A NON-PARAMETRIC CIRCULAR NETWORK CONSTRUCTION VIA SIMULATIONS AND A HIDDEN MARKOV MODEL FOR THE HIV-1 PROTEASE CLEAVAGE SITE DETECTION

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

BY

ELİF DOĞAN DAR

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

JANUARY 2023

Approval of the thesis:

A NON-PARAMETRIC CIRCULAR NETWORK CONSTRUCTION VIA SIMULATIONS AND A HIDDEN MARKOV MODEL FOR THE HIV-1 PROTEASE CLEAVAGE SITE DETECTION

submitted by **ELİF DOĞAN DAR** 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. Özlem İlk Dağ
Head of Department, **Statistics** _____

Prof. Dr. Vilda Purutçuoğlu
Supervisor, **Statistics** _____

Prof. Dr. Ashis SenGupta
Co-supervisor, **Indian Inst. of Technology and Augusta Uni.** _____

Examining Committee Members:

Prof. Dr. Barış Sürücü
Statistics, METU _____

Prof. Dr. Vilda Purutçuoğlu
Statistics, METU _____

Prof. Dr. Birdal Şenoğlu
Statistics, Ankara University _____

Assoc. Prof. Dr. Kadir Özgür Peker
Statistics, Eskişehir Technical University _____

Asst. Prof. Dr. Fulya Gökalp Yavuz
Statistics, METU _____

Date:27.01.2023

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, Surname: ELİF DOĞAN DAR

Signature :

ABSTRACT

A NON-PARAMETRIC CIRCULAR NETWORK CONSTRUCTION VIA SIMULATIONS AND A HIDDEN MARKOV MODEL FOR THE HIV-1 PROTEASE CLEAVAGE SITE DETECTION

DOĞAN DAR, ELİF
Ph.D., Department of Statistics
Supervisor: Prof. Dr. Vilda Purutçuoğlu
Co-Supervisor: Prof. Dr. Ashis SenGupta

January 2023, 74 pages

In this thesis, problems regarding network structures are investigated. In the first part, a circular network model is built. In the literature, methods used to detect network structures of circular variables either have strong distributional assumptions or strict structural assumptions. To address these issues, a novel circular regression-based, non-parametric circular network model is proposed. The performance of the proposed method is examined in an extensive simulation study. In the second part, a method to utilize known network structures for further analyses is exemplified. An HMM model is built for the detection of the HIV-1 protease cleavage sites. By using feature selection and fuzzy clustering methods, a clever starting point is given to the Baum-Welch EM algorithm and the performance of earlier proposed methods is increased.

Keywords: circular data, circadian gene interactions, circular networks, circular network simulation, wrapped Cauchy distribution, hidden Markov models, fuzzy clustering, feature selection

ÖZ

SİMULASYONLAR ÜZERİNDE PARAMETRİC OLMAYAN BİR DAİRESEL AĞ İNŞAASI VE HIV-1 PROTEAZ KESİM NOKTALARININ HIDDEN MARKOV MODELİYLE TESPİTİ

DOĞAN DAR, ELİF
Doktora, İstatistik Bölümü
Tez Yöneticisi: Prof. Dr. Vilda Purutçuoğlu
Ortak Tez Yöneticisi: Prof. Dr. Ashis SenGupta

Ocak 2023 , 74 sayfa

Bu tezde, ağ yapıları üzerinde çalışılmıştır. İlk aşamada, dairesel bir ağ yapısı inşa edilmiştir. Literatürde önerilen metodlar ya güçlü dağılım ya da katı ağ yapısı varsayımlarını gerektirmektedir. Bu çalışmada ise özgün bir parametrik olmayan dairesel ağ modeli geliştirilmiştir. Modelin performansı kapsamlı bir simulasyon çalışması ile incelenmiştir. Tezin ikinci bölümünde ise, önceden bilinen bir ağ yapısının karmaşık problemleri çözmede nasıl kullanılabileceği örneklendirilmektedir. HIV-1 proteaz kesim bölgelerinin belirlenmesi için bir Hidden Markov modeli geliştirilmiştir. Özellik seçimi ve bulanık kümeleme yöntemleriyle, Baum-Welch EM algoritmasına güçlü bir başlangıç noktası verip önceki metodların performansı iyileştirilmiştir.

Anahtar Kelimeler: dairesel veri, sirkadiyen gen ilişkileri, dairesel ağ yapıları, dairesel ağ simulasyonu, wrapped Cauchy dağılımı, hidden Markov modeli, bulanık kümeleme, değişken seçimi

Büyürken benim büyümeme de vesile olan Vera Melis'e...

Ve canım Salman'a...

ACKNOWLEDGMENTS

This thesis took lots of time and effort. Not only mine but many people around me. First, I would like to thank my supervisor Prof. Dr. Vilda Purutçuoğlu for her guidance and support. Especially the last part of the thesis was written in hard times indeed, COVID and various other life events all bundled together. Also, I would like to thank my co-supervisor Prof. Dr. Ashis SenGupta for supporting me all the way from India.

My biggest thank you goes to my partner and best friend Salman. He supported me all the way through, whatever I wanted to do with my life and career, he stood by next to me, holding my hand, patting my back and accepting me unconditionally for who I am. In the darkest times in my personal history, his presence gave me perseverance. I couldn't raise my daughter and do my PhD while staying sane if he was not sharing our responsibilities equally. I am forever grateful for his understanding, support and love. Salman Dar, you are the best husband and best father I had ever seen. I love you to the moon and back. And I cannot wait to have a more settled life together. Better years to come inşallah...

Another big thank you goes to my best friend, Bengi. We went through a similar career change and having a friend next to me throughout this process helped immensely. Having someone to rant about everything in this mad mad world helps a lot to stay sane. She is also the best cure for my impostor syndrome. She has the best advice on almost everything and she is the best in getting me back to my senses. Thank you sis, I hope we can become nearer in the future and continue to share our great but not-so-great-at-the-same-time lives together.

Annem Nuran ve babam Ender'e karşı da çok şükran doluyum. Aynı anda hem doktoramı bitirip hem kızımı büyütmeye çalışırken en büyük yardımcılarım oldular. Bir İzmit'e bir Ankara'ya mekik dokudular. Kardeşlerim Muhammed, Mustafa ve Seyyid Ahmed de ne zaman ihtiyaç duysam yanımdalardı. Destek sistemim 10 numara 5 yıldız. Teşekkürler gençler. Sizi seviyorum.

My last thank you goes to my family in Pakistan. Ajimama and Mushtaq uncle supported us a lot in our darkest time, we are forever grateful. We enjoyed visiting Tayi and Chachi almost every day, eating their food, and leaving my daughter with them when I was taking a much-needed break. May Allah help you with everything in life, and give peace and tranquillity to your hearts and to your loved ones. Vera also had the best uncles and aunties, thank you Ghazi Bhai, Samar, Maryam, Danish, Nida Baji and Atif Bhai for all the support and love. My Pakistani family, I wish we had shared a happier period of my life together, but alas, you helped me immensely, way more than you can ever imagine. May Allah bring ease and happiness to your lives... And my baby Vera, you are the cutest and kindest baby I have ever known. It is amazing to witness you growing every day. You bring indescribable joy and purpose to our lives. We love you to the moon and back...

TABLE OF CONTENTS

ABSTRACT v

ÖZ vi

ACKNOWLEDGMENTS viii

TABLE OF CONTENTS x

LIST OF TABLES xiii

LIST OF FIGURES xiv

LIST OF ABBREVIATIONS xvi

CHAPTERS

1 INTRODUCTION 1

1.1 Circular-Circular Networks 2

1.1.1 Motivation and Problem Definition 2

1.1.2 Proposed Methods and Models 3

1.1.3 Contributions and Novelties 4

1.2 HIV-1 Protease Cleavage Site Detection 4

1.2.1 Motivation and Problem Definition 4

1.2.2 Proposed Methods and Models 5

1.2.3 Contributions and Novelties 6

1.3 The Outline of the Thesis 6

2 LITERATURE REVIEW 9
2.1 Interactions of Circular Variables 9
2.2 Circular Network Simulation 13
2.3 Circadian Gene Networks 17
2.4 HIV-1 Protease Cleavage Site Detection 18
3 CIRCULAR NETWORKS 19
3.1 Model 19
3.2 Estimation and Inference 20
3.3 Simulations 23
3.3.1 Generating data 23
3.3.2 Results 25
3.3.2.1 Effect of the network size 25
3.3.2.2 Effect of the sample size 28
3.3.2.3 Effect of the network density 30
3.4 Circadian Gene interactions 31
3.4.1 Data 32
4 HIDDEN MARKOV MODELS 39
4.1 Model 39
4.2 Toy example 41
4.3 Calculation of likelihood 42
4.3.1 Naive approach 42
4.3.2 Forward algorithm 43
4.3.3 Backward algorithm 44

4.4 Viterbi algorithm: Inference of the most probable path 46

4.5 Baum-Welch algorithm: Estimating the model parameters 48

4.6 Application on HIV-1 Protease Cleavage Sites 49

4.6.1 Data description 49

4.6.2 Creation of states 49

4.6.3 Initialization of the EM algorithm 51

4.6.4 Modeling the data via HMM 54

4.6.5 Results 55

5 CONCLUSIONS AND FUTURE WORK 61

REFERENCES 63

CURRICULUM VITAE 72

LIST OF TABLES

TABLES

Table 3.1 Mean performance measures for 500 Monte Carlo runs. p represents the number of the nodes, n sample size and s network density. 27

Table 3.2 Circular gene expressions after transformation. 35

Table 4.1 States created by the fuzzy k-medoids approach with 9 number of states using 60 features which are determined via the hierarchical clustering. 51

Table 4.2 Corresponding states of the amino acids in the sequence AIMALKMR. 52

Table 4.3 Comparison of the proposed model with the multiple property grouping the state selection method. 60

LIST OF FIGURES

FIGURES

Figure 3.1 Mean performance measures of increasing Monte Carlo runs for random networks of 10 nodes with sample size 100. 26

Figure 3.2 A simulated network predicted by the proposed method. Blue edges are false positives and dashed edges are false negatives. Here, the number of nodes is 10 and the sample size is 100. The model predicted with an accuracy of 0.84 and an MCC value of 0.613. 26

Figure 3.3 Mean performance measures of 500 Monte Carlo runs for random networks with 10 and 20 nodes with density 0.15. The number of samples is taken as 50 and 100, respectively. 28

Figure 3.4 Mean performance measures of 500 Monte Carlo runs for random networks with 10 nodes and density 0.3. The number of samples is taken as 25, 50, 75 and 100. 29

Figure 3.5 Mean performance measures of 500 Monte Carlo runs for random networks with 20 nodes and density 0.15. The number of samples is taken as 50, 100 and 200. 29

Figure 3.6 Mean performance measures of 500 Monte Carlo runs for random networks with 10 nodes and sample size 50. The density, probability of an edge existing in the graph is taken as 0.15, 0.3 and 0.5. 31

Figure 3.7 Network structure with all circadian proteins from STRING database. 34

Figure 3.8 Sine wave fitting for the gene PER1. 35

Figure 3.9 Sine wave fitting of the gene EPHX2 for samples 8, 9 and 10. . . 36

Figure 3.10 An example of circadian sub-network. Our method detected the existent edge correctly while mistakenly claiming one additional edge. . 37

Figure 4.1 Hidden Markov model; nodes represent hidden states and observations while edges indicate the dependencies among them. 41

Figure 4.2 Counts of the transitions between states produced from the sequence AIMALKMR. 53

Figure 4.3 Transition matrix produced from the sequence AIMALKMR. . . 53

Figure 4.4 Effect of the number of states on the accuracy values. 56

Figure 4.5 Accuracy values for the hierarchical feature selection. 56

Figure 4.6 Accuracy values for the k-means feature selection. 57

Figure 4.7 Accuracy values for the k-medoids feature selection. 57

Figure 4.8 Accuracy values for the fuzzy k-means state selection. 58

Figure 4.9 Accuracy values for the fuzzy k-medoids state selection. 59

LIST OF ABBREVIATIONS

AIC	Akaike Information Criteria
BIC	Bayesian Information Criteria
CGDB	Circadian Gene Data Base
CMSE	the circular mean-square error
EBI	European Bioinformatics Institute
EM	expectation maximization
GAIC	generalized Akaike Information Criteria
GGM	Gaussian graphical network
LCD	the least circular distance
LCMSE	the least circular mean-square estimators
MCC	Matthew's Correlation Coefficient
MLE	maximum likelihood estimation
NCBI	National Center for Biotechnology Information
SVM	Support Vector Machines

CHAPTER 1

INTRODUCTION

Networks are a big part of scientific literature in multiple disciplines -from social to biological networks- to uncover hidden interactions among different entities on large scale. In this dissertation, we focus on tackling two major problems by leveraging network structures and their applications comprising independent albeit complementary chapters.

