replaced to the related diagonal elements of φ and φ is transformed to φ' with a probability $\min\{R_p, 1\}$ where

$$R_p = \frac{\Phi(\varphi_{i,i}/\sigma_p)}{(\gamma/\sigma_g)} \left(\frac{\gamma}{\varphi_{i,i}}\right)^{\delta+n+nb(i)-1} R'_p \tag{2.7}$$

for $R'_p = \exp\left\{-\frac{1}{2}\mathrm{tr}(\theta' - \theta)^T(D + \mathrm{tr}(Z^T Z))\right\}$.

(ii) For non-diagonal elements of φ, a new γ' is sampled from $N(\varphi_{i,j}\sigma_p^2)$. In these cases, φ is transformed to φ' with a probability $\min\{R'_p, 1\}$.

2.3.2.1 Resample the graph

In this step, only one element of the decomposed matrix, $\varphi_{i,j}$ is selected randomly. In this selection,

(i) If there is no edge between Y_i and Y_j, it will be changed by a value from $N(\varphi_{i,j}, \sigma_p^2)$ in φ with a probability $\min\{1, R_p\}$ where

$$\begin{aligned} R_p = \sigma_p \sqrt{2\pi}\varphi_{i,i} \frac{I_G(\delta, D)}{I_{G'}(\delta, D)} \times \\ \exp\left\{-\frac{1}{2}\text{tr}((\Theta' - \Theta)^T(D + \text{tr}(Z^T Z)) + \frac{(\varphi'_{i,j} - \varphi_{i,j})^2}{2\sigma_p^2}\right\}. \end{aligned} \tag{2.8}$$

(ii) If there is an edge between Y_i and Y_j, it will be replaced by a zero in φ with a probability $\min\{1, R'_p\}$ where

$$\begin{aligned} R'_p = (\sigma_p \sqrt{2\pi}\varphi_{i,i})^{-1} \frac{I_G(\delta, D)}{I_{G'}(\delta, D)} \times \\ \exp\left\{-\frac{1}{2}\text{tr}((\Theta' - \Theta)^T(D + \text{tr}(Z^T Z)) + \frac{(\varphi'_{i,j} - \varphi_{i,j})^2}{2\sigma_p^2}\right\}. \end{aligned} \tag{2.9}$$

We repeat the steps by updating the latent variables, the precision matrix and the graph in each iteration while starting with a random matrix as the initial matrix to achieve the best posterior distribution meaning that the convergence is obtained if the difference between the graphs is the iterated steps is infinitesimal. During this convergence regarding the dimension of the estimated precision matrix, a large number of MCMC iterations can be taken as burn-in such as half of the whole iteration number. Then after discarding the burn-in period, the average of the remaining iterations is taken as the point estimate of the model parameters. In the final stage, in order to convert the estimated precision into the binary form, we determine a threshold value by considering the sparsity rate of the target graph.

2.4 RJMCMC Alternatives

In this section, some alternatives of RJMCMC are introduced. Most of them are used in the application part to compare the accuracy of RJMCMC by some measures.

2.4.1 Birth-Death MCMC

This method is introduced by Mohammadi and Wit [21] based on the continuous time approach where the dimension of the parameter is not fixed. In this approach, new components are born according to the Poisson process with a rate λ_B and the ith component in a k- component configuration which dies with a rate

$$\lambda_D(i) = \frac{\pi(k-1, \theta_{1:i-1}, \theta_{i+1:k})}{\pi(k, \theta_{1:k})} \times \lambda_B q(\theta_i). \tag{2.10}$$

In Equation 2.10, $\pi(.)$ is the density kernel when $\theta_{1:k}$ implies the first k parameters and $q(\theta_i)$ represents the proposal kernel for the $i$the component of the parameter θ, as used previously.

The choice of the birth and death rates determines the birth-death process and is made in such a way that the stationary distribution is precisely the posterior distribution of interest. Contrary to the RJMCMC approach, the moves between models are always accepted, which makes the BDMCMC approach extremely efficient and fast.

2.4.2 Carlin-Chib algorithm

In the application of the MCMC technology to any problem involving a choice between K competing the Bayesian model specification, M is defined as an integer-valued parameter that indexes the model collection. The Carlin-Chib algorithm [7] shows how the Gibbs sampling methodology may be a specific method to choose across finite collections of models without destroying the convergence.

Thereby, suppose that $f(y|\theta_j, M = j)$ is the corresponding likelihood of the model j and $P(\theta_j|M = j)$ is the prior distribution of the parameter under the model j. Here, y is independent on $\theta_{i \neq j}$ given that $M = j (j = 1, ..., k)$. As it is mentioned before, M is a model indicator and for the given M, various θ_j's are assumed to be completely independent.

By defining $\pi_i = P(M = j)$ such that $\sum_{j=1}^{k} \pi_j = 1$, the joint distribution of y and θ when $M = j$ is as below.

$$P(y, \theta, M = j) = f(y, \theta_j, M = j) \times \pi_j \times \{\prod_{i=1}^{k} P(\theta_i|M = j)\}. \tag{2.11}$$

The following equation shows the full conditional independence of each θ_j and M.

$$P(\theta_j|\theta_{i \neq j}, M, y) \propto \begin{cases} f(y|\theta_j, M = j) P(\theta_j|M = j) & \text{for } M = j, \\ P(\theta_j|M \neq j) & \text{for } M \neq j, \end{cases} \tag{2.12}$$

where $P(\theta_j|M \neq j)$ is called *"pseudoprior"*. When $M = j$, and as the name suggests, pseudoprior is not really a prior but only a conveniently chosen linking density,

required to define completely the joint model specification. we generate the graph from the usual model of the full conditional distribution and when $M \neq j$, we generate from the linking density.
Hence for the model M, we have

$$P(M = j|\theta, y) = \frac{f(y|\theta_j, M = j)\prod_{i=1}^{k} p(\theta_i|M = j)\pi_j}{\sum_{n=1}^{k} f(y|\theta_n, M = n)\prod_{i=1}^{k} p(\theta_i|M = n)\pi_n}.$$

In the usual condition, the algorithm produces samples from the correct joint posterior distribution. In particular, the ratio

$$\hat{P}(M = j|y) = \frac{\text{the number of } (M^{(g)} = j)}{\text{total number of } M^{(g)}}$$

is a simple estimate to compute the Bayes factor between any two of models while g denotes the number of samples. Thus, $M^{(g)} = j$ means the jth model for the gth sample.

2.4.3 Gibbs Sampling

As discussed earlier, RJMCMC is the modified version of the Metropolis-Hasting method which provides jumps between spaces of different dimensionality. In this algorithm, the move from (k, θ^k) to $(k', \theta^{k'})$ is not always possible because of the modality of the Metropolis-Hasting calculation. The acceptance probability for this movement is computed as

$$R_{k,k'} = \frac{P(k', \theta^{k'}|y)\tilde{q}(k, \theta^k|k', \theta^{k'})}{P(k, \theta^k|y)\tilde{q}(k', \theta^{k'}|k, \theta^k)}, \tag{2.13}$$

where k is the dimension of the precision matrix θ and y refers to the normal random variables of the current position and finally, k' and θ' denote the associated proposal terms of k and θ, respectively. Hereby, in Equation 2.13, P shows the likelihood function and $\tilde{q}$ presents the kernel density.
Under the “*dimension matching*" condition, $(dim(\theta^{k'}, x) = dim(\theta^k, y))$, where x and y are variables drawn from the proposal distribution $\tilde{q}_1$, the acceptance probability is equivalent to

$$R_{k,k'} = \frac{P(k', \theta^{k'}|x)}{P(k, \theta^k|y)} \times \frac{\tilde{q}_1(k|k')\tilde{q}_2(x)}{\tilde{q}_1(k'|k)\tilde{q}_2(y)} \times \left|\frac{\partial(\theta^{k'}, x)}{\partial(\theta^k, y)}\right|. \tag{2.14}$$

In Equation 2.14, $\tilde{q}_2(.)$ refers to the kernel for the given random variable and $\left|\frac{\partial(\theta^{k'},x)}{\partial(\theta^k,y)}\right|$ represents the determinant of the Jacobin matrix.
By using the Bayes theorem, a complete model for a joint density for $j = 1, 2, ...$ can be written as

$$p(y, \theta^j, k) = p(y, \theta^k, k)p(\theta^1|\theta^2)..p(\theta^{k-1}|\theta^k)p(\theta^{k+1}|\theta^k)p(\theta^{k+2}|\theta^{k+1})... \tag{2.15}$$

If we denote π_k as the prior distribution for the unknown dimension of a parameter k and $\pi_k(\theta^k)$ as the prior distribution for $\theta^k|k$, we can represent the joint distribution of the state y with a model parameter θ under the k dimension via

$$p(y, \theta^k, k) = p(y|\theta^k, k)\pi_k(\theta^k)\pi_k. \tag{2.16}$$

Here, to move between dimensions, we have infinity choices that cause the precise probabilities which cannot be found. To solve the problem, Walker (2009) [34] introduces an auxiliary variable u which helps us to have finite choices to move between dimensions. On the other hand, the latent variable u has a distribution in which $u = k$ with a probability q and $u = k + 1$ with a probability $1 - q$.
Since u depends only on k and the complete model can be stated as

$$p(u, y, \theta^j, k) = p(u|k)p(y, \theta^j, k). \tag{2.17}$$

Thereby, the steps of the algorithm base on the Gibbs sampling method which is explained first by [12], can be listed as below:

1- Sample $\theta^{(k)}$ from $\pi_k(\theta^k|y, k)$.
Sample $\theta^{(k+1)}$ from $p(\theta^{k+1}|\theta^k)$ and sample $\theta^{(k-1)}$ from $p(\theta^{k-1}|\theta^k)$.

2- Sample u from some kind of a binomial distribution in which $p(u = k+1) = q$ and $p(u = k) = 1 - q$.

3- For the given k, sample j, which will be the next k, from the distribution below:

$$\begin{aligned}
j = k|u = k + 1 \quad & \propto (1 - q)p(y, \theta^{k+1}, k + 1)p(\theta^k|\theta^{k+1}). \\
j = k + 1|u = k + 1 \quad & \propto qp(y, \theta^k, k)p(\theta^{k+1}|\theta^k). \\
j = k|u = k \quad & \propto (1 - q)p(y, \theta^k, k)p(\theta^{k-1}|\theta^k). \\
j = k - 1|u = k \quad & \propto qp(y, \theta^{k-1}, k - 1)p(\theta^k|\theta^{k-1}).
\end{aligned}$$

In these expressions, the sampling strategy is simplified by the following equality.

$$P(\theta^k|\theta^{k+1}) \times \pi_{k+1}(\theta^{k+1}) = P(\theta^{k+1}|\theta^k) \times \pi_k(\theta^k) \tag{2.18}$$

that is valid under the Gaussian Copula graphical model. So the simplified version of the third step of the algorithm can be shown as follows.

$$\begin{aligned}
&j = k|u = k+1 && \propto (1-q)p(y|\theta^{k+1}, k+1)\pi(\theta^{(k+1)}).\\
&j = k+1|u = k+1 && \propto qp(y|\theta^{k}, k)\pi(\theta^{(k)}).\\
&j = k|u = k && \propto (1-q)p(y|\theta^{k}, k)\pi(\theta^{(k)}).\\
&j = k-1|u = k && \propto qp(y|\theta^{k-1}, k-1)\pi(\theta^{(k-1)}).
\end{aligned}$$

2.4.4 Quadratic Approximation for Sparse Inverse Covariance Estimation

This algorithm is suggested by Hsieh et al (2014) [17] to estimate the inverse of a sparse covariance matrix where the data are Gaussian. In this calculation, there is a penalty term in the general formula which controls the sparsity rate of the related graph. By increasing the underlying term, the precision matrix becomes more sparse. Hereby, in the algorithm, let Y be an $(n \times p)-$dimensional data matrix and the sample covariance matrix is denoted by

$$S = \frac{1}{n-1}\sum_{k=1}^{n}(y_k - \hat{\mu})(y_k - \hat{\mu})^T \tag{2.19}$$

where $\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} y_k$.

Given the regularization penalty term $\lambda > 0$, the regularized log-determinate is defined as below.

$$argmin\{-\log|Y| + \text{tr}(SY) + \lambda\sum_{i,j=1}^{p}|Y_{ij}|\}. \tag{2.20}$$

Here, tr and $\log$ show the trace of the matrix and logarithm, respectively, and $|Y|$ denotes the determinant of Y. Then, the algorithm computes the optimal Λ by taking the $(p \times p)$-dimensional empirical covariance matrix S, which is positive semi-definite, and the regularization matrix Λ as inputs and initializing Y on the first iteration via $Y_0 > 0$ in $t = 0, 1, ...$ until reach the best Y_t that minimize the function in Equation 2.20.