First, if we are given the data, but have little to no information about the structure, how can we deduce the structure with the data at hand? The first part of the thesis tries to answer this question with a novel circular network model where we have neither distributional nor structural information beforehand. Although the network field is a highly attractive research area, to date, networks of circular variables have received less attention from researchers. Earlier models in the circular field either have strong distributional assumptions or strict structural assumptions of the network. To address these issues, we propose a non-parametric network model which works in a more general setting. Additionally, we introduce our novel circular network simulation strategy, which we believe will be a useful contribution to the field.

In the second part of this dissertation, we try to answer the following question: Once we know the underlying structure, how can we apply this knowledge to further analyses? As a use case, we try to detect HIV-1 protease cleavage sites. To solve this classification problem, we exploit the chain-like network structure of the amino acid sequences by implementing a hidden Markov model. To improve model performance, we integrated data obtained from the physio-chemical properties of amino acids with a fuzzy clustering method. Now we will explain the problem definitions and contributions of each part in detail.

1.1 Circular-Circular Networks

1.1.1 Motivation and Problem Definition

Circular data contains directional information and is represented with angles. It can be observed in various fields such as geology (Roberts et al., 2019), ecology (Otieno and Anderson-Cook, 2006), biology (Mardia et al., 2008) or psychology (Kubiak and Jonas, 2007). We are particularly interested in interactions among circular variables which can be seen as a network. Many experimental and computational methods are constructed up to date to solve problems in the network field (Wallach et al. 2013, Meinshausen and Bühlmann 2006, Friedman, Hastie and Tibshirani 2008). However, networks constituting circular random variables are relatively under-explored.

One of the earlier methods which examine circular interactions is the application of correlation measures (Fisher 1993, Jupp and Mardia 1981, Fisher and Lee 1982). These measures quantify pairwise interactions among variables whilst not considering the joint effects of all variables at once. However, while considering networks, there needs to be a more holistic view where combined effects are taken into account. In addition to correlation measures, some model-based methods such as regression models are utilized to approach the problem (Sarma and Jammalamadaka 1993, Lund 2002, Rueda et al. 2016, Kato, Shimizu and Shieh 2008, Downs and Mardia 2002). In regression models, interactions of one dependent variable/variables with other variables are studied. Although these traditional model-based approaches hold great promise, they do not examine the system of circular random variables as a network.

To investigate interactions between circular variables, multivariate circular distributions are also proposed (Mardia and Sutton 1978). In these methods, the data are fitted to various distributions and the independence is tested via parametric tests such as the likelihood ratio test (Mardia, 1978) or the Wald test (Kim, SenGupta, and Arnold 2016). However, there are certain drawbacks to these approaches. First of all, many of these distributions are bi-variate or tri-variate (Singh, Hnizdo, and Demchuk 2002), (Shieh and Johnson 2005), and therefore, are not applicable to larger systems. In rare cases with multivariate distributions with a higher number of variables (Mardia et al. 2008), strict parametric assumptions are made, which can be sub-optimal.

There are also some network models built for circular data in specific. In a recent

paper, Gottard and Panzera (2021) build three separate graphical models based on different multivariate circular distributions. Their approach works under strict distributional assumptions. Similarly, Razavian, Kamisetty and Langmead (2011) build a graphical model where their network structure is assumed to follow multivariate von Mises distribution. In Boomsma et al. (2008), authors build mixed (linear and directional data) probabilistic Hidden Markov Models (HMMs). In their model, the underlying mechanism has to be chain-like since they use HMMs. Leguey et al. (2019) build a tree-structured Bayesian network model cleverly utilizing bi-variate wrapped Cauchy distributions. Here, the network needs to have a tree shape and all node couples should follow bi-variate wrapped Cauchy distribution. To date, no non-parametric network model has been proposed that does not require a specific graph structure. In this study, we aim to fill this gap in the field.

In specific, we are interested in circadian gene interaction networks. Circadian genes are expressed in the cells in a recurrent manner with a period of approximately 24 hours (Czeisler et al., 1999). Therefore, interactions among these genes naturally produce a circular network. Earlier, some experimental and computational models have been built for these networks (Wallach et al. 2013, McDonald and Roshbash 2001). However, there is no model-based network construction which takes circadian gene interaction data as circular. Hence, in our study, the proposed method is used after the transformation of the linear microarray data to the circular data. Accordingly, the circadian nature of the data is exploited.

1.1.2 Proposed Methods and Models

To address the issues with the earlier models, we propose the construction of a network model which does not have any distributional assumption and works in a more general setting. We regress each node on others (Kim and SenGupta, 2016) via circular least squares estimation where regression coefficients quantify the strength of the interactions among nodes. We infer the variance-covariance matrix of the resulting estimators with bootstrapping. Finally, we decide which interactions are statistically significant via the backward elimination technique utilizing the fact that the estimators are asymptotically normal. Unlike correlation measures and regression-based methods, the proposed model is taking combined effects into account and builds a

system where each node affects the others. Moreover, in the estimation step, we use circular least square estimation which doesn't require any distributional assumptions. Therefore, our model is more flexible and can be used for data coming from any distribution as well as for the cases where underlying distribution is not known. Finally, unlike tree-structured and HMM methods, our model can be applied in a more general setting where the underlying structure of the network is not known.

1.1.3 Contributions and Novelties

- Proposed network model is non-parametric, therefore can be used for networks coming from any distribution.
- Proposed network model does not assume a network structure at the beginning. Therefore, it can be applied in a more general setting.
- We propose a novel simulated data generation for circular networks which can be used to compare the efficiencies of different network models in the circular field, which was not available to date.

1.2 HIV-1 Protease Cleavage Site Detection

1.2.1 Motivation and Problem Definition

AIDS (acquired immunodeficiency syndrome) is a disease that weakens the immune system by reducing the T-cells in the body which fight off infections (Schroff et al. 1983). In 2021, around 38 million people globally were living with HIV. Furthermore, approximately 650 thousand people died from AIDS-related illnesses in 2021 making a total of 40 million people since the start of the epidemic (UNAIDS, 2021). HIV-1 (human immunodeficiency virus-1) is the virus which causes AIDS gradually (Gallo et al. 1984). However, the HIV-1 protease enzyme is needed for HIV-1 to be active. It cleaves newly synthesized polyproteins of the host cell to create the mature protein components that an HIV virion requires (Kohl et al. 1988). Usually, this enzyme extends to 8 amino acids long octamer sites on the polyprotein to cleave between the

4th and 5th amino acids (Miller et al. 1989). Occasionally, these cleavage sites can also be heptamers or nonamers. HIV-1 protease inhibitors are good candidates for AIDS treatment. Therefore, learning the key and the lock relationship between the enzyme and cleavage sites is crucial to finding the proper inhibitor key which locks the enzyme and prohibits it to create new active proteins for the virion. On the other hand, it is impossible to experimentally test all possible cleavage sites. Because there are 20 amino acids possible at each position, resulting in $20^8 = 2.56 \times 10^{10}$ cleavage sides. Hence, several methods have been used in the literature to predict the cleavage sites given earlier experimentally checked data.

The detection of the cleavage site in Chip-seq data is one of the main interests to find the lock-and-key relationship between enzymes and prohibits in certain diseases such as AIDS and producing the proper inhibitors for these illnesses. For this detection, different approaches like support vector machines and artificial neural networks have been suggested. In this study, we use the hidden Markov model (HMM) for cleavage site detection. In our application, initially, we comprehensively explain the mathematical details of HMM and the inference of the model parameters, and then we discuss the effect of various clustering approaches both in feature selection and state formation. We demonstrate the calculation of each step in a toy and bench-mark dataset and evaluate the accuracy of estimates with other approaches in the literature.

1.2.2 Proposed Methods and Models

We build a Hidden Markov model to classify the amino acid chains as cleaved or non-cleaved. During the inference, we use the Baum-Welch EM (expectation- maximization) algorithm, which can converge to a local maximum instead of the global maximum. Hence, we propose a method to give a clever starting point to the algorithm in order to increase the probability of reaching a global maximum instead. For this purpose, first, we utilized 544 features for each amino acid in the AAIndex database. Since many machine learning algorithms suffer from the high dimensionality of the feature set, we implement a clustering-based feature selection approach in our analyses (Mitra, Murthy and Pal 2002, Chormunge and Jenab 2018). Here, we apply different strategies. Initially, we examine the k-means clustering and the hierarchical clustering methods based on the correlation measures and then, we randomly

choose a representative from each cluster. Also, we perform the k-medoids clustering and select medoids as the cluster representatives. Later, we create hidden states including amino acids using these reduced number of features. An amino acid can be grouped in multiple ways according to common features. Due to this fact, the nature of the problem can be considered under a fuzzy clustering (Bezdek 1981, Gustafson and Kessel 1978, Krishnapuram et al. 2001) and in this study, we use these fuzzy methods to build the hidden states. After giving the HMM algorithm a clever starting point using the aforementioned methods, we decide whether a given amino acid chain is cleavable or not. By using the proposed approaches, we show that the states which we create give better results than the earlier states suggested by Zhang et al. (2006).

1.2.3 Contributions and Novelties

Our contributions and novelties in this work can be summarized as follows:

- We propose the application of fuzzy clustering instead of hard clustering. It shows an alternative way to utilize the physio-chemical features of amino acids, which is more reliable when applicable.
- We present a toy example in detail. It clarifies how the model works for inexperienced researchers or practitioners. Following this example, they can easily adapt the HMM model to their use cases.
- The proposed model gives better results for classifying octamers as cleaved or non-cleaved because of the clever starting point that we create. This model can be used to narrow down the search grid in experiments investigating cleavage sites of amino acid chains regarding any illness.

1.3 The Outline of the Thesis

All in all, organization of this thesis is as follows. We give a comprehensive literature review of the subjects in Chapter 2. In Chapter 3, we explain circular networks. In Section 3.1, we present our model. We explain the estimation and inference steps in

detail in Section 3.2. Afterwards, we present a way to simulate data for the interactions of the circular variables in Section 3.3. We apply our model and show the results for this simulated dataset. Finally, in Section 3.4, we apply our method to circadian gene interaction networks. In Chapter 4, we present Hidden Markov Models. We explain the model in Section 4.1. We introduce a toy dataset in Section 4.2 and show the likelihood calculations, inference and estimation procedures on this dataset in Sections 4.3, 4.4 and 4.5 respectively. In Section 4.6, we present the application of HMMs in HIV-1 protease cleavage dataset and discuss the outputs. Finally, in Chapter 5, we give final remarks and future directions for the subjects discussed.

CHAPTER 2

LITERATURE REVIEW

2.1 Interactions of Circular Variables

Circular data contains directional information which represents a recurrent phenomenon that repeats in fixed time intervals. It can be observed in various fields such as geology (Roberts et al., 2019), ecology (Otieno and Anderson-Cook, 2006), biology (Mardia et al., 2008) or psychology (Kubiak and Jonas, 2007). The traditional statistical theory does not take into account the circular nature of such data. Therefore, all statistical theory has to be built from the ground up according to the topology of the circular data which is inherently different from linear data. Our aim in this work is to build network models for circular data. We are interested in interactions among circular variables which can be seen as a network structure. Nodes represent variables and edges represent interactions among them. Networks are a big part of scientific literature in various fields where interactions among different entities are examined. However, networks constituting circular random variables are not received much attention from researchers. There are only a few works done on network building in a circular setting. We aim to fill this gap in circular theory by building a novel model based on circular regression.

Relations between circular variables are examined earlier via different approaches. Circular correlation coefficients are used to determine whether there is a relation/interaction between variables or not (Jupp and Mardia 1981, Fisher and Lee 1982, Jammalamadaka and Sarma 1988, Fisher 1993). Some of the measures in the related field consist of some rank-based coefficients as suggested by Fisher (1993), and Fisher and Lee (1982), correlations related to sine and cosine components of the variables as proposed by Jupp and Mardia (1981), and a correlation coefficient is used by Jam-

malamadaka and Sarma (1988) which is an analogue of the Pearson's correlation in the circular setting. Unlike our model, these approaches take only the pairwise interactions into account, instead of looking at the cumulative effect of all variables at once. While considering networks, there needs to be a more holistic view where combined effects are taken into account. Unlike these previous studies, here, the effects of all variables in a network are inferred simultaneously by using a model-based method.

To investigate interactions between circular variables, multivariate circular distributions are also proposed. In these methods, the data are fitted to various distributions and the independence is tested via parametric tests such as the likelihood ratio test and the Wald test (Mardia and Sutton 1978, Kim et al. 2016), whereas, there are certain drawbacks of these approaches. First of all, many of these distributions are bivariate or tri-variate, and therefore, are not applicable to larger systems. For example, Singh et al. (2002) propose a bi-variate circular distribution to explore the relation between dihedral angles of peptides. Similarly, Shieh and Johnson (2005) apply the bi-variate von Mises distribution to model the relation between wind directions at two different times of the day at the same place. Here, we can only investigate the relationship between two circular entities. In rare cases, multivariate distributions with a higher number of variables are utilized. For example, Mardia et al. (2008) construct a multivariate von Mises distribution and apply it to dihedral angles of gamma turns on peptides. Here, their model can also be performed for models with more than two variables. However, strict parametric assumptions are made, which can be sub-optimal. Unlike these methods, we apply a non-parametric estimation method without any distributional assumptions, rendering the model more flexible.

We also see various circular regression models for the investigation of the interactions among circular variables. Earlier models consist of mixed models, i.e. regression of circular variables on linear variables and vice versa. Gould (1969) regress circular variable on linear variables and uses the von-Mises distribution for the errors with a maximum likelihood approach. Laycock (1975) implements trigonometric regression to regress a linear variable on a circular variable. Mardia and Sutton (1978) define an angular-linear joint distribution and use the conditional distribution of a linear variable on the circular variable for the regression. Johnson and Wehrly (1978) also define various angular-linear distributions and use conditional distributions and

maximum likelihood estimation (MLE) for regression. They have also built some parametric tests for these models. However, they regress only one circular variable on one linear variable or they regress a linear variable on other linear and angular variables. Lund (1999) applies the least circular distance (LCD) which is a circular alternative to the Euclidean least squares for the regression of a circular variable on linear variables and one circular variable. He also shows that LCD estimation gives the same results as the maximum likelihood estimation by assuming von Mises errors for the regression. He assumes additive effect of linear covariates with a link function which maps the real line to the circle and Fourier series approximation for the circular regressor. He uses Akaike Information Criteria (AIC) for model selection and numerical methods for MLE estimation. He also says squaring circular correlation measures applied to predicted and observed values gives a circular analogue of the R^2. When specific distributions are assumed for the errors, the goodness of fit tests has to be applied to verify that the assumption is valid. Some goodness of fit statistics for von Mises distribution can be found in Lund (1998) and Lockhart and Stevens (1985). Fisher and Lee (1992) generalize their model by using a link function that sends the real line to the circle in a monotone way and they solve the identifiability problem of earlier models.

There are also models where both dependent and independent variables are circular. Sarma and Jammamaladaka (1993) build a circular regression model with one dependent and one independent circular variable, using polynomial models for the sine and cosine of the dependent variable. Since they estimate sine and cosine separately, they use the usual least squares estimation. Rivest (1997) builds a regression of circular variable on another but his model only takes some type of rotation into account. Lund (2002) proposes a tree-based method for circular-circular regression with one regressor using binary trees. Downs and Mardia (2002) regress one circular variable on another by mapping centralized half angles to the real line by tangent function, multiplying by a coefficient and pulling it back to the circle using arc-tangent. They use a maximum likelihood estimation approach assuming errors following von Mises distribution. Rueda et al. (2015) generalized this model by constructing a circular piece-wise regression model where each piece is established using the model of Downs and Mardia. They applied their method to cell-cycle biology where modelling of multiple phases of the cell cycle cannot be done by one single form. They also

construct a generalized Akaike Information Criteria (GAIC) which can be seen as an analogue of AIC on the linear case. SenGupta et al. (2013) also generalize Downs and Mardia model by adding an intercept parameter into the link function. They also used Asymmetric generalized von Mises distribution to model the error which is a more flexible distribution compared to earlier distributions used. They also suggested three new distance-based methods where no parametric assumptions need to be done. They define a new criterion, relative circular prediction bias (RCPB), to select the best estimator among the suggested ones. Kato et al. (2008) use Möbius transformations to model circular-circular regression, Möbius transformations are mappings from the complex plane to itself, with some restrictions it maps unit circle to itself in a one-to-one fashion. Their model resembles Downs and Mardia model, however, they assume the Wrapped Cauchy distribution for the error instead of the von Mises distribution which results in some desired properties. Kato and Jones (2010) generalize their method by introducing a family of distributions which corresponds to the Möbius transformations on the circle, They use the Bayesian information criterion (BIC) as well as AIC for the model selection. Di Marzio et al. (2013), however, apply non-parametric methods for the regression. Moreover, they perform a local smoothing and therefore, avoid forcing a specific shape as a general link function. Alonso-Penaa (2021) suggested some non-parametric tests for many models in the literature. However, all these aforementioned models consider only one regressor. Therefore, they cannot be used for large systems.

Additionally, there are multiple circular circular regression models. The model which we consider using in this study is a multiple circular-circular regression model by Kim and SenGupta (2016). In this model, they generalize the model in SenGupta's earlier model (SenGupta et al. 2013) by including multiple regressors. They implement the circular version of the mean square error where they minimize the mean of the circular distances between estimated values and observations. Then, they prove the asymptotic normality of the suggested estimators for inference purposes.

Experimental methods are also used for network building (Wallach et al. 2013, Baggs et al. 2009). However, these methods are computationally demanding and require a high amount of resources. Furthermore, false positive and false negatives rates are high for many of them. Cheaper, holistic and computational methods might guide experimental work for further validation and result in increased efficiency. For exam-

ple, McDonald and Roshbash used microarray data for this purpose (2001). Yet, their method only accounts for fold change instead of a model-based approach. Our model is a novel approach which constructs this circular network using a model-based approach utilizing its circular nature. We will apply a circular version of the Gaussian Graphical Network (GGM) model (Dempster, 1972). In GGM, the nodes are assumed to be following a Gaussian distribution for the inference (Edwards, 2000) and zeros in the precision matrix correspond to the conditional independence of two nodes given others (Wermuth, 1976). This problem can equivalently be formulated with all nodes regressed on others one by one, and coefficients of regressors give the strength of the conditional dependence between nodes. Many methods have been established for the underlying inference problem including the coordinate descent for the maximization of the L1-penalized log-likelihood (Yuan and Lin 2007), lasso (Meinshausen and Bühlmann 2006), blockwise coordinate descent (Banerjee et al. 2008) and graphical lasso (Friedman et al. 2008). However, all these models are built for linear data.
There are also some network models built for circular data too. In a recent paper, Gottard and Panzera (2021) build three separate graphical models based on different multivariate circular distributions. Their approach works under strict distributional assumptions. Similarly, Razavian, Kamisetty and Langmead (2011) build a graphical model where their network structure is assumed to follow multivariate von Mises distribution. In Boomsma et al. (2008), authors build mixed (linear and directional data) probabilistic Hidden Markov Models (HMMs). In their model, the underlying mechanism has to be chain-like since they use HMMs. Leguey et al. (2019) build a tree-structured Bayesian network model cleverly utilizing bi-variate wrapped Cauchy distributions. Here, the network needs to have a tree shape and all node couples should follow bi-variate wrapped Cauchy distribution. Unlike these models, our model does not assume any underlying distribution or network shape. Therefore, it is applicable to a wider range of problems.

2.2 Circular Network Simulation

We also apply our method to general cases via simulations using wrapped normal distribution and various graphs with different topologies. This way, we aim to show

the applicability of our model to various fields where graphs of circular data emerge. The effects of underlying network structures on the model's performance are examined using simulated data. Simulation data is produced on which model performance for different network parameters is assessed. Data is generated from the multivariate normal distribution with pre-determined symmetric precision matrices which gives a conditional dependence measure between variables given others where zeros correspond to conditional independence. This data is transformed into circular wrapped normal distribution. This way, circular data with a known network structure is designed for model evaluation. The estimated networks are compared with the ground truth networks using multiple model performance criteria such as accuracy, precision, sensitivity, specificity, F1 score and Matthews correlation coefficient. Here, the main objective is to observe the effect of network parameters on model performance.

Simulation data can be produced via multivariate normal distribution. If vertices are multivariate normally distributed, then conditional independence corresponds to zero entries in the inverse covariance matrix. Therefore, we can start with a graph and generate multivariate normally distributed data given an inverse variance-covariance matrix with zeros if there is no edge between corresponding nodes.

In literature, it has been done in multiple ways. First of all, the variance-covariance matrix has to be symmetric. Therefore, the precision matrix which is the inverse of the variance-covariance matrix has to be symmetric too. Also, the precision matrix has to be positive definite to be invertible. In many of the applications in the literature, to be able to produce a positive definite matrix, authors use the fact that a symmetric matrix which is diagonally dominant and has all diagonals positive, is positive definite. Here, diagonally dominant means the absolute value of the diagonal entries is greater than or equal to the sum of the corresponding rows.

Schafer and Strimmer (2005) start with an empty symmetric matrix with zero diagonals. They decide the number of nodes, edges and sample size. Then they choose off-diagonal elements which correspond to existing edges randomly according to the number of edges chosen at the beginning. They set these non-zero entries from the uniform distribution between -1 and 1. Afterwards, they set diagonal elements to the sum of the absolute values of the columns plus a small constant (e.g. 0.0001) so that the matrix becomes positive definite. Then they standardize the matrix so that all diagonal entries are 1.