2.5 Application

In this section to compare its accuracy with some or all of the explained alternatives we use three data sets: the Ovarian cancer data, the Rochdale data, and Cellsignal data. The accuracy measures which are used in this part are F_1-score and Matthew's correlation coefficient (MCC) whose definitions are listed below.

$$F_1 - \text{score} = \frac{2\text{TP}}{(2\text{TP} + \text{FP} + \text{FN})}, \tag{2.21}$$

$$MCC = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP+FP})(\text{TP+FN})(\text{TN+FP})(\text{TN+FN})}}, \tag{2.22}$$

In Equations 2.21 and 2.22, TP is the number of truly found edges, FP presents the numbers of falsely found edges (similar to type one error) and FN denotes the number of edges that exist, but are not recognized (similar to type two error). They create a matrix called the confusion matrix which can be shown in a very simple way in Table 2.1. As an explanation, the perfection level of F_1-score is 1 and the range lies from 0 to 1. On the other hand, the Matthew's correlation coefficient is also known as *phi coefficient* turns a value between -1 and $+1$. A coefficient $+1$ represents a perfect prediction, 0 implies no better than a random prediction and -1 indicates the total disagreement between the prediction and the observation.

Table2.1: Definition of the confusion matrix

	Positive edge in true graph	Negative edge in true graph
Estimated edge	True Positive	False Positive
Non-estimated edge	False Negative	True Negative

In the last part of this section, we try to estimate the precision matrix under singularity situation which is one of the common obstacles in data analyses. In our calculation, we used one data set in which most of the variables are highly correlated.

2.5.1 Ovarian Cancer data

In this analysis, weapply the gynecological cancer data. The gynecological cancer consists of the ovarian, cervix and endometrial cancer and this cancer type is the second most common cancer in women in the world after breast cancer. In our study, we use 11 core genes that validated in the literature that they are active in this cancer type. These genes are named as MPK2K1, MK01, CEBPB, CTNNB1, TFAM, TP53, PDIA3, IMP3, ERBB2, CHD4, and MBD3. Then, from the ArrayExpress database [26], we take an Affymetrix dataset, which is collected under ovarian cancer, and choose the observations belonging to the underlying 11 genes.
In the data, each gene has 14 samples and the true network composed of these genes is complete, i.e., its adjacency matrix has the value one in all entries. In the estimation, 10,000 MCMC iterations are conducted and the first 2,000 runs are discarded as the burn-in period. From the outcomes, we calculate F_1-score=1 for RJMCMC and F_1-score=0.79 for BDMCMC. Thereby, as observed from other analyses, the findings show that RJMCMC overperforms BDMCMC with a higher accuracy.

2.5.2 The Rochdale data

The second data set which is is implemented in our study is presented in Table 2.2. Thus data set is a binary dataset collected from 665 samples in order to assess the relationship among eight factors affecting the women's economic activity. The data presents the eight binary (yes or no) factors that influence women activities, which are named by a: wife economically active, b: wife' age > 38, c: husband unemployed, d: the number of children 4, e: education level of wife, (high-school+), f: education level of husband (high-school+), g: Asian origin, and h: other household member working.

For instance, the first cell of Table 2.2 shows that 5 of the 665 persons, a=1, b=1, c=1, d=1, e=1, f=1, g=1, h=1 and also, for 57 persons in the 9th row and the 13th column, a=2, b=1, c=1, d=1, e=2, f=2, g=1, h=1. Accordingly, the true network based on the study by Wittaker (1990) [35] is in the form of f g, e f, dh, dg, cg, c f, ce, bh, be, bd, ag,

Table2.2: The Rochdale data

5	0	2	1	5	1	0	0	4	1	0	0	6	0	2	0
8	0	11	0	13	0	1	0	3	0	1	0	26	0	1	0
5	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	8	2	6	0	1	0	1	0	1	0	0	0	1	0
17	10	1	1	16	7	0	0	0	2	0	0	10	6	0	0
1	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0
4	7	3	1	1	1	2	0	1	0	0	0	1	0	0	0
0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
18	3	2	0	23	4	0	0	22	2	0	0	57	3	0	0
5	1	0	0	11	0	1	0	11	0	0	0	29	2	1	1
3	0	0	0	4	0	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	25	0	1	37	26	0	0	15	10	0	0	43	22	0	0
0	0	0	0	2	0	0	0	0	0	0	0	3	0	0	0
2	4	0	0	2	1	0	0	0	1	0	0	2	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ae, ad, ac. Table 2.3 indicates the estimated graph in term of a symmetric $0-1$ matrix.

Table2.3: The estimated graph for the Rochale data

	a	b	c	d	e	f	g	h
a	0	0	1	1	1	0	1	0
b	0	0	0	1	1	0	0	1
c	1	0	0	0	1	1	1	0
d	1	1	0	0	0	0	1	1
e	1	1	1	0	0	1	0	0
f	0	0	1	0	1	0	1	0
g	1	0	1	1	0	1	0	0
h	0	1	0	1	0	0	0	0

Hereby, in Table 2.3, 1 refers to the existence of an edge between two related nodes and 0 stands for the conditional independence between two nodes. The accuracy of RJMCMC and some alternatives are shown in Table 2.4.

In our calculation, the number of iteration for RJMCMC, BDMCMC, and Gibbs is taken as 10^6 and for QUIC, it sets to 1000 iterations. According to the results presented in Table 2.4, it is seen that after QUIC, Gibbs sampling and RJMCMC have the same highest accuracy among alternatives. Here, although QUIC is the most speedy

Table2.4: The comparison between accuracies of different methods for the Rachdale data

Methods	TP	FP	FN	TN	F_1-score	MCC
True graph	13	0	0	15	1	1
RJMCMC	12	1	1	14	0.923	0.856
BDMCMC	10	9	3	9	0.625	0.272
Gibbs ($q = 0.5$)	12	2	1	13	0.888	0.787
QUIC ($\lambda = 0.12$)	13	2	0	13	0.928	0.866

method, it is completely non-parametric and suggests a numeric solution for the inference of the precision.

On the other hand, the remaining approaches are fully parametric and can be grouped in the same class. Whereas, if we compare the computational demand of RJMCMC and Gibbs sampling, it is seen that the Gibbs sampling reduces the computational time of RJMCMC significantly without losing accuracy.

2.5.3 The CellSignal Data

This dataset firstly was investigated by Sachs et al., (2005) [30] and it is attached to the *BDgraph* package [22] in 11672 samples in which each independent measurement consists of quantitative amounts of each of the 11 phosphorylated molecules, simultaneously measured from single cells. Figure 2.8 shows the true undirected network of the 11 genes via the Bayesian network analysis to the multivariate flow of the cytometry data.

Moreover, the performance of the accuracy in the estimation of the precision matrix via all the suggested approaches is shown in Table 2.5. According to the results in Table 2.5, it can be seen that the Gibbs sampling performance is better than others

Table2.5: The comparison between accuracy of different methods for the CellSignal dataset

Methods	TP	FP	FN	TN	F_1-score
True graph	17	0	0	38	1
RJMCMC	13	27	5	10	0.448
BDMCMC	16	32	5	2	0.485
Gibbs ($q = 0.5$)	18	37	0	0	0.493
QUIC ($\lambda = 0.2$)	10	22	8	31	0.40

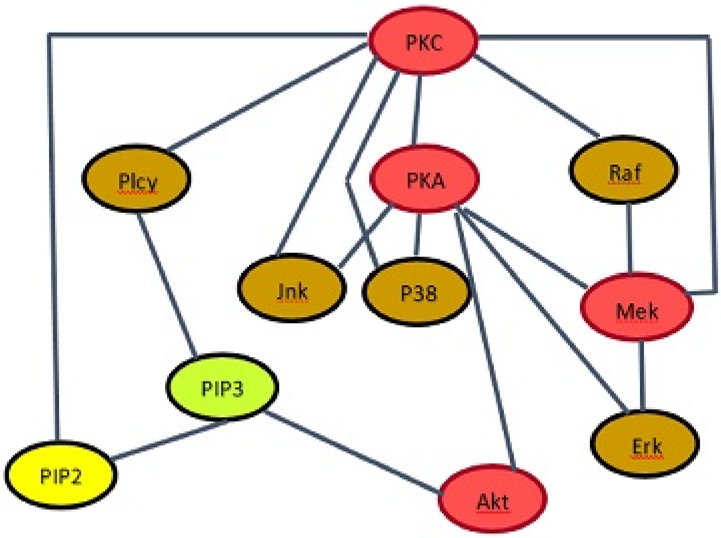


Figure 2.8: The true undirected network of the CellSignal data

by F_1-score. So, the Gibbs sampling not only needs less time in the comparison with RJMCMC, its performance is better than RJMCMC and BDMCMC.

2.5.4 The geneExpression data

Singularity Problem

Here, we introduce our suggested approach to deal with the problem of singularity in the inference of the RJMCMC algorithm. This challenge is commonly observed if the calculation is done for real biological systems. The reason is that from the description of the real systems, some of the proteins or genes can be defined in terms of other components in the systems and this can happen typically if these proteins are only seen in the product or reactant side of the complete reaction list at one time. In this case, since the firing of the underlying protein is merely dependent on a particular single reaction and its associated reactants, the change in the concentration of this protein can be explained by means of those underlying proteins during whole biological activations. Accordingly, the variance-covariance matrix of the system becomes singular as the associated column (or row) of the matrix can be found via the linear combination of other columns (or rows). Under this condition, since the inverse of the matrix cannot be computed, the likelihood in RJMCMC cannot be calculated too. Furthermore, the underlying singularity does not only result in a problem in the likelihood, but also, results in an infeasible candidate generator due to the linear dependence on some of the state values that directly affect the acceptance probabilities R_p and R'_p which cannot be defined under these conditions. In order to unravel this problem, we propose a pre-processing step in advance of RJMCMC by checking the eigenvalues of the correlation matrix of the raw data. The entries one in this matrix present a perfect correlation between the corresponding pair of nodes, leading to zero eigenvalues in the associated variance-covariance matrix. Therefore, we eliminate these columns and associated rows from the variance-covariance matrix as they are formed from a linear combination of some independent columns. On the other hand, in the statistical sense, such elimination does not lead to any loss of information in the inference because of the fact that the perfect correlation implies the certainty in the calculation of the likelihood if all independent components are included in the computation. Hereby, in the RJMCMC algorithm, we apply the inverse of the reduced variance-covariance matrix as the starting precision matrix of RJMCMC. Then, following by the estimated precision matrix as the mean of the entries of matrices at the end of each iteration after the burn-in period, we get its inverse as the estimated variance-covariance matrix. Finally, by using the knowledge of the linear dependency obtained from the initial correlation matrix which are the coefficients of the linear expression assigned to the eliminated columns, we fill again the reduced estimated variance-covariance matrix. At this stage, since our ultimate aim is to infer the structure of the network which has a binary form, i.e., zero and one entry, we do not lead to an overestimation in the final estimated adjacency matrix. As a result, we do not change the biological validation of the system and we prevent the loss of information in the interpretation of the biological explanation.

Description of data

The "geneExpression" dataset is freely available in the "bdgraph" package [20] in the R programming language. This dataset contains the human gene expressions of 100 transcripts (with unique Illumina TargetID) measured on 60 unrelated individuals. The genotypes of the proteins can be found from the Sanger Institute website at ftp://ftp.sanger.ac.uk/pub/genevar. For this dataset, even though the true network has not been known yet, resulting in the complete list of all interactions cannot be

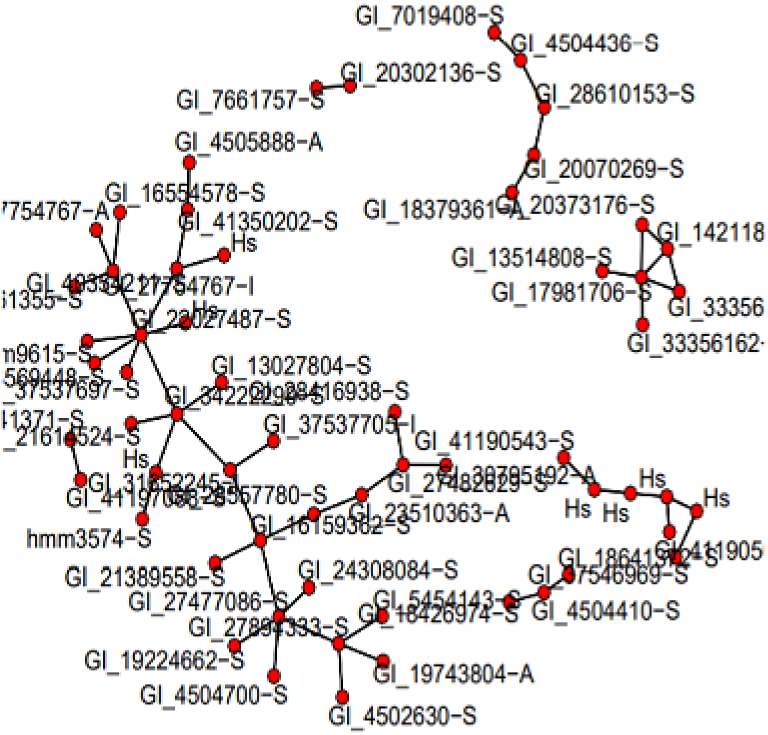


Figure 2.9: Biologically validated links of the gene expression data according to the study of [4]

validated, 55 links in this system can be biologically controlled by using the study of [4]. In Figure 2.9, we present the links that can be biologically validated. Hereby, in the detection of the singularity over 100 proteins, in the system, we eliminate 49 of them and conduct the inference based on 51 proteins, each has 60 samples. In the computation, the number of iterations is set to 10000 MCMC runs and the first 2000 iterations are discarded as the burn-in period. From the results in Table 2.6, it is seen that RJMCMC which takes into account the structural dependency in the systems, is more accurate than the findings of BDMCMC in terms of F_1-score and MCC values while BDMCMC can be applicable without eliminating those components. On the other hand, if we compare the computational demand of both approaches, we see that RJMCMC uses more CPU time but we think that the advantage of BDMCMC is not due to the plausible high computational demand of the improved RJMCMC. Further, it may be caused by the programming language of each algorithm. The RJMCMC approach is originally written in R which is an interpreted language, whereas, BDMCMC is written in C which is a compiled language.