In Meinshausen and Bühlmann (2006), they locate all nodes on two-dimensional square $[0,1]^2$ uniformly and randomly. Then, they declare an edge between them with a probability proportional to their distance and inversely proportional to the number of nodes. Since they work on sparse graphs, they put a condition that each node can have at most 4 edges. Therefore, they remove edges which don't satisfy this constraint randomly until all nodes have a maximum 4 number of edges. They declare diagonal elements of the precision matrix as 1 and entries corresponding to edges as 0.245, this way since each row has at most 4 entries of 0.245, sum of them will be less than 1 and diagonal dominance will be satisfied. Finally, they rescale all variables such that diagonal elements are 1 in the covariance matrix. They don't explain this procedure in detail, but rescaling causes problems with symmetricity. A possible meaning for the normalization of the matrices can be as follows. If we take a symmetric matrix and multiply it from both sides with the inverse of the diagonal matrix whose non-zero entries are the same as the square root of the diagonals of the original matrix, we get a symmetric matrix whose diagonals are 1. In mathematical notation, if we have a symmetric matrix Σ, then $D^{-1}\Sigma D^{-1}$ with diagonals as 1, where $D = \sqrt{diag(\Sigma)}$.
Li and Gui (2006) are generating networks with different sparsities by setting a different number of neighbours for each node, but in their graphs, each node has the same number of edges. They uniformly and randomly distribute nodes on the two-dimensional square and add an edge with k-nearest neighbours of each node. For each entry in the precision matrix corresponding to the edges, they put a random number from $[-1,-0.5] \cup [0.5,1]$. Each diagonal entry was added as a factor of the sum of the absolute values of the rows. The factors they used are 2, 1, 0.8 and 0.5. Then each row is divided by the diagonal entry to make it 1. Here, positive definiteness is not guaranteed for factors 0.8 and 0.5. And since each row is divided by a different number symmetry is lost. In Fan, Feng and Wu (2009), they applied the same procedure pointing out that using a factor of 2 will guarantee positive definiteness.
Following Schafer and Strimmer (2005) and Li and Gui (2006) with some modifications, we can set the number of nodes, p. We randomly and uniformly choose p positions from the lower triangle of an empty symmetric matrix with dimensions $p \times p$. We randomly assign a number to those entries from $[-1,0.5] \cup [0.5,1]$. We choose from both positive and negative values since all positive values in the precision matrix will result in most of the values in the inverse matrix, variance-covariance

matrix as negative, where we don't need that kind of restriction. Also, instead of using $[-1, 1]$, we use $[-1, 0.5] \cup [0.5, 1]$, so that the association is strong and does not become negligible. Then we fill the upper triangle so that matrix becomes symmetric. We set diagonal entries to the sum of the absolute values of the corresponding rows plus a small term, 0.005 to ensure diagonal dominance, hence positive definiteness. We divide the row by the diagonal entry to make it 1. Since we divide each row with a different number, symmetry is lost, therefore, we added another step. We set both $\Sigma^{-1}(i, j)$ and $\Sigma^{-1}(j, i)$ to the value whose absolute value is minimum. This way, symmetry and diagonal dominance are retained. We take the inverse of this precision matrix we produced and generate multivariate normal data with mean 0 and variance-covariance matrix Σ^{-1}. Then we wrap this data around $[-\pi, \pi]$ to produce wrapped normal data.

Another way to simulate data following Meinshausen and Bühlmann (2006) with some modifications can be as follows. We set the number of nodes, p. We randomly and uniformly choose p positions from the lower triangle of an empty symmetric matrix with dimensions $p \times p$. We check the number of non-zero entries for each row. Let's say the maximum number of this value is x. We set non-zero entries to either $\frac{1}{x} - 0.005$ or $-\frac{1}{x} + 0.005$ randomly. This way diagonal dominance is ensured. We fill the upper triangle so that it becomes symmetric. We fill diagonals with 1's. And the rest is the same as earlier data generation steps.

In our study, we used a data generator from R-package Huge for different types of graphs. In their implementation, the adjacency matrix theta has all diagonal elements equal to 0. Also, each pair of off-diagonal elements of the adjacency matrix are randomly set $\Theta[i, j] = \Theta[j, i] = v$ for $i \neq j$ with a given probability, and 0 otherwise. Instead of giving a random number to each non-zero value, they give the same number, default 0.3, to the magnitude of partial correlations. To obtain a positive definite precision matrix, the smallest eigenvalue of Θ is computed. Then, they set the precision matrix equal to $\Theta + (|\Theta| + 0.1 + u)I$. The covariance matrix is then computed to generate multivariate normal data.

2.3 Circadian Gene Networks

There are different methods used by researchers to reveal interactions among circadian genes. Earlier work on this subject mostly consists of experimental methods. Wallach et al. (2013) use a yeast-two-hybrid approach where the interaction of each pair of proteins is tested in a yeast medium. However, this approach results in high quantities of false negatives and false positives. Two genes interacting in yeast medium might not interact in human cells or two proteins artificially put together might not encounter in the cell due to spatial limitations. Therefore, they try to validate co-immunoprecipitation experiments in human cells as well as the enrichment of the network from the literature. Baggs et al. (2009) use a perturbation-based model where they knock down certain genes and analyze the expression of clock genes after the knockdown. But this method is also not immune to false negatives, because, usually in biological networks there are alternative paths which result in undetected interactions. These methods are computationally demanding and require a high amount of resources. False positive and false negatives rates are high for many of them.

Therefore, computational methods and high-throughput data sets are also suggested in the literature to make a grid search smaller for experimental validation. Since experimental approaches are computationally demanding and expensive, there should be alternative approaches for building interactions by using high-throughput datasets. For example, McDonald and Roshbash (2001) use microarray data for this purpose. They fit cosine curves and apply the cross-correlation coefficients and fold changes for the analysis. Yet, their method only accounts for the fold change instead of a model-based approach. Moreover, they treat data as linear although data is inherently circular in nature. We propose transformations of this linear data to circular data to utilize the circular nature of the data.

To detect circadian genes and reveal their periodic patterns, sine waves have been used (Hickey et al. 1984), however, circadian patterns do not always follow sinusoidal waveforms. Luan and Li (2004) used cubic B-splines instead, which give a more flexible fit compared to the sine waves. However, they need guide genes which are known as circadian for the implementation of their method. For non-sinusoidal form detection, also, Lomb–Scargle periodograms have been used by Glynn et al. (2006), the Laplace periodogram has been used by Liang et al. (2009) and bayesian

methods have been suggested by Chudova et al. (2009). Unfortunately, it is not quite obvious how to transform microarray data to angular data by using these methods, unlike traditional sinewave forms. Therefore, we proceed with the sinewave forms.

2.4 HIV-1 Protease Cleavage Site Detection

In the literature, some well-known approaches are used to predict HIV-1 cleavage sites including support vector machines (SVMs) (Cai et al. 2002), artificial neural networks (ANNs) (Cai and Chou 1998, Thompson, Chou and Zheng 1995) and different encoding techniques (Turhal, Gök and Durgut 2014), such as orthonormal encoding (OE) (Nanni 2006, Cai and Chou 1998). In addition, the physicochemical properties of amino acids are being used in many papers. For instance, Jaeger and Chen (2010) use 4 biophysical properties, namely, hydropathy index, molecular mass, polarity and occurrence percentage, and Kim et al. (2010) suggest a feature subset selection method using multi-layered perceptron (MLP) learning. Also, some researchers perform a subset of physicochemical properties from the AAIndex database (Niu 2013, Turhal, Gök and Durgut 2014). For a comprehensive review of the HIV-1 cleavage site detection, Rögnvaldsson (2015) can be also seen.

Hereby, the present study proposes a hidden Markov model to capture the sequential nature of the problem. The hidden Markov model (HMM) is a model which utilizes the sequential relationship of the data unlike many other methods above. HMM is successfully applied to speech (Juang and Rabiner 1991), handwriting (Jianying, Brown and Turin 1996) and gesture recognition (Starner and Pentland 1995) as well as biological applications such as the sequence alignment (Durbin et al. 1998, Pachter, Alexandersson and Cawley 2002), gene prediction (Munch and Krogh 2006) and the protein modelling (Stultz 1993, White 1994). HMM is also implemented to the HIV-1 cleavage site detection problem (Jayavardhana, Rama and Palaniswami 2005). However, in this model, random starting parameters are given to HMM instead of a guided choice. In the present study, guided starting parameters using physicochemical properties of the amino acids from the AAIndex database (Nakai, Kidera and Kanehisa 1988) are proposed inspired by the work of Zhang et al. (2006).

CHAPTER 3

CIRCULAR NETWORKS

In this chapter, we propose a novel non-parametric circular network model. Earlier models built for circular networks either had strong distributional assumptions or strict structural assumptions. Here, we build the network for the cases when we have no distributional or structural information beforehand. We explain estimation and inference steps in detail. Afterwards, we generate simulated data by using wrapped normal distributions with a novel strategy, and show our model's performance on this data set. This is a novel strategy which we believe that circular network field will benefit from. Furthermore, we apply our model to circadian gene interaction networks which are circular in nature.

3.1 Model

In this work, we propose to apply the multiple circular regression model defined by Kim and SenGupta (2016) for describing the interactions. For each node, we build a separate regression model where that node, which is a gene or protein in the biological networks, is the dependent variable and the remaining nodes are independent variables. In this network, the regression coefficients which are statistically zero indicate that there is no edge among associated nodes. That is there is no interaction among the related genes or proteins. Other coefficients represent an edge between the dependent variable and the corresponding independent variable, i.e., support the existence of functional/physical interactions.

Considering that $\phi_1, ..., \phi_{p-1}$ are independent variables and θ is the dependent vari-

able, our model can be represented as below:

$$\theta_i = \mu_\theta + 2\arctan\Bigg(\alpha + \beta_1 \sin\left(\frac{(\phi_{1,i} - \mu_{\phi_1})}{2}\right) + \ldots + \beta_{p-1}\sin\left(\frac{(\phi_{p-1,i} - \mu_{\phi_{p-1}})}{2}\right)\Bigg) + \epsilon_i,$$

where ϵ_i follows a circular distribution with a mean 0, $\alpha \in \mathbb{R}$ and, $\beta_1, ...\beta_{p-1} \in \mathbb{R}$ denote the intercept and slope parameters, respectively. Furthermore, μ_ϕ stands for the centering constant of ϕ for $i = 1, 2, ..., n$ and μ_θ refers to the centering constant of θ. Moreover, $arctan(.)$ and $sin(.)$ represent the arctangent and sinus of the given values, respectively. Additionally here, n denotes the number of data points and p describes the number of nodes in our system. Finally, the values of circular random variables $\phi_1, ...\phi_{p-1}$ and θ are in $[-\pi, \pi)$. As a result, by dividing values by 2, we get unique values for sin. α in the model is put because otherwise, whenever $\phi_1 = \mu_{\phi_1}, ..., \phi_k = \mu_{\phi_k}$, we would get $\theta = \mu_\theta$. Using this regression model, we build p regressions where the dependent variable is one of the nodes at each time. In this expression, $\beta_1, ..., \beta_{p-1}$ give the strength of the interactions among the corresponding node θ and the remaining nodes $\phi_1, ...\phi_{p-1}$ in the network.

3.2 Estimation and Inference

For the estimation of these parameters, we use the circular mean-square error (CMSE), which is the circular version of the traditional least squares estimation. This specific approach does not require any distributional assumptions. The circular mean square error of an estimator $\widehat{\mu_0}$ of a circular parameter μ_0 is defined as follows.

$$CMSE = E\left(1 - \cos\left(\widehat{\mu_0} - \mu_0\right)\right).$$

In the above equation, CMSE is 0 when $\widehat{\mu_0} - \mu = 0$ and it equals to 2 when $\widehat{\mu_0} - \mu = \pm\pi$. For other values, it takes values in between. $E(.)$ denotes the expectation of the given term and $cos(.)$ implies the cosine value. In our model, we aim to minimize sample CMSE via

$$Q\left(\alpha, \beta_{1,...,}\beta_{p-1}, \mu_\theta, \mu_{\phi_1,...,}, \mu_{\phi_{p-1}}\right) = \frac{1}{n}\sum_{i=1}^{n}\left[1 - \cos\left(\theta_i - \hat{\theta}_i\right)\right],$$

where

$$\hat{\theta}_i = \mu_\theta + 2\arctan\left(\alpha + \beta_1 \sin\left(\frac{(\phi_{1,i} - \mu_{\phi_1})}{2}\right) + \ldots + \beta_{p-1}\sin\left(\frac{(\phi_{k,i} - \mu_{\phi_{p-1}})}{2}\right)\right).$$

These resulting estimators are called the least circular mean square estimators (LCMSE) and for large n, they are asymptotically normal (Kim and SenGupta 2016). Minimizing this CMSE function is a numerical optimization problem since it does not have a closed-form solution. Therefore, for the optimization problem, we apply a Newton-type algorithm which is performed by nlm() function in the base R programming language. In many optimization problems, we encounter problems such as non-convergence, being stuck at a local minimum etc. Therefore, we check codes after each estimation which indicates why the optimization process terminated, i.e. whether it was successful or it failed. Since we use CMSE which is a non-parametric approach, there is no need to specify the distribution of the error except for the mean being zero. Also, by using the large sample theory, μ_{ϕ_i}'s can be taken as the mean of the sample ϕ_i's.

For the inference, we utilize the fact that the Least Circular Mean Square Estimators (LCMSE) are asymptotically normal. Kim and SenGupta (2016) prove that these estimators are asymptotically normal, however, their proof is existential instead of constructive. Therefore, we propose to implement the bootstrap method in order to estimate the variance-covariance matrix. Hereby, we resample with replacement, fit the model with new data and find the regression coefficients. Then, we repeat this process m times. The resulting distribution of β_i's gives the sampling distribution and the variance-covariance matrix of regression coefficients. Finally, to decide which coefficients are statistically different than zero, we consider performing a backward elimination method which is described below.

Let's assume that we try to decide whether the correlation coefficients are significantly different than 0 or not while regressing the node i on other nodes where $i = 1, 2, ..., p$. First, we can test if any of these $(p-1)$ coefficients are 0 given all other genes exist in the model, namely, $H_0 : \beta_j = 0$ where $j \in \{1, 2, \ldots, p\} - \{i\}$. Since LCMSE are asymptotically normal, we have $\frac{\widehat{\beta}_j}{\text{se}(\widehat{\beta}_j)} \sim N(0, 1)$ while $se()$ shows the standard error of the given random variable. By computing the bootstrap standard deviation as the standard error of the coefficients, we apply the z-test. Then, we choose the coefficient with the largest p-value among the ones which are more than the se-

lected significance level $\alpha = 0.05$ and declare this coefficient as zero, say $\beta_x = 0$. Later, we can calculate the Wald-test to check the hypothesis $H_0 : \beta_x = \beta_j = 0$ where $j \in \{1, 2, \ldots, p\} - \{i, x\}$ given all other coefficients significant. Here, we use the fact that $\left[\widehat{\beta}_x, \widehat{\beta}_j\right] [\mathrm{Cov}(\hat{\beta})]^{-1} \left[\widehat{\beta}_x, \widehat{\beta}_j\right]' \sim \chi^2_2$ where $\mathrm{Cov}(.)$ and $(.)'$ indicate the variance-covariance matrix of the coefficients and the transpose of the given vector respectively. Again among β_j's, we can select the one which gives the largest p-value which is more than α. We can declare this coefficient to be statistically zero together with β_x. Continuing this way, until none of the p-values is larger than α, we aim to find all coefficients which are statistically zero. We repeat the same process for each node in the network. Accordingly, all coefficients which are statistically different than zero will correspond to edges and others as non-edges in the adjacency matrix A of the network.

Here, the adjacency matrix is a binary matrix composed of 0 and 1 values. Edges are represented by 1 which shows that the pairwise regression coefficients are statistically significant and 0 entries show the conditional independence between the associated pair of elements as their regression coefficient is not significantly different than 0 at given α. Hence, A represents the neighbourhood matrix of the network and should be symmetric and square. Accordingly, as the last step, we transform our adjacency matrix to make it symmetric. Here, there is no bound in our model which forces $A_{ij} = A_{ji}$. Therefore, we apply an "AND Rule" in such a way that we set the edge between the ith and the jth gene existent if both A_{ij} and A_{ji} are nonzero. Similarly, an "OR Rule" could have been applied. "OR Rule" produces denser graphs.

In order to evaluate the performance of the estimated network, we consider computing the accuracy, precision, specificity, recall and Matthew's Correlation Coefficient (MCC) values. In the calculation, accuracy, precision, specificity and recall can help us to evaluate the model performance from different standpoints whereas MCC gives a more general view which can be used alone for the evaluation of the performance as an average of the earlier metrics.

Hereby, to summarize, our algorithm can be described as follows:

1. (*Initialization*) Number of bootstraps (m: default 50) and alpha value(α: default 0.05) is set.

2. (*Model*) For each node, we build a circular regression model where the chosen

node is the dependent variable and other nodes are independent variables.

3. (*Estimation*) We estimate regression coefficients with nlm() function by minimizing CMSE.

4. (*Bootstrapping*) To estimate the variance-covariance matrix of the regression coefficients, we resample with replacement and repeat Step 3 with the new data. We repeat this m times.

5. (*Inference with backward elimination*) We utilize the variance-covariance matrix inferred in Step 4 and apply the z-test and Wald-tests to deduce insignificant coefficients.

6. We repeat Steps 2-5 for each node.

7. We apply the "AND Rule" to make the adjacency matrix symmetric.

8. (*Output*) Output circular network with accuracy measures.

3.3 Simulations

3.3.1 Generating data

This project can be applied to data from different fields where a network structure of circular data exists. Therefore, to implement in a more general setting, we simulate the circular data with given network structures to show the performance of the suggested model. For this purpose, we will use the wrapped distributions which are produced by wrapping a distribution around the n-variate torus component-wise (Kurz, Gilitschenski and Hanebeck, 2014). In this representation, let $f(.)$ be the probability distribution function of a multivariate distribution in Euclidean space. The wrapped circular distribution function $g(.)$ is defined as below.

$$g(x, x_2, ..., x_p) = \sum_{k_i \in \mathbb{Z}: i=1,2,...,p} f(y_1 + 2\pi k_1, ..., y_p + 2\pi k_p).$$

After producing data with a known network structure, we will wrap this data around the torus component-wise. This way, we have circular data with a known network structure.

First, we simulate the data from the multivariate normal distribution with a given variance-covariance matrix. In this calculation, the zeros in the covariance matrix imply the independence between corresponding random variables, whereas, the zeros in the inverse of the covariance matrix, namely, the precision matrix, imply conditional independence. In our network structure, we need conditional independence where interactions among variables given all other variables are investigated. Accordingly, we give a precision matrix with zeros for non-edges between random variables and generate the simulated data by using the inverse of this matrix as a covariance matrix (Zhao et al. 2012).

First of all, the variance-covariance matrix has to be symmetric. Therefore, the precision matrix which is the inverse of the variance-covariance matrix has to be symmetric too. Also, the precision matrix has to be positive definite to be invertible. In many of the applications in the literature, to be able to produce a positive definite matrix, authors use the fact that a symmetric matrix which is diagonally dominant and has all diagonals positive, is positive definite. Here, diagonally dominant means the absolute value of the diagonal entries is greater than or equal to the sum of the corresponding rows.

We generate data, using huge.generator() function of the "Huge" R-package, with random graph structure where the probability that a pair of nodes has an edge is constant. In their implementation, the adjacency matrix theta has all diagonal elements equal to 0. Also, each pair of off-diagonal elements of the adjacency matrix are randomly set $\Theta[i,j] = \Theta[j,i] = v$ for $i \neq j$ with a given probability, and 0 otherwise. Instead of giving a random number to each non-zero value, they give the same number, default 0.3, to the magnitude of partial correlations. To obtain a positive definite precision matrix, the smallest eigenvalue of Θ is computed. Then, they set the precision matrix equal to $\Theta + (|\Theta| + 0.1 + u)I$. The covariance matrix is then computed to generate multivariate normal data.

After generating data from the multivariate normal distribution, say $Y_j = (y_{j_1}, ..., y_{j_p})$ for $j = 1, ..., n$, we wrap data around p-torus component-wise in order to transform into the multivariate wrapped normal sample as follows.

$$(x_{j_1}, ..., x_{j_p}) = (y_{j_1} \pmod{2\pi}, ..., y_{j_p} \pmod{2\pi}).$$

We rebuild the network structure by using this circular dataset and the proposed method. Later, the performance of the model is compared for different sample sizes (denoted by n), network sparsities (i.e., the ratio of the number of zeros in the precision matrix, denoted by s) and network sizes (denoted by p).

3.3.2 Results

To examine the strengths and weaknesses of the proposed method, we applied our model to various simulated data coming from the wrapped normal distribution. We generated simulated networks with 10 and 20 nodes to examine the effect of the size of the network on the performance of our model. Furthermore, since our model is based on a regression technique, we expect our model to work only when the sample size is sufficiently large and work better as the sample size increases. To test this hypothesis, we generated data with 10 nodes with sample size 25, 50, 75, 100 and a dataset with 20 nodes with sample size 50, 100 and 200. Finally, to inspect how the density of the networks affects the model performance, we generated datasets with varying densities of 0.15, 0.3 and 0.5. For each network, we did 500 Monte Carlo Runs. As can be seen in Figure 3.1, mean accuracy measures stabilize after 125 runs. In our work, we took 500 runs for more reliable results. The mean performance measures for 500 Monte Carlo runs for all the aforementioned networks can be found in Table 3.1.

In Figure 3.2, you can see an example network with 10 nodes and a sample size of 100 inferred by our method. Our model captured the network with an accuracy of 0.84 and an MCC of 0.613. The model captured 9 out of 12 edges correctly, i.e., has a recall of 0.75, while mistakenly claims 4 edges.

3.3.2.1 Effect of the network size

Sizes of the circular networks vary between fields. For example, Leguey et al. (2019) try to infer the network structure of 7 meteorological stations around Europe using wind directions. Recently, Gottard and Panzera (2021) tried to capture the dependence structure of proteins using dihedral angles where example networks had 8,

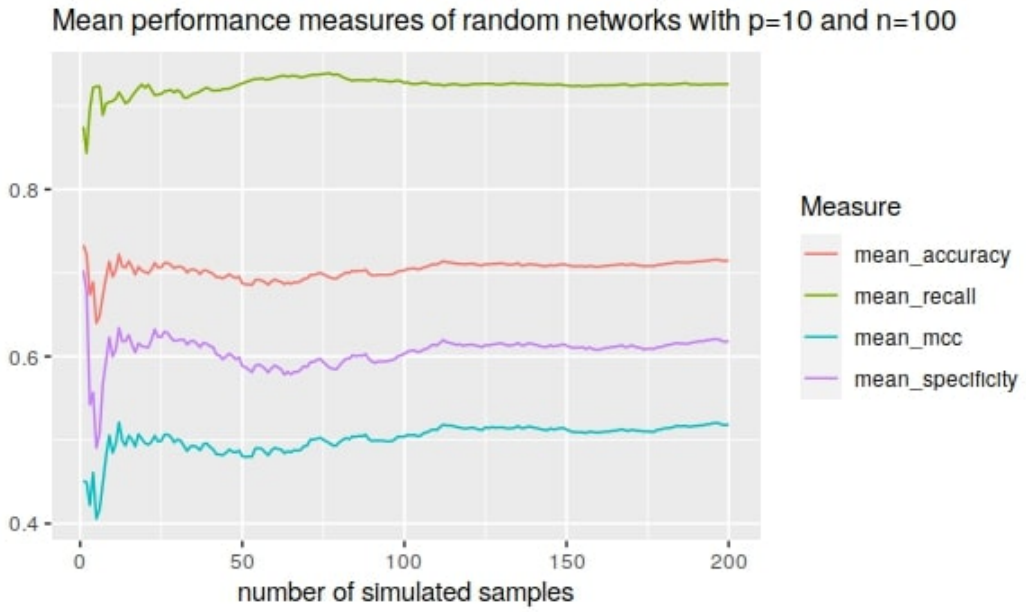

Figure 3.1: Mean performance measures of increasing Monte Carlo runs for random networks of 10 nodes with sample size 100.