So we could propose an alternative solution to infer complex protein-protein interac-

Table2.6: The comparison between accuracy of different methods for the CellSignal dataset

Methods	TP	FP	FN	TN	F_1-score	MCC	CPU
True graph	175	0	0	4775	1	1	-
RJMCMC	130	1497	45	3275	0.144	0.168	0.37
BDMCMC	175	4392	0	383	0.074	0.055	0.03
BDMCMC2	57	384	118	4391	0.1857	0.159	0.37

tion networks that have highly dependent components. We have considered a two-step calculation in the estimation. At the first step, we have eliminated highly correlated proteins and kept their linear relationships, with other components and then the RJMCMC algorithm to construct a smaller dimensional pathway. Later, we have included those dependent proteins in the estimated precision matrix which is enrolled by the Gaussian graphical model.

CHAPTER 3

TIME SERIES CHAIN GRAPHICAL MODELS

3.1 Introduction

In this chapter, we particularly deal with a graphical representation of a generalized version of the Gaussian Graphical models which was introduced and discussed in Chapter2. This extended model enables us to combine the observations from different time points and it can detect the relationship between variables (genes) in different lags. In the study of Abegaz and Wit [1], the model parameters of this model, called the time series chain graphical model (TSCGM), is estimated by the penalized likelihood approach under the state-space model defined by Sima et al. (2009) [31]. Here, we use the Pearson correlation for the autoregressive coefficient correlation matrix which shows the time dependency structure between variables as the data assumed to be normally distributed not only in each time step but also, between time steps. We suggest the reversible jump Markov chain Monte Carlo (RJMCMC) algorithm [11,13] to estimate the plausible interactions between the systems' elements. Thereby, this paper first gives a brief overview of RJMCMC and the Copula Gaussian graphical Model (CGGM) [15, 19, 20] and RJMCMC which is the fundamental modeling of TSCGM. Section 3.2 begins by laying out the theoretical dimensions of the research and looks at how the state-space model fits TSCGM, the definition of RJMCMC that is previously inserted in CGGM for CGGM and TSCGM as well. Later, we continue our proposed method based on RJMCMC for the generalized CGGM with/without tuning parameter. To examine the performance of all of the mentioned methodologies, we used simulated data sets in the application section.

3.2 Time Series Chain Graphical Model

Time Series Chain Graphical model (TSCGM) is a generalized version of the Gaussian graphical model in such a way that the GGM is repeated through time steps. To depict the model, it is better to explain the data as $Y = Y_1, ..., Y_T$ where each Y_j is in the form of $(n \times p)$-dimensional sub-matrix and T is the number of total time points of the data array as below. In this representation y_{ijt} denotes the ith sample of the jth protein at time t.

$$Y = \left[\begin{array}{cccc|cccc|cccc} y_{111} & y_{121} & \cdot & y_{1p1} & y_{112} & y_{122} & \cdot & y_{1p2} & \cdot & y_{11T} & y_{12T} & \cdot & y_{1pT} \\ y_{211} & y_{221} & \cdot & y_{2p1} & y_{212} & y_{222} & \cdot & y_{2p2} & \cdot & y_{21T} & y_{22T} & \cdot & y_{2pT} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ y_{n11} & y_{n21} & \cdot & y_{np1} & y_{n12} & y_{n22} & \cdot & y_{np2} & \cdot & y_{n1T} & y_{n2T} & \cdot & y_{npT} \end{array}\right]$$

According to the first-order Markov property, the joint probability density of $Y_1, ..., Y_T$ can be written as

$$f(Y_1, ..., Y_T) = f(Y_1)f(Y_2|Y_1) \times ... \times f(Y_T|Y_{T-1}) \tag{3.1}$$

By dividing both sides of Equation 3.1 by the initial term $f(Y_1)$, the conditional likelihood for TSCGM can be written as

$$f(Y_1, ..., Y_T|Y_1) = \prod_{t=2}^{T} f(Y_t|Y_{t-1}).$$

We assume that for $t = (1, 2, ..., T)$, $f(Y_t|Y_{t-1}) \sim N(\Gamma Y_{t-1} + \epsilon_t)$ where $\epsilon_t \sim N(0, \Theta^{-1})$. In the case of real data, the normality assumption of the conditional distribution can be provided but it should be noted that $Y_t|Y_{t-1}$s are not identically distributed because their mean vectors are not the same. Therefore, in order to test if $Y_t|Y_{t-1}$s are identically and independently distributed as the main assumption, it is enough to investigate whether the error terms ϵ_t are identically and independently distributed or not.

In the proposed graphical model in Section 2.1.2, we define only indirect edges between nodes such that every zero elements in the precision matrix Θ implies the conditional independence between the corresponding nodes. Here in the TSCGM, there are two types of edges, directed and undirected edge. The directed edges come from the autoregressive coefficient matrix shown by Γ which is a non-symmetric and non-diagonal matrix that represents the relationship between variables comes from time points (different lags). It means that a variable in time t is correlated with any variable even itself in time $(t + 1)$. The undirected edges are obtained through the precision matrix Θ in the TSCGM model similar to CGGM.

Therefore, the proposed model in this study connects the graphical models through time in the vector autoregressive model with lag (1), i.e., VAR(1).

TSCGM can be also defined by the state-space model as follows [1].

$$Z_t = AZ_{t-1} + BY_{t-1} + \omega_t \tag{3.2}$$

where

$$Y_t = CZ_t + DY_{t-1} + v_t \tag{3.3}$$

In Equations 3.2 and 3.3, Z_ts are unobserved hidden factors, ω_t and v_t denote independent noise terms for Z_t and Y_t, respectively.

Thereby, the above statements can be assembled under a single equation in the form of the VAR(1) model as below:

$$Y_t = (CB + D)Y_{t-1} + r_t, \tag{3.4}$$

where $r_t = v_t + C\omega_t + CAZ_{t-1}$. So, the matrix $CB + D$ plays the role of Γ matrix in the VAR(1) model. Therefore, every non-zero element in the Γ matrix corresponds to a directed edge between the related nodes in a way that

$$\begin{bmatrix} Y_{1,t+1} \\ Y_{2,t+1} \\ . \\ . \\ Y_{p,t+1} \end{bmatrix} = \Gamma \begin{bmatrix} Y_{1,t} \\ Y_{2,t} \\ . \\ . \\ Y_{p,t} \end{bmatrix}.$$

In the above equation, the matrix Γ is the correlation of nodes in consecutive times such that Γ_{ij} is the correlation of Y_i at time $t = (1, 2, ..., T - 1)$ and Y_j at time $t = (2, ..., T)$ and if it is far from zero, we can conclude that there is a directed edge from Y_i to Y_j.

So, there are two kinds of matrices to be estimated:

i) The symmetric precision matrix shows the undirected graph inside of each time step.

ii) The non-symmetric auto-regressive coefficient matrix which shows the directed graph between time steps.

In the estimation of the model parameters we proposed two main methods: i) Semi-Bayesian and ii) fully-Bayesian. In the semi-Bayesian approach, we suggest an algorithm based on two methods, RJMCMC and BDMCMC and in the fully-Bayesian approach, we consider an alternative way by defining an extra tuning hyper parameter in order to control the fluctuations to increase the accuracy of the fully-Bayesian approach.

3.2.1 Semi-Bayesian RJMCMC for the Estimation of Parameter Matrices of TSCGM

Once Γ is estimated by the Pearson correlation, the precision matrix can be obtained by RJMCMC by the following algorithm:

1. If the normality holds for data, we estimate the autoregressive coefficient matrix by the Pearson correlation and the precision matrix by RJMCMC or BDMCMC method based on $n \times T$ samples for p variables and call them Γ_1 and Θ_1, respectively.

2. We use Γ_1 and Θ_1 to simulate new data from the multivariate density (MVN) according to the method as explained below.

    - Initially, we sample a random variable Y_1 from MVN with a mean zero and a covariance matrix Θ^{-1}, i.e., MVN $(0, \Theta^{-1})$.
    - Then, we sample random variables Y_t from MVN with a mean $\Gamma Y_{(t-1)}$ and a covariance matrix Θ^{-1}, i.e., MVN $(\Gamma Y_{(t-1)}, \Theta^{-1})$ for $t = 2, \ldots, T$.

3. We estimate the autoregressive coefficient matrix by the Pearson correlation and the precision matrix by RJMCMC or BDMCMC method and call them Γ_2 and Θ_2, respectively.

4. We turn back to the second step until the convergence.

In this computation, if the normality assumption does not hold for data, Copula can make them normally distributed to be used in order to estimate the unknown parameters.

3.2.2 Full Bayesian approach by Reversible Jump Markov Chain Monte Carlo Method in TSCGM

As defined in Section 2.3, RJMCMC is applied to estimate the precision matrix in CGGM. In TSCGM, we deal with two kinds of matrices that RJMCMC can estimate both of them simultaneously. Finally, under the normality of data, the strictly positive precision parameters $\sigma_p = \sigma_g = 0.1$ is used in the prior and the posterior conditional distributions for the precision and an autoregressive coefficient matrix, respectively. The procedure is similar to both proposed full-Bayesian approaches. The difference is only fixed or changing σ_g to control the fluctuation in the burn-in period.

3.2.2.1 Fulyl Bayesian RJMCMC without a tuning parameter

Hereby, we take the following conditional posteriors in the calculation of the joint posterior density.

$$\begin{aligned}
\Gamma|(\Gamma_0, \sigma, \Theta^{-1}) &\sim \mathrm{N}(\Gamma_0, \frac{1}{\sigma}\Theta^{-1}), && (3.5)\\
(\Theta^{-1}|D) &\sim \mathrm{IW}(\Theta, D, \nu), && (3.6)\\
(\Gamma, \Theta^{-1}) &\sim \mathrm{NIW}(\Gamma_0, \sigma, D, \nu), && (3.7)\\
(\Gamma, \Theta^{-1})|Y &\sim \mathrm{NIW}(\Gamma_n, \sigma_n, D_n, \nu_n), && (3.8)
\end{aligned}$$

where IW and NIW denote the inverse G-Wishart and the joint distribution of the normal-inverse-G Wishart distribution, respectively. Furthermore,

$$\begin{aligned}
\Gamma_n &= \frac{\sigma\Gamma_n + n\bar{Y}}{\sigma + n}, && (3.9)\\
\sigma_n &= \sigma + n, && (3.10)\\
D_n &= D + nU + \frac{\sigma n}{\sigma + n}\text{tr}(Y - \Gamma_0), && (3.11)\\
\nu_n &= \nu + n. && (3.12)
\end{aligned}$$

in which tr(.) shows the trace of the given matrix. Moreover, $\sigma = 3$, $D = I_p$, $\nu = p$ and Γ_0 indicates a $(p \times p)$-dimensional zero matrix. For more information, the study of [8] provides more details about the natural prior and the best selection of the hyperparameters σ_p and σ_g. Then, RJMCMC repeats the following steps until the convergence of the parameter satisfies:

Resample the latent data

Here, Z is a $((n \times T) \times P)$-dimensional matrix where T denotes the time steps, n refers to the sample size for each node or variable, and p is the number of variables. In inference of the model parameters in each column, which is related to each node, we calculate its minimum L and its maximum U as the vectors of p elements based on the original data. In this step, by using the Θ matrix and L as well as U vectors, we generate another Z_{1i}i's from the truncated normal density in L_i and U_i distributions in a way that for $i = 1, 2, ..., p$, we have $Z_{1i}|Z_{l\setminus li} \sim N(\mu_i, \sigma_i^2)$, where $\mu_i = -\Sigma_{y \in bd(i)} \frac{\Theta_{i,y}}{\Theta_{i,i}} z_{y,j}$ for $bd(f) = y \in (1, \ldots, p) : (f, j) \in E$ when $E = (f, j)|\Theta_{f,y} \neq 0, f \neq y$ and $\sigma_i^2 = \frac{1}{\Theta_{i,i}}$. Then, for $t = 2, \ldots, T$, we simulate Z_{ti} from the truncated normal in the L_i and U_i from $Z_{ti}|Z_{t\setminus ti} \sim N(\Gamma Z_{(t-1)i} + \mu_i, \sigma_i^2)$. Then, in the second step, these z_{ijt} s $j = 1, 2, \ldots, n$ are used.