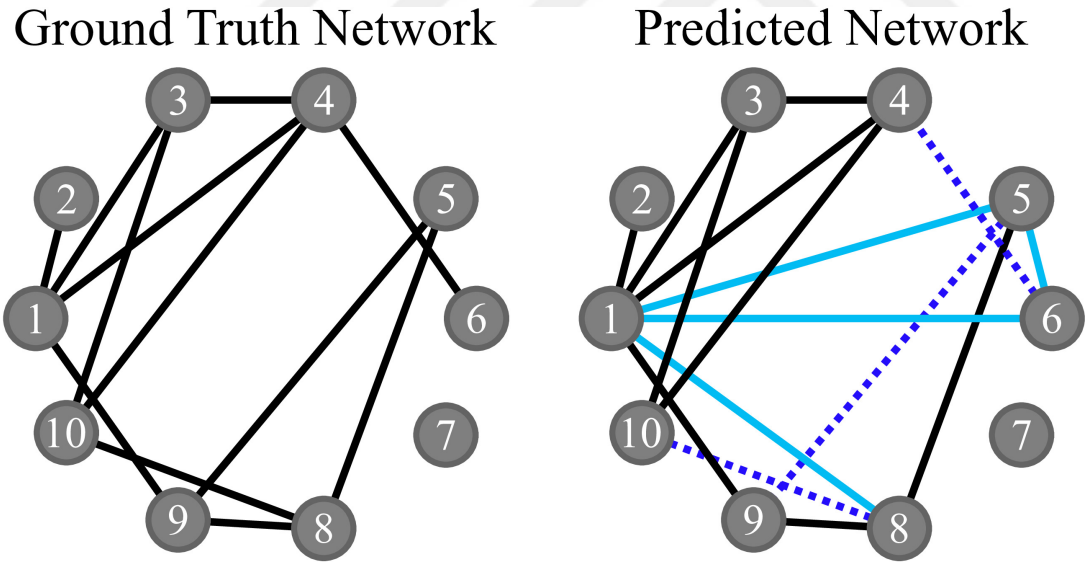

Figure 3.2: A simulated network predicted by the proposed method. Blue edges are false positives and dashed edges are false negatives. Here, the number of nodes is 10 and the sample size is 100. The model predicted with an accuracy of 0.84 and an MCC value of 0.613.

Table 3.1: Mean performance measures for 500 Monte Carlo runs. p represents the number of the nodes, n sample size and s network density.

p	n	s	accuracy	precision	recall	specificity	MCC
10	25	0.15	0.847	0.626	0.184	0.969	0.301
10	50	0.15	0.854	0.615	0.690	0.884	0.553
10	75	0.15	0.805	0.512	0.884	0.790	0.562
10	100	0.15	0.747	0.446	0.967	0.707	0.534
10	25	0.3	0.711	0.662	0.157	0.952	0.214
10	50	0.3	0.762	0.672	0.594	0.836	0.457
10	75	0.3	0.759	0.611	0.832	0.726	0.537
10	100	0.3	0.717	0.549	0.930	0.621	0.522
10	25	0.5	0.541	0.759	0.151	0.935	0.154
10	50	0.5	0.671	0.731	0.589	0.755	0.362
10	75	0.5	0.718	0.703	0.821	0.605	0.448
10	100	0.5	0.708	0.659	0.918	0.489	0.456
20	50	0.15	0.853	0.581	0.212	0.965	0.274
20	100	0.15	0.871	0.570	0.742	0.893	0.573
20	200	0.15	0.819	0.477	0.972	0.792	0.598

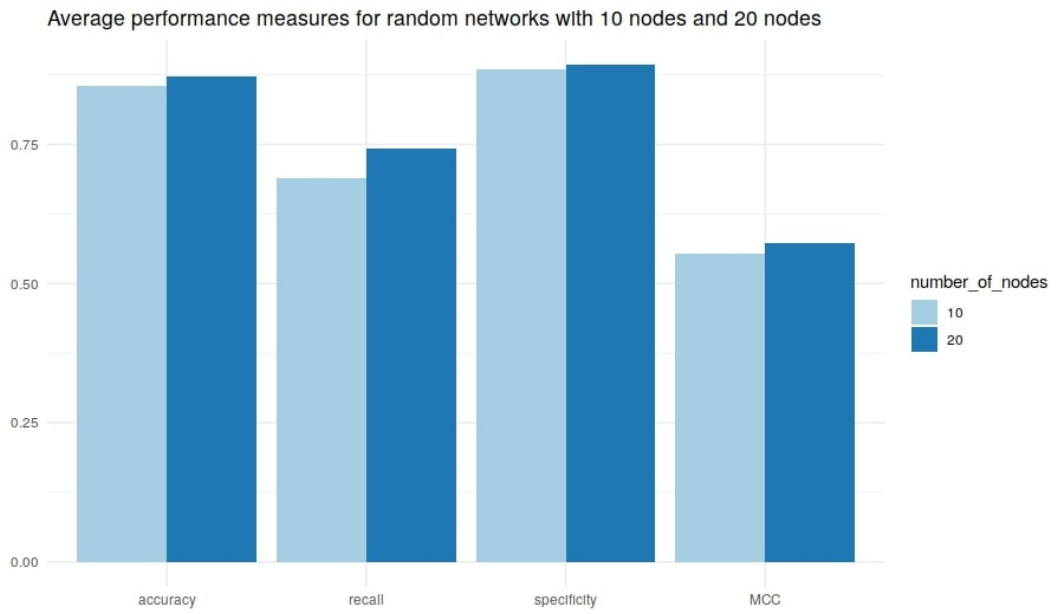

Figure 3.3: Mean performance measures of 500 Monte Carlo runs for random networks with 10 and 20 nodes with density 0.15. The number of samples is taken as 50 and 100, respectively.

9 and 40 nodes. To compare the performance of our model with varying network sizes, we compared networks with number of nodes 10 and 20. To have a meaningful comparison we kept the ratio of the sample size to the number of nodes constant. Therefore, sample sizes are taken as 50 and 100 respectively. As can be seen in Figure 3.3, mean accuracy, recall, specificity and MCC values are similar. However, as the network size gets bigger, the number of parameters in the search space of the optimization algorithm gets higher. This results in convergence problems and time exceeding in the optimization step.

3.3.2.2 Effect of the sample size

Since our model is based on a regression model, we expect the algorithm to work only if the sample size is higher than the number of parameters, and we expect the performance to get better as the sample size increases. To test this claim, we generated

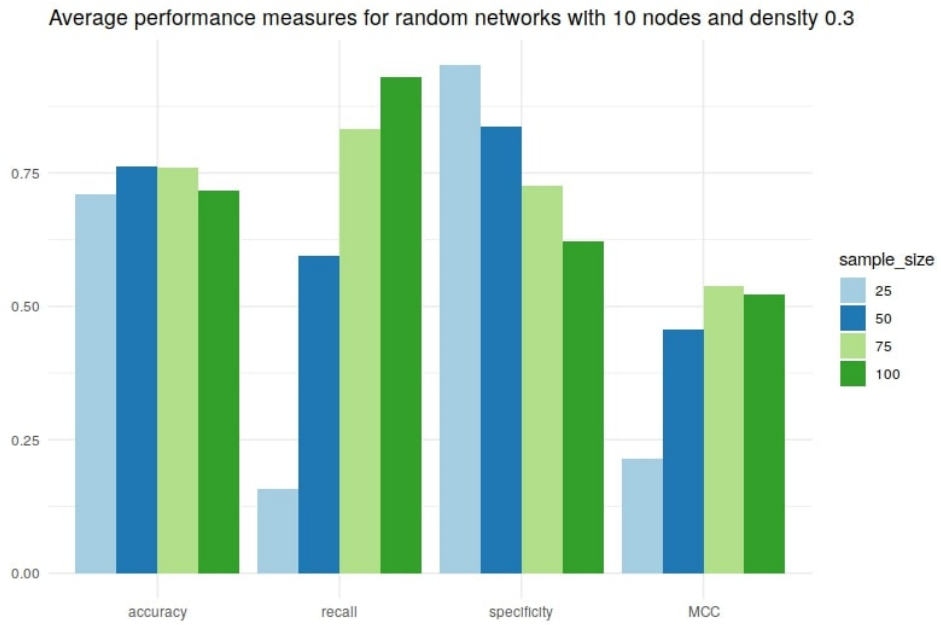

Figure 3.4: Mean performance measures of 500 Monte Carlo runs for random networks with 10 nodes and density 0.3. The number of samples is taken as 25, 50, 75 and 100.

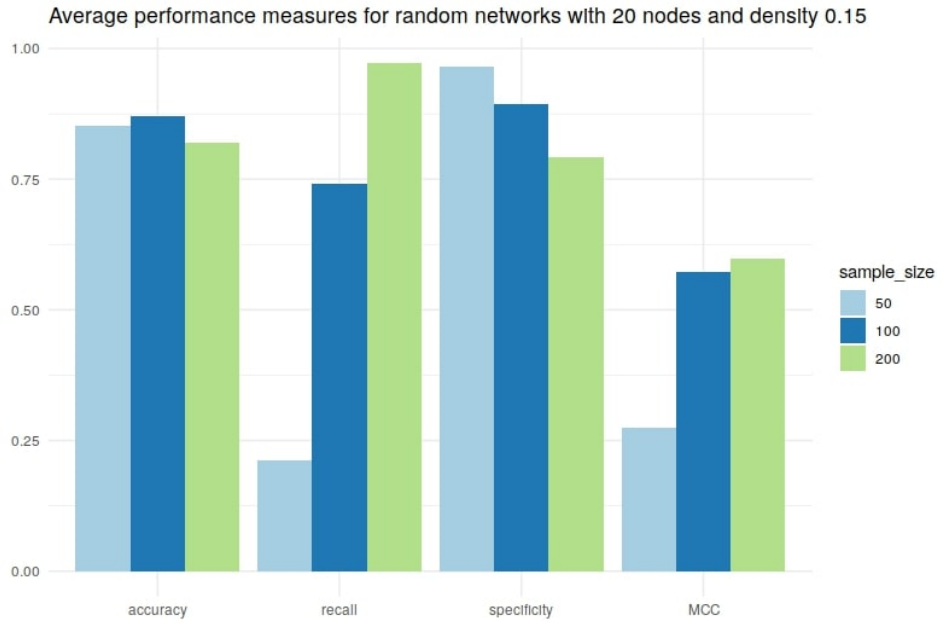

Figure 3.5: Mean performance measures of 500 Monte Carlo runs for random networks with 20 nodes and density 0.15. The number of samples is taken as 50, 100 and 200.

simulated networks with 10 nodes and varying sample sizes of 25, 50, 75 and 100. Additionally, we generated simulated networks with 20 nodes and varying sample sizes of 50, 100 and 200. Figure 3.4 shows that accuracy and MCC values are similar for sample sizes 50, 75 and 100. Although the accuracy value is similar for the case of sample size 25, MCC value is significantly lower, this stems from the highly low recall value. When the sample size is too small, models produce sparser graphs than expected. It seems like recovering when the number of samples is 5 times or higher than the number of nodes. A similar trend can be seen in Figure 3.5 for the networks with 20 nodes.