Resample the precision matrix

In this step, the Cholesky decomposition is applied for non-zero diagonal elements and the Metropolis- Hasting update of φ is done by sampling γ from a normal distribution truncated below at zero with a mean $\varphi_{i,i}$ and a variance σ_p^2. Later, γ is replaced to the related diagonal elements of φ and φ is transformed to φ' with a probability $\min\{R_p, 1\}$ where

$$R_p = \frac{\Phi(\frac{\phi_{i,i}}{\sigma_p})}{\Phi(\frac{\gamma}{\sigma_p})} (\frac{\gamma}{\phi_{i,i}})^{\omega+n+nb(i)-1} R'_p,$$

where

$$R'_p = \exp\left\{-\frac{1}{2}\text{tr}(\Theta' - \Theta)^T (D + \text{tr}(Z^T Z))\right\}.$$

Thus, Φ is the cumulative distribution function of the multivariate normal density. Here, tr$(.)$ and $(.)^T$ describe the trace and the transpose of the given term. For non-diagonal elements of φ, a new γ is sampled from $N(\varphi_i, \sigma_p^2)$. In these cases, φ is transformed to φ' with a probability $\min\{R'_p, 1\}$.

Resample the autoregressive coefficient matrix

In this step, all of the elements of the Γ matrix will be perturbed by the Metropolis-Hasting algorithm. In the calculation for the non-zero diagonal elements, the update of Γ is done by sampling a ω from a normal distribution truncated below at zero with a mean $\Theta_{i,i}$ and variance σ_g^2. Then, ω is replaced to the related diagonal elements of Γ and Γ is transformed to Γ' with a probability $\min\{R_p, 1\}$ in which

$$R_p = \frac{p(\Gamma' | z^{(1:n)}, G^s)}{p(\Gamma^s | z^{(1:n)}, G^s)} \times \frac{q(\Gamma_{i,i} | \omega)}{q(\omega | \Gamma_{i,i})} = \frac{\Phi(\Gamma_{i,i}/\sigma_g)}{\Phi(\omega/\sigma_g)} \left(\frac{\omega}{\Gamma_{i,i}}\right)^{\delta+n+nb(i)-1} R'_p.$$

Herein, $R'_p = \exp\left\{-\frac{1}{2}\text{tr}(\Gamma' - \Gamma)^T (D + \text{tr}(Z^T Z))\right\}$. On the other hand, in the update of the non-diagonal elements of Γ, a new ω' is sampled from $N(\Gamma_{i,j}, \sigma_g^2)$. In these cases, Γ is transformed to Γ' with a probability $\min\{R'_p, 1\}$.

Resample the undirected graph

In this step, only one element of the Cholesky matrix $\varphi_{i,j}$, which is obtained in the previous step, is selected randomly. If there is no edge between Y_i and Y_j, it will be changed by a value from $N(\varphi_{i,j}, \sigma_p^2)$ in φ with a probability $\min\{R_p, 1\}$ where

$$R_p = \sigma_p \sqrt{2\pi} \varphi_{i,i} \frac{I_G(\delta, D)}{I_{G'}(\delta, D)} \times \exp\left\{-\frac{1}{2}\text{tr}((\Theta' - \Theta)^T (D + \text{tr}(Z^T Z)) + \frac{(\varphi'_{i,j} - \varphi_{i,j})^2}{2\sigma_p^2}\right\}.$$

If there is an edge between Y_i and Y_j, it will be replaced by a zero in φ with a probability $\min\{1, R'_p\}$ where

$$R'_p = (\sigma_p \sqrt{2\pi} \varphi_{i,i})^{-1} \frac{I_G(\delta, D)}{I_{G'}(\delta, D)} \times \exp\left\{-\frac{1}{2}\text{tr}((\Theta' - \Theta)^T (D + \text{tr}(Z^T Z)) + \frac{(\varphi'_{i,j} - \varphi_{i,j})^2}{2\sigma_p^2}\right\}.$$

In these expressions, I_G implies the normalization constant of the G-Wishart distribution as implemented beforehand.

Resample the directed graph

Here, only one element of the Γ matrix is selected randomly. If there is no directed edge from Y_i to Y_j, it will be changed by a value from $N(0, \sigma_g^2)$ in Γ with a probability $\min\{1, R_p\}$ while

$$R_p = \sigma_g \sqrt{2\pi}\Gamma_{i,i} \times \exp\left\{-\frac{1}{2}\text{tr}((\Gamma' - \Gamma)^T(D + \text{tr}(Z^T Z)) + \frac{(\Gamma'_{i,j})^2}{2\sigma_g^2}\right\}.$$

If there is an edge between Y_i and Y_j, it will be replaced by a zero in φ with a probability $\min\{1, R'_p\}$ where

$$R'_p = (\sigma_g \sqrt{2\pi}\Gamma_{i,i})^{-1} \times \exp\left\{-\frac{1}{2}\text{tr}((\Gamma' - \Gamma)^T(D + \text{tr}(Z^T Z)) + \frac{(\Gamma_{i,j})^2}{2\sigma_g^2}\right\}.$$

3.2.2.2 Modified fully-Bayesian RJMCMC with an Adaptable Tuning Parameter

From the results of the Γ estimation, it is seen that a fluctuated value can be included in the generation of the proposed value in RJMCMC since a novel proposal can be almost around the first simulated value otherwise. The parameter in the modified RJMCMC which controls the number of allowed changes in every step for the estimation of Γ matrix is σ_g. So, we suggest an adaptive algorithm to control the amount of change in every 100-iteration in the burn-in-period of the inference whose steps are listed below.

- If the sum of non-coincided values of two matrices in Γ_{i+1}th and Γ_{i+100}th for $i = 0, 100, 200, \ldots$ until the end of burn-in-period is more than 0.6, the new value of σ_g will be updated as $\sigma_g \times 1.1$. By applying this, we expect more fluctuation in the burn-in time but more accurate $\tilde{\Gamma}$.
- If the acceptance probability of the proposal between Γ_{i+1}th and Γ_{i+100}th is less than 0.05, σ_g is taken as $\sigma_g/1.1$ so that we can search the target posterior density with a small posterior density under small fluctuations.

- Finally, if the acceptance probability of the proposal of Γ_{i+1}th and Γ_{i+100}th is around 0.30, we do not change σ_g during the burn-in period.

In the calculation, we select 0.3 and the condition of the optimal mixing probability since an acceptance probability for a single parameter is suggested around 0.24 a good mixing [12]) which is close to the best value for the tightness hyperparameter, σ_g in the study of [8] . On the other hand, for a large number of parameters, this probability can be taken very low like 0.05 and mixing of it can be chosen around 0.6 as most as the mixing of the proposal distribution and the convergence of the model can be difficult [28].

3.3 Application

This section has three parts which indicate the accuracy of two suggested methods (one semi-Bayesian and two Fully-Bayesian approaches) in Section 3.2. The accuracy of the first method is computed through measures, namely,

Accuracy = (TP+TN)/N
Sensitivity = TP/(TP+FN)

The meaning of TP, FP, TN, and FN are true positive, false positive, false positive and false negative edges in the graphical model which were introduced previously.
In the second scenario, we compare the proposed modified RJMCMC method with its alternative penalized likelihood method by Abegaz and Wit (2013) [1] by using only the specificity measure (Specificity=TN/(TN+FP)) since the available comparison in the study of Abegaz and Wit (2015) [2] applies this measure.

3.3.1 Accuracy of Semi-Bayesian RJMCMC for a Simulated Dataset

The accuracy of the estimated Γ and Θ by BDMCMC (left) and RJMCMC (right) under different dimensional systems (p), number of observations (n) and number of time points (T) is shown in Table 3.1. In this comparison, Θ indicates the ultimate estimated precision matrix of the system while Γ shows the correlation structure between time points, Γ is more important than Θ for us to correctly infer the biological system. Here, from the outputs, it is seen that the semi-Bayesian approach gives tolerable results under small n, p and T, but the performance of the estimates improves significantly when n, p or T increases. On the other side, the accuracy of the estimated Θ is high particularly to catch the true negative elements via BDMCMC and with a negligible difference via RJMCMC. But in general, for both ways, we obtain promising estimated outputs for such complex systems having a large number of parameters as seen in Table 3.1.

Table3.1: Results of accuracy (Acc) and sensitivity (Sens) measures for the simulated dataset under different sample size (n), number of genes (p) and number of time points (T).

			BDMCMC				RJMCMC			
			Θ		Γ		Θ		Γ	
n	p	T	Acc	Sens	Acc	Sens	Acc	Sens	Acc	Sens
10	10	5	0.42	0.80	0.86	0.00	0.64	0.76	0.18	0.83
20	10	5	0.57	1.00	0.82	0.00	0.47	0.58	0.67	0.00
50	10	5	0.75	0.00	0.80	0.00	0.56	0.39	0.44	1.00
20	20	5	0.31	0.40	0.89	0.00	0.67	1.00	0.34	0.88
50	20	5	0.80	0.10	0.81	0.00	0.48	0.43	0.41	0.79
100	50	5	0.95	0.00	0.90	0.00	0.59	0.30	0.40	0.76
10	10	10	0.42	0.50	0.93	0.00	0.64	0.74	0.33	1.00
20	10	10	0.75	0.00	0.91	0.00	0.53	0.58	0.22	1.00
50	10	10	0.77	0.00	0.84	0.50	0.63	1.00	0.49	0.80
20	20	10	0.76	0.30	0.90	0.00	0.50	0.49	0.33	0.89
50	20	10	0.91	0.28	0.90	0.20	0.53	1.00	0.59	0.40
100	50	10	0.97	0.10	0.90	0.20	0.77	0.14	0.47	0.79
10	10	50	0.88	0.30	0.95	0.67	0.53	0.29	0.67	0.90
20	10	50	0.89	0.00	0.93	0.67	0.73	0.00	0.42	1.00
50	10	50	0.97	1.00	0.86	0.00	0.38	0.92	0.83	0.07
20	20	50	0.79	0.10	0.84	0.46	0.83	0.07	0.38	1.00
50	20	50	0.79	0.00	0.84	0.63	0.96	0.00	0.50	1.00
100	50	50	0.99	0.00	0.91	0.16	0.99	0.00	0.40	0.92

3.3.2 Accuracy of Fully-Bayesian RJMCMC for a Simulated Dataset

We compare our proposed method with its alternative, penalized likelihood model which is introduced by Abegaz and Wit (2013) [1]. The results are shown in Table 3.2.

From the tabulated outcomes, it is observed that the performance of both methods is

Table3.2: Results of specificity for the fully-Bayesian method with a likelihood based method under different sample sizes (n), number of genes (p) and number of time points (T).

			Fully-Bayesian RJMCMC		Likelihood-based	
p	n	T	Θ	Γ	Θ	Γ
10	10	5	0.87	0.82	0.85	0.97
		15	1.00	0.84	0.91	1.00
	50	5	0.92	0.99	0.94	0.99
		15	0.95	1.00	0.99	1.00
50	10	5	1.00	0.85	0.96	0.98
		15	1.00	0.85	0.98	0.99
	50	5	0.54	0.99	0.99	0.99
		15	0.88	0.99	0.99	0.99
	100	5	0.77	0.99	0.99	0.99
		15	0.98	0.98	0.99	1.00

good especially for the precision matrix Θ. The likelihood-based method is better in the estimation of the autoregressive coefficient matrix especially for low n. But the performance of both methods becomes closer while p and T increase. In general, both approaches are acceptable and the likelihood-based method is better under certain conditions.

3.3.3 Accuracy of Fully-Bayesian RJMCMC with Adaptive Tuning Parameter for a Simulated Dataset

In the implementation of the adaptive tuning parameter run with 100000 MCMC runs and take the first 20000 iterations as the burn-in-period in order to calibrate this value. Then, we compute the accuracy and the sensitivity measures as listed in Table 3.3. Later, we also compute MCC accuracy measure whose expressions were presented in Equations 2.22 in order to check the gain in accuracy via the proposal calibration. The results are represented in Table 3.3. In Table 3.3, we present the outcomes without tuning strategy in RJMCMC via the index A, which corresponds to the proposed RJMCMC plan in Section 3.2.2. Accordingly, we indicate the outputs of new findings under adaptive RJMCMC via an index B. Finally, the specificity measure of the likelihood-based approach of Abegaz and Wit (2013) [1] is reported via an index L. The findings show that adaptive RJMCMC is almost as accurate as of the likelihood-based method.

Table3.3: Results of Matthew correlation coefficient (MCC) of the modified RJMCMC method with adaptable tuning parameter (MCC$_A$), without tuning parameter (MCC$_B$) and also the likelihood based method (MCC$_L$).

			Γ			Θ		
p	n	T	MCC$_A$	MCC$_B$	MCC$_L$	MCC$_A$	MCC$_B$	MCC$_L$
10	10	5	0.41	0.09	0.56	0.51	0.78	0.60
		15	0.60	0.05	0.82	0.57	0.70	0.74
	50	5	0.64	0.00	0.83	0.66	0.58	0.76
		15	0.62	0.00	0.92	0.68	0.11	0.94
50	10	5	0.51	0.01	0.45	0.65	1.00	0.54
		15	0.52	0.01	0.71	0.68	0.78	0.75
	50	5	0.62	0.02	0.82	0.70	0.21	0.85
		15	0.63	0.00	0.91	0.73	0.55	0.95
	100	5	0.64	0.01	0.92	0.94	0.41	0.92
		15	0.66	0.00	0.94	0.97	0.80	0.92

From Table 3.3 and Table 3.2, it is seen that under all scenarios, adaptive RJMCMC has better accuracy than RJMCMC without calibration in the tuning parameter. This result shows that the new RJMCMC strategy can be a promising alternative to estimate TSCGM via fully Bayesian approaches.

3.4 Conclusion

In the case of some accuracy measures like specificity, our proposed method could be compared with its alternative likelihood-based method which was done by Abegaz and Wit (2013) [1]. In the first part of our study, we compared the first proposed method and compared RJMCMC and BDMCMC. However, RJMCMC has better accuracy measures than BDMCMC in terms of accuracy and sensitivity measures. In the second part, we compared the RJMCMC based method with the likelihood-based study done by Abegaz and Wit (2013) [1] by their specificity measures. As a result, the fully-Bayesian method could perform somehow as same as the likelihood-based method in the estimation of Θ in all scenarios while its performance for small values of n and T is worse than the likelihood-based method. In the last part, we compared the RJMCMC based method with and without tuning parameter with their powerful alternatives as well as the likelihood-based method. Here, we observed a significant improvement in the selected accuracy measure, i.e., Matthew Correlation Coefficient.

Furthermore, we found that the estimation of the precision matrix Θ is more accurate than the autoregressive coefficient matrix in all scenarios about RJMCMC. In the likelihood-based method, the estimation of Γ is slightly more accurate than Θ for small n, p and Ts. Finally, for all scenarios, all accuracy measures increase as the sample size and the number of time points increases. As a conclusion, we consider that the RJMCMC based approach with the tuning parameter can be a good alternative to the likelihood-based method with a comparable accuracy.

YÖK

CHAPTER 4

COPULA GRAPHICAL MODELS IN THE CONSTRUCTION OF BIOLOGICAL NETWORKS

4.1 Introduction

In CGGM which was explained in Chapter 2, the captured dependence structure is the undirected edge between the nodes and the used copula is the Gaussian copula because of its exclusive property that the uncorrelateness implies the independence. But in some cases, Gaussian may not be an appropriate model between the marginals of the variables since it requires the symmetry and the tail dependency is zero. So another copula could be more appropriate for modeling this type of datasets. Thereby, in this chapter, we aim to use the vine copulas which enables us the flexibility to select the non-Gaussian copula in the construction of protein-protein interaction networks' models. Accordingly, in the first section, the Copula aspect is introduced in terms of its foundations, types and properties of Copula pair families. Then, data analysis and model building worked by Copula are explained briefly. Finally, in the Application and Conclusion parts, we compare the mentioned methodologies by some accuracy measures such as F_1 –score and Matthews correlation coefficient and summarize our results, respectively.

4.2 Copula

The Sklar's theorem is the most fundamental theorem which constitutes the important role of copulas for describing dependence in statistics. It establishes the link between multivariate distribution functions and their univariate margins. Let F be the d-dimensional distribution function of the random vector $Y = (Y_1, Y_2, .., Y_d)^T$ with margins $F_1, F_2, .., F_d$. Then there exists a copula C such that for all $y = (y_1, y_2, .., y_d) \in (-\infty, +\infty)^d$, $F(y) = C(F_1(y_1), .., F_d(y_d)$. C is unique if $F_1, F_2, .., F_d$ are continuous.

To show the pair copula construction model, we show it for $d = 3$. The data set has three variables Y_1, Y_2, Y_3. Their joint distribution function is $f(y_1, y_2, y_3) = f_1(y_1)f(y_2|y_1)f(y_3|y_2, y_1)$

$$f(y_2|y_1) = \frac{f(y_1, y_2)}{f_1(y_1)} = \frac{c_{1,2}(F(y_2)F(y_1))f_1(y_1)f_2(y_2)}{f_1(y_1)} = c_{1,2}(F(y_2)F(y_1))f_2(y_2)$$

Accordingly $f(y_3|y_2, y_1) = c_{2,3|1}(F(y_2|y_1), F(y_3|y_1))c_{1,3}(F_1(y_1), F_3(y_3))f_3(y_3)$

By combining the past two equations, we have:

$$\begin{aligned} f(y_1, y_2, y_3) = f_1(y_1)f_2(y_2)f_3(y_3)c_{1,2}(F(y_2)F(y_1)) \\ c_{1,3}(F_1(y_1), F_3(y_3))c_{2,3|1}(F(y_2|y_1), F(y_3|y_1)) \end{aligned} \tag{4.1}$$

Therefore, the whole structure is shown in terms of pair copulas in a way that each of them can be investigated independently. It means in Equation 4.1, instead of working with three variables Y_1, Y_2 and Y_3 at the same time (shown as 1-2-3), we work with 1-2, 1-3 and 2-3|1 in one scenario, 1-2, 2-3 and 1-3|2 in another scenario and also changing the order of variables which can be written as a model itself. So, there are many ways ($\frac{p(p-1)}{2}$) to write a multivariate structure in terms of pair copula. So, one of the most challenging issues is to choose the best model among all available models.

In the case of analysis by the pair copula, we have three phases: a regular vine, a canonical vine and a drawable vine shown briefly as R, C, and D vine. Each has a different structure. R-Vine is a general form of vine copula which can be written as a combination of the other two models. Their structures are different but both of them deals with pair-copulas. It depends on the way the joint distribution function is written. A good way to show their structure and difference is their graphical representation in Figure 4.1. It is for a data set with 4 variables.

In the C-vine, the first edge (variable) is selected in the first tree, the second root is selected in the second tree and so on. So for d-dimensional data, we should choose $d - 1$ roots for every tree. The order of variables in the left side of Figure 4.1 is

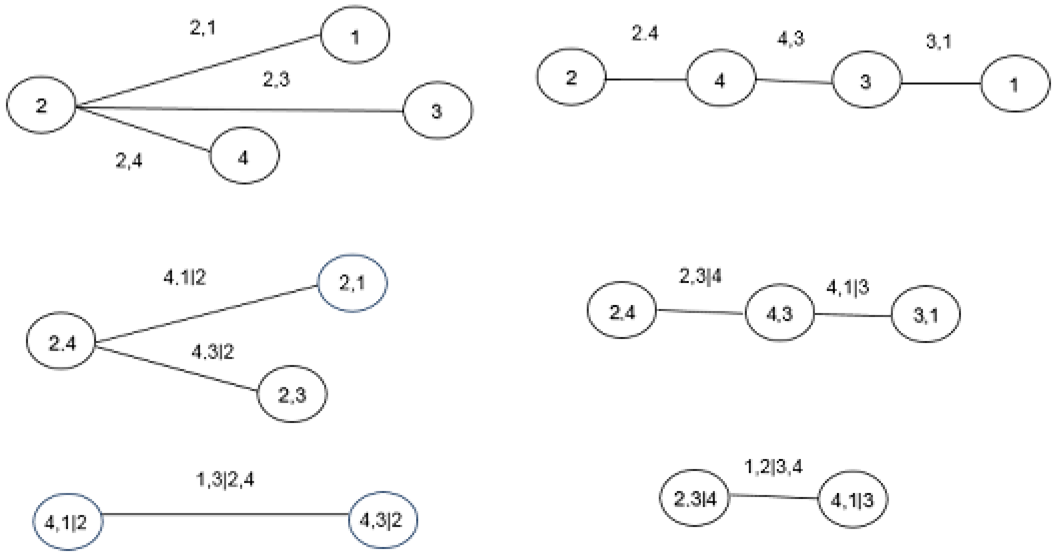


Figure 4.1: Examples of a four-dimensional C-Vine (left panel with order $\{2, 4, 1, 3\}$) and a D-vine structure (with order$\{2, 4, 3, 1\}$) in the right panel)

$\{2, 4, 1, 3\}$.
In the D-vine copulas, the nodes of the first tree determine the whole model. And finally, both vine copulas look like a vine and they include bivariate cases, only. The order of variables in the right side of figure 4.1 is $\{2, 4, 3, 1\}$.
In each ith tree $i = 1, 2, d - 1$ there are $d - i + 1$ nodes (links) and $d - i$ edges. Each edge can be represented by an appropriate copula. The empirical copula which is used in data analysis with the copula approach is defined as

$$C_n(u) = \frac{1}{n} \sum_{i=1}^{n} 1(\hat{U}_i < u), u \in [0, 1]^d$$

where $\hat{U}_i = R_i/(n+1)$ in which $R_i = (R_{i1}, ..., R_{id})$ and R_{ij} is the rank of Y_{ij} among $Y_{1j}, ..., Y_{nj}$ for $i = 1, 2, ..., n$ and $j = 1, 2, ..., d$. this transformation makes the data in the interval of $(0, 1)$ to be used as a copula data.

4.2.1 Types of Copula families

Once the structure of vine copula is determined, the connection between variable (nodes) needs to be defined by a pair of the copula family. The pair- copula can be from two main categories: Elliptical and Archimedean copula. Of course, there are other copula families, However, in this study we will not cover them.

The Elliptical copulas are in the form of $C(u_1, u_2) = F(F_1^{-1}(u_1), F_2^{-1}(u_2))$ where u_is are from the uniform $(0, 1)$ distribution. The most famous elliptical copulas are the bivariate Gaussian (one parameter) and the bivariate student's t copula (two parameters).

Table4.1: The denotation and properties of the bivariate Elliptical families

	Distribution	Parameter range	Kendall's τ	Tail dependence
1	Gaussian	$\rho \in (-1, 1)$	$\frac{2}{\pi}\arcsin(\rho)$	0
2	Student's t	$\rho \in (-1, 1), \nu > 2$	$\frac{2}{\pi}\arcsin(\rho)$	$2\mathrm{t}_{\nu+1}(-\sqrt{\nu+1}\sqrt{\frac{1-\rho}{1+\rho}})$

In Table 4.1 the properties of these two copulas are written. In this family, there is a one-to-one corresponding between the correlation parameter ρ and the well-known non-parametric correlation measure τ. So, the model parameters can be estimated by Kendall's τ and vise versa. In t-copula, there is the tail dependence, too. When two variables Y_1 and Y_2 are tail-dependent, means in the very small or very large values of Y_1, there is some expectancy to face very small or very large values for Y_2 and vice versa. One of the facts that achieved by copula is to realize the tail dependence, which is not negligible especially in the data sets sensitive to the extreme values.

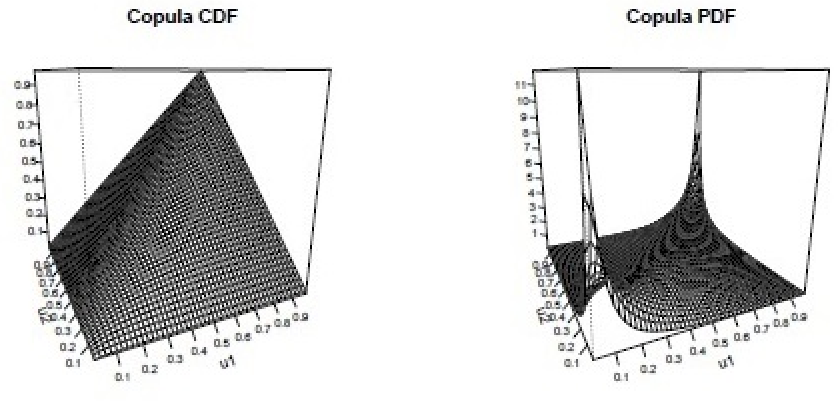
The tail dependence can be seen clearly in Figure 4.2 which is in terms of the copula parameters, the correlation parameter ρ and the degrees of freedom ν. As ν increases, the t-copula converges to the Gaussian copula and the tail dependence value converges to zero.

In Figure 4.2, you can see an example of a copula CDF and PDF of a bivariate student's t copula with a dependence parameter $\rho = 0.7$ and $df = 4$. Figure 4.3 shows PDF and scatter plot of simulated Gaussian copula for different values of τ in order to emphasize the effect of the correlation in the elliptical copula family structure.

On the other side, the archimedean copulas are in the form of

$$C(u_1, u_2) = \psi^{[-1]}(\psi_1(u_1) + \psi_2(u_2)) \tag{4.2}$$

Figure 4.2: CDF and PDF of a bivariate student's t copula



where ψ is the generator function and $\psi^{[-1]}$ is the inverse function of positive values. We present the generator function in Table 4.2. To clear the relationship between the generator function and the bivariate copula, for instance, in the Clayton family, the generator function is taken as $\psi(t)=\frac{1}{\theta}(t^{\theta}-1)$. So the inverse of the generator function can be written as $\psi^{-1}(t)=(t\theta+1)^{-\frac{1}{\theta}}$. Herein, the pseudo-inverse is equal to the inverse when the inverse is not negative which is equivalent to the rectifier function. In order to get the copula formula, the bivariate Clayton family uses Equation 4.2 and puts u_1 and u_2 in it, i.e, $C(u_1,u_2)=\psi^{[-1]}(\frac{1}{\theta}(u_1^{-\theta}-1))+\frac{1}{\theta}(u_2^{-\theta}-1))$ which is equal to $max((u_1^{-\theta}+u_2^{-\theta}-1),0)^{-\frac{1}{\theta}}$. Therefore, we see that the generator function is simpler to write especially for the archimedean families with more than one parameter such as BB1, BB6, BB7 and BB8 which stand for the Clayton-Gumbel, the Joe Gumbel, the Joe-Clayton and the Joe-Frank copulas, respectively. These families with more than one parameter (two-parameter families) are made by a combination of one parameter archimedean families to provide a more flexible structure like covering one and two-sided tail dependence and also most of them are appropriate for a non-symmetric joint distribution. Table 4.2 shows the generator function and also some other properties of the members of the archimedean copula families.