Another important observation is that recall values get higher and specificity values get lower as the sample size increases. This stems from the fact that an increased sample size will produce extra sensitivity in the hypothesis-testing step. As put by Faber and Fonseca (2014), as the sample size gets too large, "What is insignificant becomes significant.". Therefore, the model has an exaggerated tendency to reject null hypotheses resulting in finding more edges than expected. Thus, we suggest keeping the sample size higher than at least 5 times the network size and less than 10 times to keep the balance of recall and specificity if the underlying distribution is known to be wrapped normal. Accordingly, if the nature of the work requires more importance to be given to the recall value, we can keep the sample size small. On the other hand, if the work requires more importance to be given to the specificity, then we can keep the sample size large.

3.3.2.3 Effect of the network density

We define the density of a network as the probability of having an edge in the network. We wanted to examine if our method is as powerful for the networks while density changes. Figure 3.6 shows the performance measures of networks with size 10 and sample size 50 while sparsity takes values of 0.15, 0.3 and 0.5. Although the model gives reasonable accuracy and MCC values for all networks, performance values decrease as the density increases. We can conclude that model works better for sparse networks than dense ones. In other words, the model has better accuracy in scale-free networks than the random networks.

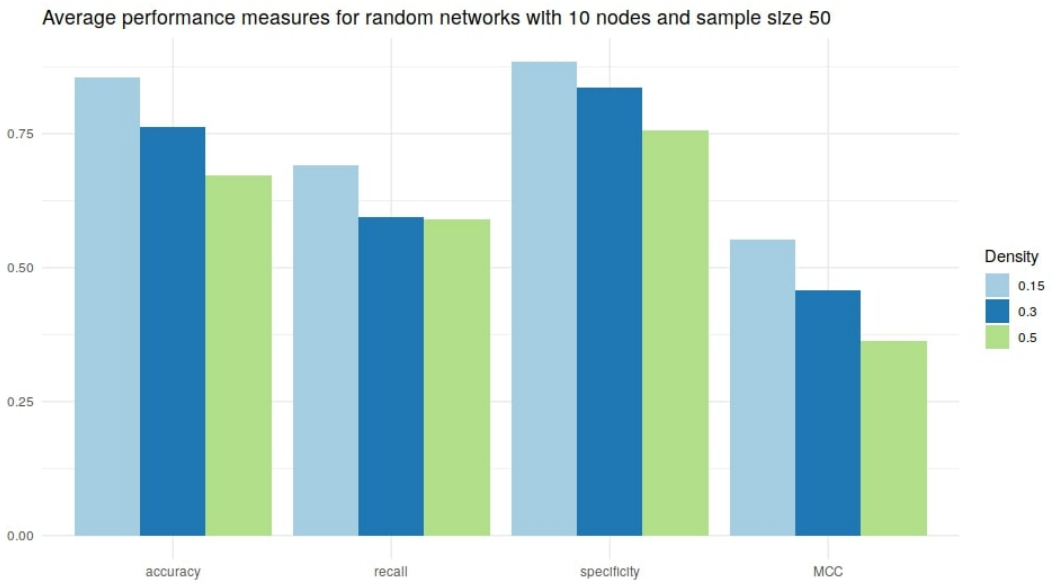

Figure 3.6: Mean performance measures of 500 Monte Carlo runs for random networks with 10 nodes and sample size 50. The density, probability of an edge existing in the graph is taken as 0.15, 0.3 and 0.5.

3.4 Circadian Gene interactions

In specific, apart from the simulations on wrapped normal data, we will implement the proposed method on real data which is circadian gene interaction networks. Circadian genes are expressed in the cells in a recurrent manner with a period of approximately 24 hours (Czeisler et al. 1999, Sukumaran et al. 2010). Therefore, interactions among these genes naturally produce a circular network. Hence, in our study, the proposed method is used after the transformation of the linear microarray data to the circular data. Accordingly, the circadian nature of the data is exploited. Additionally, such an investigation of circadian gene interactions using a model-based approach with the circadian data is the first in this field up to date to the best of our knowledge.

3.4.1 Data

We used microarray data of the gene expressions taken on a 24 or more hours time course with 2-3 hours intervals. The full network of circadian genes is large. For this reason, we pick subsets of the circadian genes and explore interactions within these sub-networks. We estimate the network structure by using this data with the proposed method and compare it with the real network from the literature.

Microarray technology allows us to observe gene expressions of the whole genome at once. Microarray chips consist of microscopic DNA spots, namely probes, each representing a gene, attached to a solid surface. When we put mRNAs taken from a cell on this microchip, they will bind to their complementary parts on the chip. We attach a fluorescent dye to mRNAs before. By measuring the amount of colour emitted by the array we deduct the amount of mRNA in the cell. A more intense signal means a more active gene. This way, at any given time and condition, we can observe high throughput gene expressions in a cell.

There are many public databases including GEO (Edgar et al., 2002) from NCBI (National Center for Biotechnology Information) or ArrayExpress (Athar et al., 2019) from EBI (European Bioinformatics Institute) which store microarray data from different experiments. The organ or tissue this data is obtained from has crucial importance on the upcoming steps of the investigations since circadian rhythms are highly organ-specific. We only model a subset of genes which are circadian in nature for the organ/tissue under investigation. We used CGDB (Circadian Gene DataBase) to detect these genes. CGDB (Li et al., 2016) is an online source containing circadian genes from different organisms including humans with the information of the organism, tissue/cell and PubMed ID as well as the source of the evidence whether it is experimentally identified or predicted. We gather circadian genes by choosing the specific tissue/cell type the data is coming from and experimentally identified evidence sources. Since all variables are circular in our model, we choose only circadian genes of corresponding tissue/cell type from microarray data, to begin with.

After determining the nodes of the network, we need to extract the interactions among them too. However, there are no databases which specifically give interactions among circadian genes only. For that reason, we use the STRING (Szklarczyk 2019) database which is a database of known and predicted protein interactions. We look only at in-

teraction sources of curated databases, co-occurrence and co-expression and use this network as the true network to compare with our model results.

Although circadian gene expressions are circular, microarray data is linear. To reveal circular structure, we transform these periodic expressions into circular data by using sinusoidal waves: expression levels taken at 2-3 hours time intervals can be fit by a sine wave (Hickey 1984) with a period of 24 hours and angles in the sine can be used as the corresponding angle at this point. After doing this for each gene and sample, we slice a time point and use these angles as our circular data.

Since circadian genes are organ-specific, data should come from one organ/tissue. To capture the circadian nature, we need consecutive time points, i.e. measurements every 2 hours and we need data for at least one day. We searched databases GEO from NCBI and ArrayExpress from EBI for this purpose. We found data set "GSE113883" from GEO-NCBI from work titled "TimeSignature: A Universal Method for Robust Detection of Circadian State from Gene Expression" (Braun et al., 2018). Whole blood transcriptional profiles of 11 individuals with intermediate circadian phenotype, were generated by RNA sequencing, using Illumina NextGen 500. Whole blood was collected every 2 hours over 28 hours (15 time-points), yielding a total of 165 samples, of which 153 passed the quality assurance and underwent further analysis. This data contains the whole profile but we need only circadian genes expressed in that specific cell. Hence, afterwards, we gather known circadian genes from the literature. We gather circadian genes from CGDB by choosing blood tissue/cell type from humans with experimentally identified evidence. There are 55 genes in total in this database on this search. After gathering circadian gene information, we need to gather interaction information. For this purpose, we used the STRING (Search Tool for Recurring Instances of Neighbouring Genes) database (Szklarczyk 2019) which is a database of protein interactions. We look only at interaction sources of databases, experiments, co-occurrence and co-expression. We ended up with a network of 55 genes shown in Figure 3.7.

We have sinusoidal fits for each gene and we slice data at a given time point. We used lm() function in R which uses least squares estimation for linear models. Then after pulling points to the sine wave, we use arcsin of the values to transfer them to the

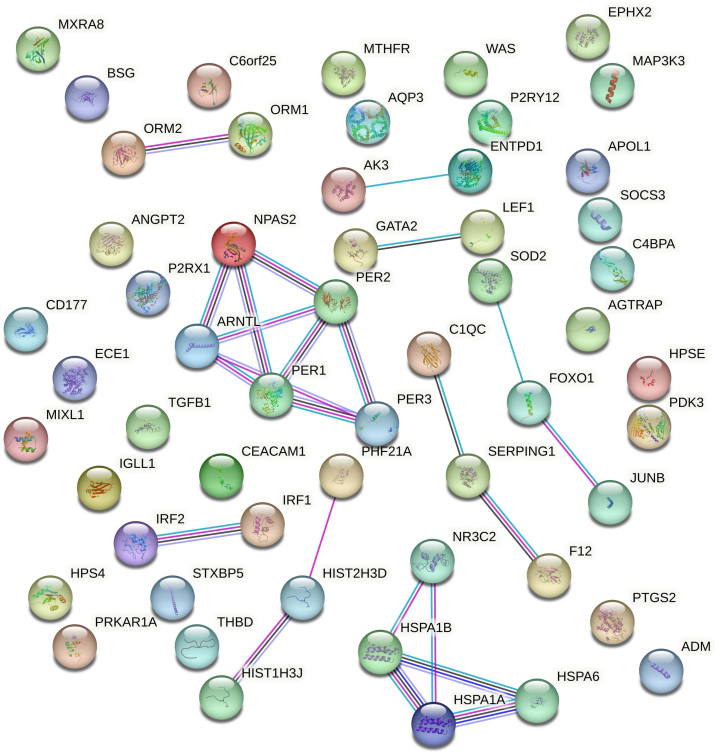

Figure 3.7: Network structure with all circadian proteins from STRING database.

circle. An example of this transforming procedure can be seen in Figure 3.8 and an example of circular gene expression values after transformation can be seen in Table 3.2. We check the p-values for the fits. There can be many reasons that there is no good fit. There can be biological reasons, for that particular time, genes might not be expressed in a circular manner. There can be problems with the data collection. Also, we use sine waves which have a very specific symmetric shape, however, not all circadian genes follow that specific shape. After checking the p-values of the fits, we excluded the samples in which multiple genes gave high p-values.

As can be seen in Table 3.2, some of the genes have multiple repeated entries. This stems from the fact that expressions of these genes are tightly controlled. Some of them are having a peak at a similar time of the day. Therefore, when we make this circular transformation, we end up with some repeated measures. An example of this can be seen in Figure 3.9. Therefore, this transformation results in losing information. This is a general problem with circadian genes. One possible direction to compensate for this information loss can be utilizing the height of the expression together with the

sine-wave forms. Methods with real data and circular data integration can be utilized here.

Nevertheless, for illustration purposes, we applied our method to a sub-network of circadian genes. After extracting 3 samples, we end up with genes with reasonable sine wave fits. Also, these genes have varying expression patterns, therefore, our method is applicable. As can be seen in Figure 3.10, our algorithm caught the existent edge while mistakenly claiming one additional edge.