As an example of the archimedean copula family, Figure 4.4 represents some one-parameter bivariate archimedean copula families with the same Kendall's τ.
All of the families can rotate regarding the best fit for the data set according to the

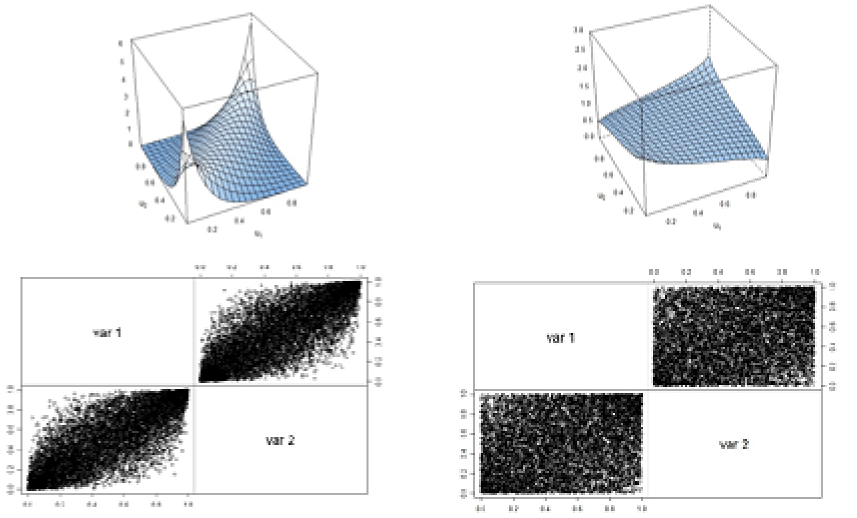


Figure 4.3: PDF and the scatter plot of the simulated Gaussian copula for $\rho = 0.8$ (left) and $\rho = 0.2$ (right)

Table4.2: the notation and the properties of the bivariate archimedean families

#	Name	Generator function	Parameter range	Tail dependence
3	Clayton	$\frac{1}{\theta}(t^{-\theta} - 1)$	$\theta > 0$	$(2^{-\frac{1}{\theta}}, 0)$
4	Gumbel	$-(\log t)^{\theta}$	$\theta \geq 1$	$(0, 2 - 2^{-\frac{1}{\theta}})$
5	Frank	$-\log(\frac{e^{-\theta t}-1}{e^{-\theta}-1})$	$\theta \in R$	(0,0)
6	Joe	$-\log(1 - (1-t)^{\theta})$	$\theta > 1$	$(0, 2 - 2^{-\frac{1}{\theta}})$
7	BB1	$(t^{-\theta} - 1)^{\sigma}$	$\theta > 0, \sigma \geq 1$	$(2^{-\frac{1}{\theta\sigma}}, 2 - 2^{\frac{1}{\sigma}})$
8	BB6	$(-\log(1 - (1-t)^{\theta}))^{\sigma}$	$\theta > 0, \sigma \geq 1$	$(0, 2 - 2^{-\frac{1}{\theta\sigma}})$
9	BB7	$(1 - (1-t)^{\theta})^{-\sigma} - 1$	$\theta \geq 1, \sigma > 0$	$(2^{-\frac{1}{\sigma}}, 2 - 2^{\frac{1}{\theta}})$
10	BB8	$-\log(\frac{1-(1-\sigma\theta)^{\theta}}{1-(1-\sigma)^{\theta}})$	$\theta \geq 1, \sigma \in (0,1)$	$(0, 0)$

following way:

$$\begin{aligned} C_{90}(u_1, u_2) &= u_2 - C(1 - u_1, u_2), \\ C_{180}(u_1, u_2) &= u_1 + u_2 - 1 + C(1 - u_1, 1 - u_2), \\ C_{270}(u_1, u_2) &= u_1 - C(u_1, 1 - u_2), \end{aligned}$$

where C_i represents the i degrees rotated version of the copula for $i = 90, 180, 270$. Figure 4.5 indicates an example of the rotation for the bivariate Clayton copula with Kendall's $\tau = 0.5$.

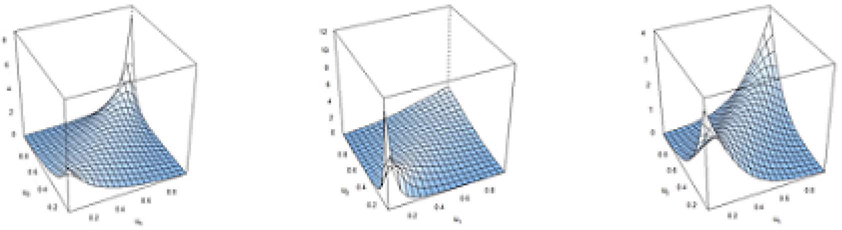
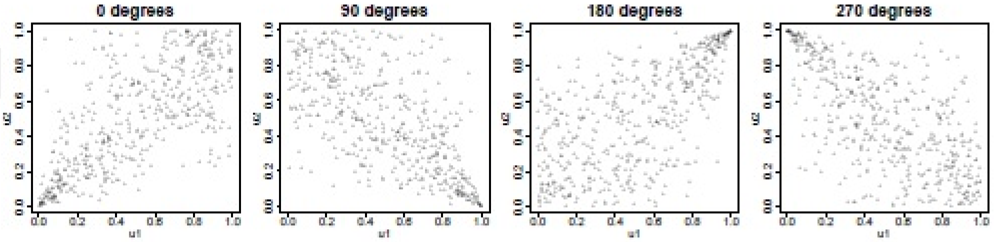


Figure 4.4: Gumbel, Clayton and Frank copulas, respectively, from left to right with parameters corresponding to Kendall's τ values of 0.5.

Figure 4.5: Samples from the Clayton copulas rotated with Kendall's $\tau = 0.5$

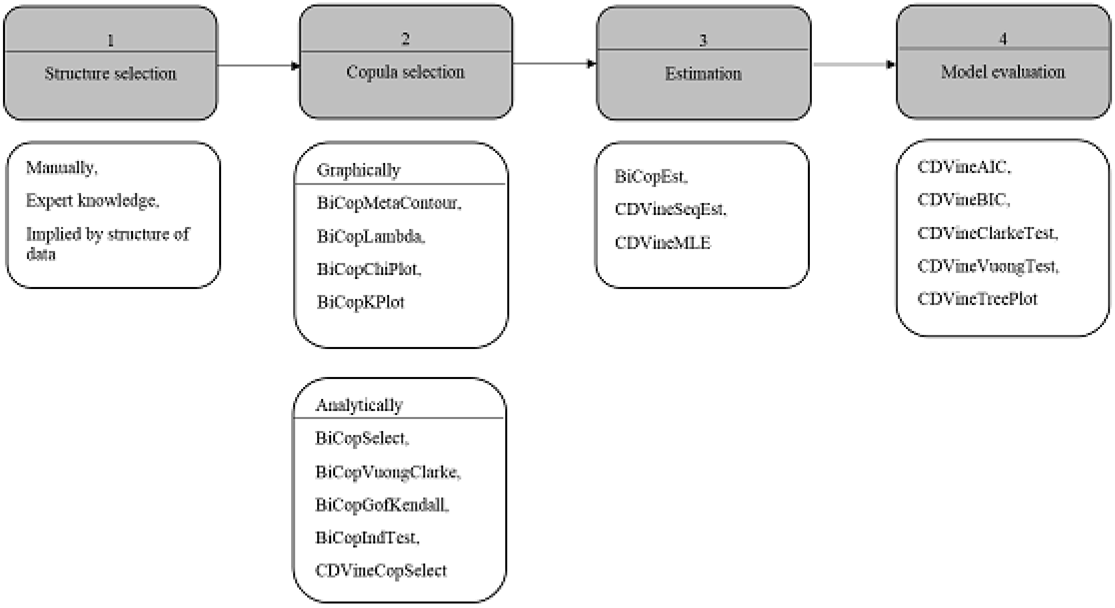


4.3 Vine Copula in Inference of Complex Data

As the model dimensionality increases, the model becomes more complicated and in most cases, the relationship between every two variables is not the same for all of the variables that means all of the edges cannot be modeled by a specific copula family. In Equation 4.1 which is represented as the simplest form of the joint distribution for a three cases function, we saw that it could be decomposed to some pair copula terms in a way that some of the conditioned and the rest not conditioned without any specific assumption or elimination.

There are several ways to construct the best model. As mentioned earlier, a regular vine is a general form of the vine copula that can be a C-vine in the tree and D-vine in another tree or even their combinations. So in every data set, there are $\frac{p(p-1)}{2}$ edges that should be determined with a pair copula from different families and from any rotation for each pair copula. There are some tests that determine the best-fitted pair copula for every two variables (given some others) as well as some goodness of fit tests. There are also some tests that compare two models based on the Maximum-likelihood values [5]. The diagram in Figure 4.6 shows the process of the analysis by considering the available suitable functions in **R**. There are several functions in the "CD-Vine" or its last version "VineCopula" package in **R** to be used in analyzing the data by vine copula but the function which is written in the diagram, is the most important ones. In Table 4.3, their action is indicated very briefly.

Figure 4.6: Proposed data analysis and model building work by some functions in package **VineCopula** in **R**



On the other hand, the order of variables determines the root of each tree in the C-vine and the path in the first tree in the D-vine. The algorithm of the order selection is briefly described as follows:

- Compute the empirical distribution function of the data to transform them into the uniformly distributed data.
- Compute the Kendall's τ correlation coefficient of the new data and select the variable with the largest τ as the first root.
- Select the best copula for each node between the first root and other variables and then, estimate the parameter(s). (The model parameters can be estimated by the MLE method for one or two-parameter families and by Kendall's τ in

Table4.3: The most important functions' names and descriptions according to the VineCopula package in R [5].

	Name	Explanation
1	BiCopMetaContour	Contour Plot of Bivariate Meta Distribution.
2	BiCopLambda	Lambda-Function (Plot) for Bivariate Copula Data.
3	BiCopChiPlot	Chi-plot for Bivariate Copula Data.
4	BiCopKPlot	Kendall's Plot for Bivariate Copula Data.
5	BiCopSelect	Selection and MLE of Bivariate Copula Families.
6	BiCopVuongClarke	Scoring GOF Test based on Vuong And Clarke Tests.
7	BiCopGofTest	Goodness-of-Fit Test for Bivariate Copulas.
8	BiCopIndTest	Independence Test for Bivariate Copula Data.
9	RVineClarkeTest	Clarke Test Comparing Two R-Vine Copula Models.
10	BiCopEst	Parameter Estimation for Bivariate Copula Data.
11	RVineSeqEst	Sequential Estimation of an R-Vine Copula Model.
12	RVineMLE	Maximum Likelihood Estimation of an R-Vine Copula Model.
13	RVineAIC	AIC and BIC of an R-Vine Copula Model.
14	RVineClarkeTest	Clarke Test Comparing Two R-Vine Copula Models.
15	RVineVuongTest	Vuong Test Comparing Two R-Vine Copula Models.
16	RVineTreePlot	Visualization of R-Vine Tree Structure.

only one-parameter copula families.)

- Transform the data by conditioning to the first selected variable via the h function by using the parameters estimated from the previous step as

  $$h(x|\nu,\theta) := \frac{\partial C_{x\nu_j(F(x|\nu_{-j}),F(\nu_j|\nu_{-j})|\theta)}}{\partial F(\nu_j|\nu_{-j})}$$

  where ν is the estimated parameter(s) in the previous step and ν_j refers to an arbitrary component.

- Select the variable among the new data which has the largest Kendall's τ as the second root.

- Continue the process until the $(d-1)$th root is found.

The algorithm is somehow similar to the forward selection in the multiple regression. To estimate the model parameters by the maximum likelihood estimation (MLE), the order and the copula families should be determined before. There are some methods to select the best pair copula between the nodes such as graphical tools like the contour plot and some other statistical tests like the Voung and Clarke or the goodness of fit test. In Voung and Clarke, the test compares all available choices two by two and gets a score for each family if it was better than its alternative. The family with the maximum score is chosen as the best one and then another goodness of fit test based on a χ^2 statistics determines if the proposed family is well fitted or not by giving a p-value. Similarly, there are some other tools to compare two models by using the AIC and BIC criteria or the Vungo test as well. All of the details for the mentioned tests are available in [5].

4.4 Application

In this chapter, we use five bench-mark data sets. The "CellSignal" data which was introduced in Section 2.5 and the second data which consist of ten proteins in 285 samples as used in [29] are analyzed by the C-Vine approach. The remaining data set are analyzed through R-Vine method which are the "Rochdale" data which was used in Section 2.5, too, and two additional real data sets gained from STRING database with 23 and 50 genes in 53 samples, respectively. In our analysis, we compare the accuracy between RJMCMC and vine-copula methods for each data sets and in the comparison, we use the F_1-score and MCC values whose expressions were firstly represented in Chapter 2.

4.4.1 The "CellSignal" data

We used this data set to see how our algorithm works for C-Vine. The output presented in Table 4.4 is the estimated matrix of the families obtained by the "VineCopula" package [5] with the **R** programming language.

By comparing Table 4.4 with its true graph in the study of [30], we obtain F_1-score= 0.45 and MCC=0.14. In this table, the values are related to a pair of copula and all of the zero's in the upper triangle of the adjacency matrix imply the (conditional) independence between associated genes. On the other side, the results applied by RJMCMC under the Gaussian copula have F_1-score= 0.63 and MCC=0.09. Hence, it is seen that F_1-score decreases under the vine copula with the MLE method in inference, whereas, MCC improves. Additionally, since the inference can be conducted under MLE, it can decrease the computational demand in the estimation regarding

Table4.4: The upper triangular of the estimated adjacency matrix of the CellSignal data with the copula families via numbers in the "VineCopula" package.

Name	AKT	PKC	PIP2	Pmek	Pjnk	PIP3	Plcy	PkA	Praf	P38	Erk
AKT	0	10	10	10	10	5	10	2	10	10	17
PKC	0	0	2	16	10	13	1	2	40	2	9
PIP2	0	0	0	1	1	9	10	30	1	13	9
pmek	0	0	0	0	9	13	40	29	40	2	20
Pjnk	0	0	0	0	0	30	6	30	7	5	20
PIP3	0	0	0	0	0	0	0	26	0	0	0
Plcy	0	0	0	0	0	0	0	0	0	0	0
PkA	0	0	0	0	0	0	0	0	0	0	0
praf	0	0	0	0	0	0	0	0	0	0	0
p38	0	0	0	0	0	0	0	0	0	0	0
Erk	0	0	0	0	0	0	0	0	0	0	0

RJMCMC. Furthermore, the values in Table 4.4 came from the codes defined in the "VineCopula" package in **R** as the following list.

0=independence			
1=Gaussian			
2=Student's t			
3=Clayton	13=180° rotated Clayton	23=90° rotated Clayton	33=270° rotated Clayton
4=Gumbel	14=180° rotated Gumbel	24=90° rotated Gumbel	34=270° rotated Gumbel
5=Frank			
6=Joe	16 =180° rotated Joe	26=90° rotated Joe	36=270° rotated Joe
7=BB1	17 =180° rotated BB1	27=90° rotated BB1	37=270° rotated BB1
8=BB6	18 =180° rotated BB6	28=90° rotated BB6	38=270° rotated BB6
9=BB7	19 =180° rotated BB7	29=90° rotated BB7	39=270° rotated BB7
10=BB8	20 = 180° rotated BB8	30=90° rotated BB8	40=270° rotated BB8

4.4.2 Data 2

The second data which we used to see the performance of the Vine copula, is a gynecological cancer network whose observations are assembled from the ArrayExpress database. In these kinds of data sets, we need to have the true graph which is obtained from the biological literature. This data set includes ten proteins, named as MAP2K1, PDIA3, MAPK1, IMP3, ERBB2, TFAM, MBD3, CHD4, CTNNB1 and CEBPB with order $\{4, 6, 5, 7, 8, 2, 10, 9, 1, 3\}$ [29]. These genes are selected as the core genes in the literature of gynaecological cancer and the quasi true network structure of these genes is represented by a complete graph meaning that all the entries of the adjacency matrix are composed of ones [3]. Thereby, in this study, we present the C-vine copula model for the data in Table 2 in order to estimate the true complete graph. Finally, in the construction of the network, we apply the “VineCopula” package in R. As seen in Table 2, the graph related to this network is a full graph. By comparing it with the related true network, we find F_1-score=1 indicating a higher accuracy via the C-Vine copula for Data 2. Whereas, the F_1-score is computed as 0.94 with RJMCMC. On the other side, MCC cannot be computed for this data set as both TP and FN are observed as zero. This result indicates a better accuracy under the C-Vine copula model. As seen in Table 4.5, the graph related to this network is a full graph. By comparing it with the related true network, we find F_1-score=1 presenting a higher accuracy via the C-Vine copula for Data 2. Whereas, the F_1-score is computed as 0.94 with RJMCMC. On the other hand, MCC cannot be computed for this dataset as both TP and FN are observed as zero.

Table4.5: The upper triangular of the estimated adjacency matrix of Data 2 with the copula families via numbers in the "VineCopula" package

Name	mp2k	pdia	mpk1	imp	erb2	tfm	mbd3	chd4	ctnb1	cbpb
mp2k	0	5	1	5	14	5	5	5	5	14
pdia	0	0	3	13	1	1	19	5	2	1
mpk1	0	0	0	40	5	3	23	10	30	23
imp	0	0	0	0	23	10	1	5	13	1
erb2	0	0	0	0	0	3	1	5	1	2
tfm	0	0	0	0	0	0	14	5	1	2
mbd3	0	0	0	0	0	0	0	33	33	5
chd4	0	0	0	0	0	0	0	0	1	5
ctnb1	0	0	0	0	0	0	0	0	0	5
cbpb	0	0	0	0	0	0	0	0	0	0

4.4.3 The "Rochdale" data

As we analyzed this data set in Section 2.5, as a reminder, this data set consists of eight binary variables for 665 women to investigate the relationship between the effective factor on women economical activities. To be used in copula methodology, we transformed the data to the Gaussian data through a method suggested by Hoff (2007) [16] and the R-Vine method was applied to this latent data to see the relationship the variables named from a to h. According to our proposed method, R-Vine, the significant non-zero edges are estimated as ef, dg, cg, cf, ce, bh, be, bd, ag, ae, ad, cd meaning that 10 of 13 relationships are estimated correctly and cd is the overestimated edge. By these outputs, we computed TP=10, FP=1, FN=3 and TN=14. Accordingly, F_1-score= 0.83 and MCC=0.72 while in the work done by Purutcuoglu and Farnoudkia (2017) [27], these values are found as F_1-score= 0.93 and MCC=0.86, respectively.

4.4.4 23 gene

The data came from a categorized and normalized data from Mok et al. (2009) [23] with an accession number GSE18520 in GEO. Moreover, these data were selected from the second category of the mentioned big data, i.e., 23 genes. Besides, the true network is obtained by the STRING database.
The name of the proteins in this data set is listed as "S100A8", "SOX9", "UBE2C", "RGS1", "C7" ,"MTSS1" ,"PAPSS2" ,"CAV1" ,"DAPK1" ,"GLS", "GATM" ,"ALDH1A3" ,"CDK1" ,"ST3GAL5" ,"PDLIM5" ,"BAMBI" ,"CAV2" , "PSD3" ,"EZH2" ,"MAD2L1" ,"TBC1D4" ,"NELL2" and "PDGFRA" with a very sparse true network shown in Figure 4.7

The true graph has eight significant edges. We examined C-Vine and D-Vine to realize the edges but there were so many extra positive edges. Therefore, as it is clear from the true graph, too, the true network differs from C or D vine. That is why R-Vine is used as it is a general format of the vine copula. Through R-Vine, we detect 4 true positives (TP), 2 false positives, 4 false negatives (FN) and 243 true negatives (TN). Accordingly, F_1-measure and MCC are found as 0.57 and 0.59, respectively. Finally, the corresponding nodes of the realized edges with a copula family are shown in Table 4.6

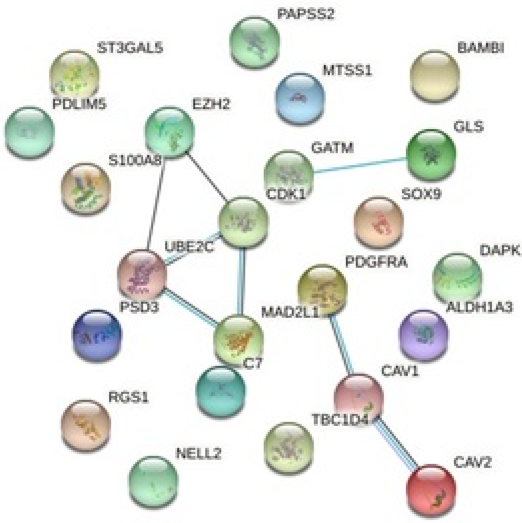


Figure 4.7: The quasi tree structure of 23 genes obtained from the STRING database.

Table4.6: Four positive edges, the best appropriate pair copula and associated Kendall's τ for data with 23 genes.

Pair of genes	Copula family	Kendall's τ
MAD2L1and CDK1	180^o rotated Joe-Clyton	0.62
EZH2 and UBE2C	Gumbel	0.52
CDK1 and UBE2C	t	0.50
TBC1D4 and CAV2	Joe	0.46

4.4.5 50 genes

Similarly to the data set in the Section 4.4.4, these data are selected randomly from the third cluster of data sets with 50 proteins. The name of available proteins of this data arelisted as "NKX3.1", "SERPINB9", "CHRDL1", "CD24", "ZNF330", "VLDLR", "RIPOR2", "TSPAN5", "SLIT2", "SLC16A1", "TGFB2", "GATA6", "IDH2", "TPX2", "PMP22", "PLA2G4A", "SF1", "TNFAIP8", "FHL1", "TPD52L1", "PTGER3", "SCTR", "DEFB1", "CLEC4M", "CDK1", "WSB1", "TFPI", "LGALS8", "DAB2", "DHRS7", "ELF3", "PLPP1", "TPM1", "IL6ST", "TCEAL2", "CASP1", "SULT1C2", "DPP4", "HSPA2", "LRIG1", "LAMB1", "MDFIC", "HBB", "TRO", "DCN", "IGFBP5", "CD44", "C1R", "ADGRG1" and "SPTBN1". Furthermore, its true network is obtained from the STRING data base, as well, which is presented in Figure 4.8. So that it is not difficult to see its accuracy by comparing with the true network. According to the

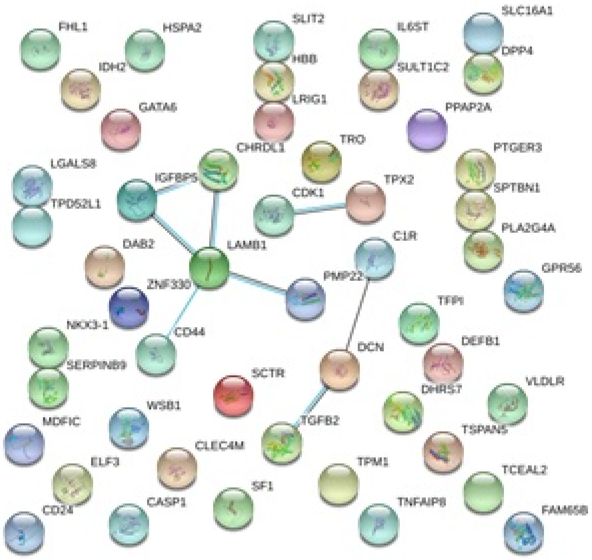


Figure 4.8: The quasi tree structure of 50 genes obtained from the STRING database

true graph represented in Figure 4.8, there are eight significant edges. C-Vine and D-Vine are tested to fit a model for these data but the number of false positive edges was large as they caught many more edges than the true network. On the other hand, it is clear from the true graph that the structure differs from C or D vine. That is why R-Vine is used as it is a general format of the vine copula. Through R-Vine, we detect TP=2, FP=1, FP=6, and TN=1216. Accordingly, F_1-measure and MCC are obtained as 0.36 and 0.41, respectively. The corresponding nodes of the realized edges with a copula family are shown in Table 4.7.

Table4.7: Two positive edges, the best appropriate pair copula and the associated Kendall's τ for data with 50 genes.

Pair of genes	Copula family	Kendall's τ
TPX2 and CDK1	Joe-Clyton	0.60
PMP22 and LAMB1	Joe	0.54

4.5 Conclusions

In this chapter, the copula method is introduced as a solution to some obstacles such as relating the normality assumption or dealing with the complex or high dimensional systems. The application of the copula starts with non-parametric methods via applying only the order of the data to have a uniformly distributed data set. Then, it is converted to a completely parametric method as it makes an exclusive joint density (distribution) function by using the marginal densities (distributions). By this way, the other advantage is shown in a way that all of the model parameters can be estimated by MLE. A better scenario of the Copula is the pair copula that makes it possible to model the data by the vine Copula. The determination of the orders of variables in the vine structure can be challenging, but, the algorithm is similar to the forward method in the multiple regression. The R-Vine copula in the vine's general format fits the model to data by the pair copula, but, its structure is not necessarily like C or D vine. In the application section, we saw that C or D vine could be a good model for some data set and its accuracy is almost the same as the RJMCMC method. We analyzed some data set by implementing the R-Vine to show its performance which is verified through some accuracy measures. But in the confusion matrix, we need to decrease the number of false negative edges as it is more crucial than false positive. To do it, we need to diminish the threshold and in this case, some false positive edges show up. So, depending on the data, and also the sparsity rate, the threshold value can play a main role in the determination of the confusion matrix.

CHAPTER 5

RESULTS AND CONCLUSIONS

In this study, the main target is to detect the relationship between the variables in every kind of data set which is mostly observed in the biological data regarding the increased attention to the mathematical modeling in biology, especially in the past decades. In Chapter 2, the relationship is the conditional dependence between genes based on a regression model of the data. In order to use the correlation matrix, the normality assumption is needed. For the continuous data, the Gaussian copula is the only choice and for the categorical data, by defining some thresholds, the normality can be caught for the joint density between the genes with the same covariance before the transformation. The whole model is called the Copula Gaussian Graphical (CGGM) which has a parameter, the precision matrix which can be estimated by the Reversible Jump Markov Chain Monte Carlo (RJMCMC) method. We investigate RJMCMC's most relevant alternatives and we compared the proposed method with them by some accuracy measures. According to the results, RJMCMC is more accurate than its alternative which is based on Bayesian approach or is not completely parametric. All of the codes are written in R, and so are not so fast. That is why we tried to start with a good initial matrix which is the inverse of the covariance matrix in the estimation of the precision matrix. We tried to catch the convergence, on the other hand.

In Chapter 3, a generalized version of CGGM was investigated by some semi-Bayesian and fully Bayesian approaches. CGGM which is repeated through time is called the Time Series Chain Graphical model (TSCGM) which includes two parameters: the precision and an autoregressive coefficient matrix. Firstly, the semi-Bayesian model was proposed as the data is supposed to have more samples by considering the time steps as a multiplication of the sample size. Accordingly, the autoregressive coefficient matrix is estimated by the Pearson correlation method as the data is supposed to be normally distributed. But the accuracy was not different from the random approximation for the autoregressive coefficient matrix. So, we tried to update the three-steps algorithm in the first chapter to a five-steps algorithm to estimate both matrices, simultaneously. Again, we saw that the convergence is very hard to catch as there were so fluctuations. To shorten the burn-in period or on other words, to catch the target matrices earlier, we used a strategy that controls the fluctuation ranges by hiring a tuning parameter which was the variance of the prior distribution for the precision matrix as well as the autoregressive matrix. By this, we could see its help to see a much better output.

Chapter 4 was about the Copula method which is more flexible and applicable than RJMCMC as it can model the non-symmetric or heavy-tailed models, too. The copula's definition, types and families are described in that chapter as well as the data analyzing format. In the application part, we examined different types of pair copula models for some real data sets. As a result, it was seen that the accuracy of the copula is almost as same as RJMCMC besides some advantages that copula has, such as detecting the tail dependence and the joint density which is very important in some cases.

To sum up, the RJMCMC performs better than its alternatives while it is not fast enough. Also, the Copula is a robust method to model the data and to estimate the joint density parameters very easily. Their accuracy is acceptable and for both methods i.e., RJMCMC and R-Vine, the determination of the threshold (and other hyper-parameters) should be done very carefully as it defines the confusion matrix to evaluate them in terms of some measures.

REFERENCES

[1] F. Abegaz and E. Wit. Sparse time series chain graphical models for reconstructing genetic networks. *Biostatistics*, 14(3):586–599, 2013.

[2] F. Abegaz and E. Wit. Copula gaussian graphical models with penalized ascent monte carlo em algorithm. *Statistica Neerlandica*, 69(4):419–441, 2015.

[3] B. Bahçivancı, V. Purutçuooğlu, E. Purutçuoğlu, and Y. Ürün. Estimation of gynecological cancer networks via target proteins. *J. Multidiscip. Eng. Sci*, 5(12):9296–9302, 2018.

[4] A. Bhadra and B. K. Mallick. Joint high-dimensional bayesian variable and covariance selection with an application to eqtl analysis. *Biometrics*, 69(2):447–457, 2013.

[5] E. Brechmann and U. Schepsmeier. Cdvine: Modeling dependence with c-and d-vine copulas in r. *Journal of statistical software*, 52(3):1–27, 2013.

[6] T. Cai, W. Liu, and X. Luo. A constrained 1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

[7] B. P. Carlin and S. Chib. Bayesian model choice via markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(3):473–484, 1995.

[8] A. Carriero, T. E. Clark, and M. Marcellino. Bayesian vars: specification choices and forecast accuracy. *Journal of Applied Econometrics*, 30(1):46–73, 2015.

[9] R. Davidson, J. G. MacKinnon, et al. Estimation and inference in econometrics. *OUP Catalogue*, 1993.

[10] D. Dereniowski and M. Kubale. Cholesky factorization of matrices in parallel and ranking of graphs. In *International Conference on Parallel Processing and Applied Mathematics*, pages 985–992. Springer, 2003.

[11] A. Dobra, A. Lenkoski, et al. Copula gaussian graphical models and their application to modeling functional disability data. *The Annals of Applied Statistics*, 5(2A):969–993, 2011.

[12] D. Gamerman and H. F. Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press, 2006.

[13] P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[14] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[15] Y. He, X. Zhang, P. Wang, and L. Zhang. High dimensional gaussian copula graphical model with fdr control. *Computational Statistics & Data Analysis*, 113:457–474, 2017.

[16] P. D. Hoff et al. Extending the rank likelihood for semiparametric copula estimation. *The Annals of Applied Statistics*, 1(1):265–283, 2007.

[17] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. Quic: quadratic approximation for sparse inverse covariance estimation. *The Journal of Machine Learning Research*, 15(1):2911–2947, 2014.

[18] A. Lenkoski. A direct sampler for g-wishart variates. *Stat*, 2(1):119–128, 2013.

[19] H. Liu, F. Han, M. Yuan, J. Lafferty, L. Wasserman, et al. High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326, 2012.

[20] A. Mohammadi, F. Abegaz, E. van den Heuvel, and E. C. Wit. Bayesian modelling of dupuytren disease by using gaussian copula graphical models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 66(3):629–645, 2017.

[21] A. Mohammadi and E. C. Wit. Gaussian graphical model determination based on birth-death mcmc inference. *arXiv preprint arXiv:1210.5371*, 2012.

[22] R. Mohammadi and E. C. Wit. Bdgraph: An r package for bayesian structure learning in graphical models. *arXiv preprint arXiv:1501.05108*, 2015.

[23] S. C. Mok, T. Bonome, V. Vathipadiekal, A. Bell, M. E. Johnson, D.-C. Park, K. Hao, D. K. Yip, H. Donninger, L. Ozbun, et al. A gene signature predictive for outcome in advanced ovarian cancer identifies a survival factor: microfibril-associated glycoprotein 2. *Cancer cell*, 16(6):521–532, 2009.

[24] B. Muthén. A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika*, 49(1):115–132, 1984.

[25] V. Öllerer and C. Croux. Robust high-dimensional precision matrix estimation. In *Modern nonparametric, robust and multivariate methods*, pages 325–350. Springer, 2015.

[26] H. Parkinson, M. Kapushesky, M. Shojatalab, N. Abeygunawardena, R. Coulson, A. Farne, E. Holloway, N. Kolesnykov, P. Lilja, M. Lukk, et al. Arrayexpress—a public database of microarray experiments and gene expression profiles. *Nucleic acids research*, 35(suppl_1):D747–D750, 2007.

[27] V. Purutçuoğlu and H. Farnoudkia. Gibbs sampling in inference of copula gaussian graphical model adapted to biological networks. *Acta Physica Polonica, A.*, 132(3), 2017.

[28] V. Purutçuoğlu, E. Wit, et al. Bayesian inference for the {MAPK}/{ERK} pathway by considering the dependency of the kinetic parameters. *Bayesian analysis*, 3(4):851–886, 2008.

[29] Y. Rahmatallah, F. Emmert-Streib, and G. Glazko. Gene sets net correlations analysis (gsnca): a multivariate differential coexpression test for gene sets. *Bioinformatics*, 30(3):360–368, 2014.

[30] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

[31] C. Sima, J. Hua, and S. Jung. Inference of gene regulatory networks using time-series data: a survey. *Current genomics*, 10(6):416–429, 2009.

[32] A. Sklar, A. SKLAR, and C. Sklar. Fonctions de reprtition an dimensions et leursmarges. 1959.

[33] D. Sun and X. Sun. Estimation of the multivariate normal precision and covariance matrices in a star-shape model. *Annals of the Institute of Statistical Mathematics*, 57(3):455–484, 2005.

[34] S. G. Walker. A gibbs sampling alternative to reversible jump mcmc. *arXiv preprint arXiv:0902.4117*, 2009.

[35] J. Whittaker. *Graphical models in applied multivariate statistics*. Wiley Publishing, 2009.

[36] T. Zhang and H. Zou. Sparse precision matrix estimation via lasso penalized d-trace loss. *Biometrika*, 101(1):103–120, 2014.

YÖK

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Farnoudkia, Hajar
Nationality: Iranian
Date and Place of Birth: 04.01.1986, Tabriz- IRAN
Email Address: hajar.farnoudkia@metu.edu.tr
Phone Number: 0553850 04 28

EDUCATION

Degree	**Institution**	**Year of Graduation**
M.S.Mathematical Statistics	University of Tabriz	September 2010
B.S. Statistics	University of Tabriz	September 2008
High School Mathematical Physics	Parvin Etesami	September 2004

PROFESSIONAL EXPERIENCE

Year	**Place**	**Enrollment**
1 year	Amar Gostar Zagros	Statistician
3 years	Perla Mensujat	Personnel master and statistician
2 years	TUBITAK (114E636)	Research assistant
2 years	UNHCR	Interpreter and case identifier
1 year	TUBITAK (118E765)	Research assistant

PUBLICATIONS

published Journal Articles

- Purutçuoğlu, V., and H. Farnoudkia., Gibbs Sampling in Inference of Copula Gaussian Graphical Model Adapted to Biological Networks. Acta Physica Polonica, A. 132.3 (2017).
- Purutçuoğlu, V., Farnoudkia, H., Copula Gaussian graphical modelling of biological networks and Bayesian inference of model parameters. Scientia Iranica, 26(4) (2019), 2495-2505.

Published Book Chapter

- Vine Copula and Artificial Neural Network Models to Understand Breast Cancer Data, DeGruther, (2020)

International Conference Publications

Published Abstracts

- Farnoudkia, H. (2016) “Gibbs Sampling vs RJMCMC”, Numerical Analysis of Stochastic Partial Differential Equations (NASPDE2016), Chalmers University-Sweden.
- Purutçuoğlu, V. and Farnoudkia, H. (2016)“ Bayesian inference of deterministic MAPK-ERK pathway via reversible jumps Monte Carlo method ”, Proceeding of the 8th International Conference Inverse Problems: Modeling and Simulation (IPMS-2016), Fethiye, Turkey, page: 1.
- Purutçuoğlu, V. and Farnoudkia, H. (2016)“ Bayesian inference of deterministic MAPK-ERK pathway via reversible jumps Monte Carlo method ”, Proceeding of the 8th International Conference Inverse Problems: Modeling and Simulation (IPMS-2016), Fethiye, Turkey.

- Purutçuoğlu, V. and Farnoudkia, H. (2016)“ Modelling of biological networks via copula Gaussian graphical model and Bayesian inference of model parameters ”, Proceeding of the 2nd Researchers-Statisticians and Young Statisticians Congress (IRSYSC), Turkey, page: 88.
- Farnoudkia, H. and Purutçuoğlu, V. (2018) “ Expectation-maximization algorithm for inference of time series chain graphical model , Proceeding of the 5th International Conference on Computational and Experimental Science and Engineering (ICCESEN 2018), Antalya, Turkey.
- Farnoudkia, H. and Purutçuoğlu, V. (2018) “ Semi-bayesian inference of time series chain graphical model in biological network , Proceeding of the International Conference on Innovative Engineering Applications (CIEA 2018), Sivas, Turkey.
- Farnoudkia, H. and Purutçuoğlu, V. (2019) “Nested Bayesian inference algorithm in the construction of time-course biological networks’ data”, Proceeding of international conference on applied analysis and mathematical modelling (ICAAMM) Istanbul, Turkey.
- Farnoudkia, H. and Purutçuoğlu, V. (2019) “ Non-Gaussian model construction of biological networks via copulas , Proceeding of 2nd Euroasia Biochemical Approaches and Technologies Congress (EBAT 2019), Antalya, Turkey.
- Farnoudkia, H. and Purutçuoğlu, V. (2019) “ Vine copula graphical models in the construction of biological networks , Proceeding of 5th International Conference on Engineering Sciences (ICES 2019), Ankara, Turkey.

Published Full Proceeding

- “Acceleration of reversible jump Markov chain Monte Carlo method within Gaussian copula graphical model”, Proceeding of the 3rd International Conference on Computational and Experimental Science and Engineering (ICCESEN-2016), Antalya, Turkey, 19-24 October, 2016.
- “Comprehensive analyses of Gaussian graphical model under different biological networks”, Proceeding of the 3rd International Conference on Computa-

tional and Experimental Science and Engineering (ICCESEN-2016), Antalya, Turkey, 19-24 October, 2016.

- “Bayesian inference of deterministic MAPK-ERK pathway via reversible jumps Monte Carlo method”, proceeding of the 8th International Conference Inverse Problems: Modeling and Simulation (IPMS-2016), Fethiye, Turkey, 23-28 May, 2016.
- “ Bayesian inference of biological networks whose components are linearly dependent ”, Proceeding of Istanbul International Conference on Progress in Applied Science (ICPAS 2017), İstanbul, Turkey, page: 1-7.
- “ Inference of time series chain graphical model , Proceeding of the International Conference on Mathematics (ICOMATH 2018), Istanbul, Turkey.
- “Vine Copula Graphical Models in the Construction of Biological Networks”, 5th International Conference on Engineering Sciences (ICES 2019), Ankara, Turkey, 19 Sep, 2019.
- “ Nested Bayesian inference algorithm in the construction of time-course biological networks data , Proceeding of the International Conference on Applied Analysis and Mathematical Modeling (ICAAMM 2019), Istanbul, Turkey.