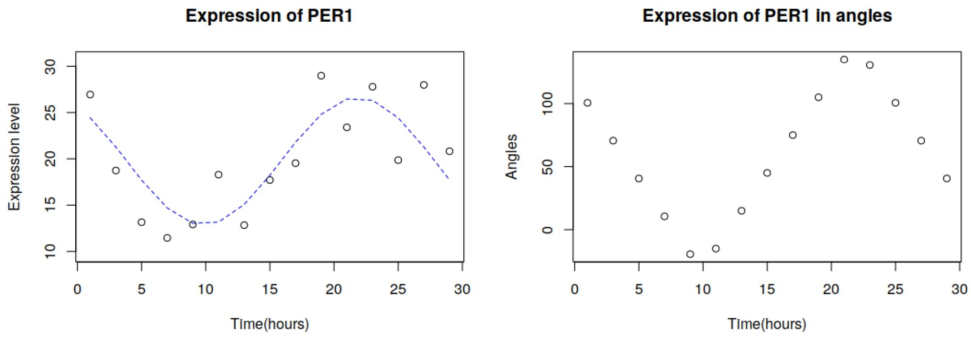

Figure 3.8: Sine wave fitting for the gene PER1.

	sample1	sample2	sample3	sample5	sample7	sample8	sample9	sample10	sample11
F12	43.178	51.128	13.007	65.247	1.828	-68.116	74.816	-72.993	-16.830
SERPING1	37.146	75.000	75.000	75.000	75.000	75.000	75.000	75.000	70.157
ORM1	-25.924	75.000	34.881	75.000	-5.284	75.000	-105.000	53.123	75.000
ORM2	51.277	-55.572	4.142	44.096	23.785	88.484	36.998	47.176	33.505
HPSE	75.000	-3.368	-47.186	56.621	13.157	-5.631	-105.000	75.000	13.009
TGFB1	-4.703	75.000	75.000	75.000	75.000	75.000	75.000	75.000	75.000
C1QC	-60.627	75.000	75.000	75.000	25.213	66.438	-9.486	-38.032	-36.225
PER1	-105.000	-105.000	32.872	-24.309	-105.000	-105.000	55.373	-105.000	3.439
PER2	-34.371	72.032	14.747	75.000	6.711	-22.940	75.000	12.928	-26.055
PER3	40.602	-42.646	-56.661	-97.019	-64.938	-1.422	81.331	-10.272	-61.368
ARNTL	75.000	61.619	51.283	70.619	30.103	75.000	75.000	-43.379	75.000
NPAS2	74.496	-105.000	45.167	75.000	74.048	75.000	75.000	74.979	75.000

Table 3.2: Circular gene expressions after transformation.

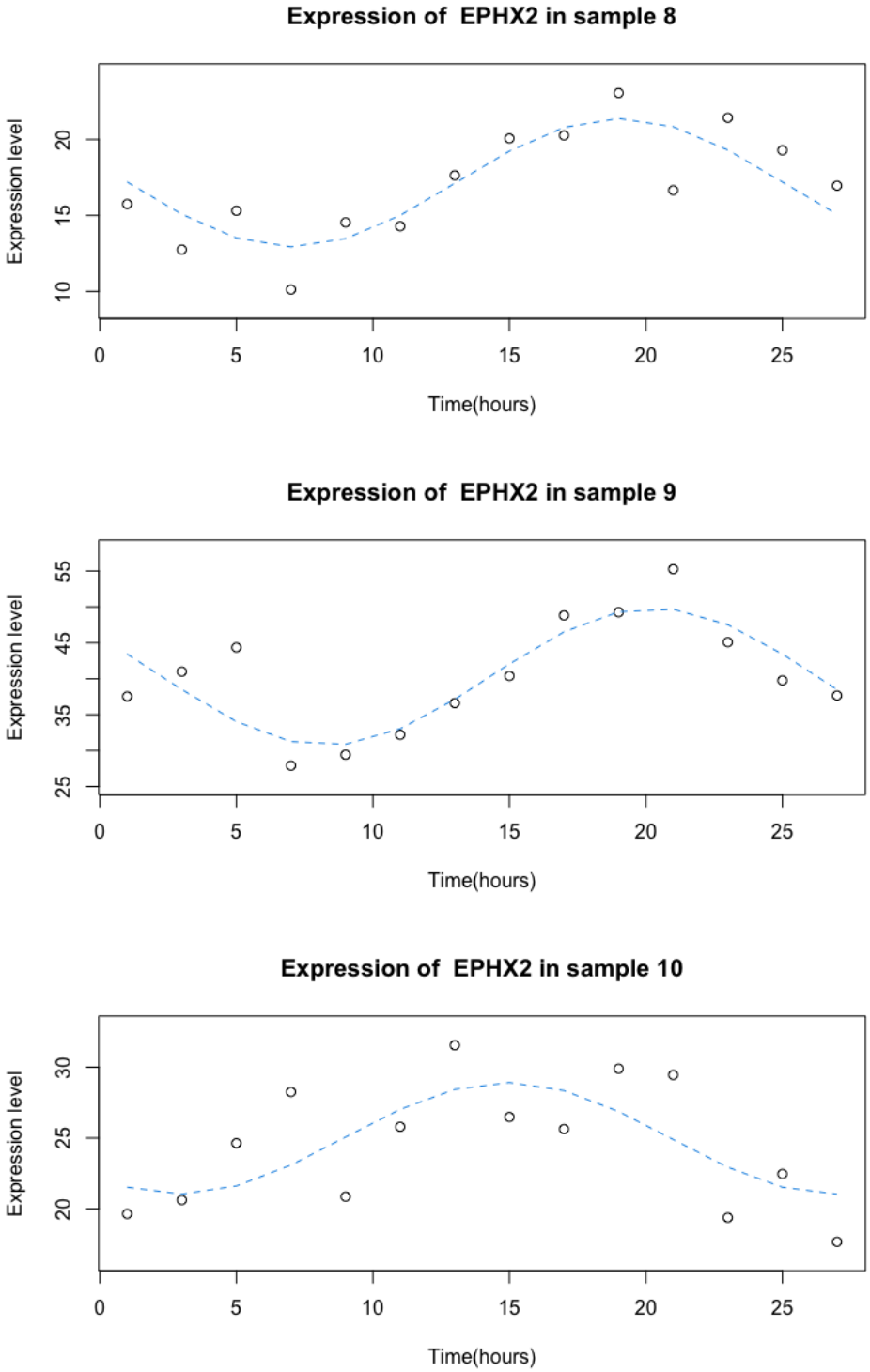

Figure 3.9: Sine wave fitting of the gene EPHX2 for samples 8, 9 and 10.

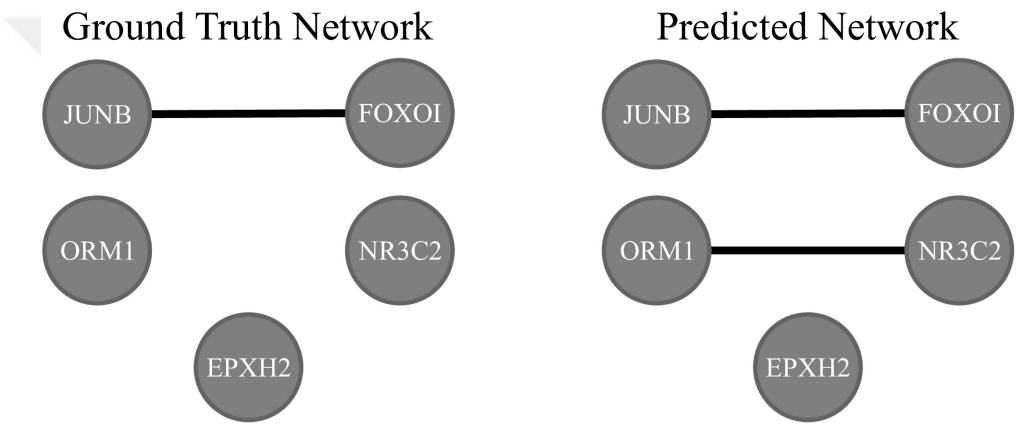

Figure 3.10: An example of circadian sub-network. Our method detected the existent edge correctly while mistakenly claiming one additional edge.

CHAPTER 4

HIDDEN MARKOV MODELS

In this chapter, we will show how can we utilize known network structures for the betterment of the data analyses. To classify HIV-1 protease cleavage sites, we will model amino acid chains with a hidden Markov model. Structure of the network can be seen in Figure 4.1. The hidden Markov model (HMM) is a special case of the probabilistic graphical models where entities are represented by nodes and dependencies by edges between them. In general, the probabilistic graphical models are intractable. However, in HMMs, most of the dependencies are replaced with independence relations. These assumptions help to make the problem tractable while keeping the necessary spatial dependencies. In this chapter, we present HMM model in 4.1 and a toy dataset in 4.2. Afterwards, in 4.3, we show the calculation of likelihood. In section 4.4, we explain the inference procedure and in Section 4.5, we explain the estimation procedure. All these calculations are applied to the toy dataset throughout. Lastly, in Section 4.6, we present the application in the HIV-1 Protease Cleavage dataset and discuss the outputs. This work has been published as a book chapter in "Numerical Solutions of Realistic Nonlinear Phenomena" (Dar, Purutçuoğlu and Purutçuoğlu 2020).

4.1 Model

HMM has a sequence of observations and a sequence of states which produces them. We denote the observation sequence as $O = (O_1, O_2, \ldots, O_T)$ where each observation is an object from the set $o = \{o_1, o_2, \ldots, o_M\}$. Here, T represents the length of the observation and state sequences, and M denotes the number of possible ob-

servations. The hidden states which produce these observations are shown by $S = (S_1, S_2, \ldots, S_T)$ where each state is an object from a set of states $s = \{s_1, s_2, \ldots, s_N\}$. Here, N represents the number of possible states. We also define the following conditional independence assumptions:

- $P(O_k|S_1, \ldots, S_T, O_1, \ldots, O_T) = P(O_k|S_k)$ for any $1 \leq k \leq T$.
- $P(O_i, O_j|S_i, S_j) = P(O_i|S_i, S_j)P(O_j|S_i, S_j) = P(O_i|S_i)P(O_j|S_j)$ for $1 \leq i, j \leq T$.
- $P(S_k|S_1, \ldots, S_{k-1}) = P(S_k|S_{k-1})$ for any $2 \leq k \leq T$, i.e., states form a Markov Chain.

Because of these assumptions, the joint probability of the system can be written as

$$\begin{aligned} P(O_1, ..., O_T, S_1, ..., S_T) &= P(O_1, ..., O_T|S_1, ..., S_T)P(S_1, ..., S_T) \\ &= P(O_1|S_1)P(O_2|S_2)...P(O_T|S_T)P(S_1)P(S_2|S_1) \\ &\quad P(S_3|S_2)...P(S_T|S_{T-1}) \\ &= \left(\prod_{i=1}^{T} P(O_i|S_i)\right) P(S_1) \left(\prod_{i=2}^{T} P(S_i|S_{i-1})\right). \end{aligned} \quad (4.1)$$

Therefore, to define an HMM we only need the probabilities below.

- Transition probabilities: $a_{ij} = P(S_t = j|S_{t-1} = i)$ for $1 \leq i, j \leq N$.
- Emission probabilities: $b_{ij} = P(O_t = j|S_t = i)$ for $1 \leq i \leq N$ and $1 \leq j \leq M$.
- Initial probabilities: $\pi_i = P(S_1 = s_i)$ for $1 \leq i \leq N$.

Furthermore, we can write transition and emission probabilities in a matrix form, say A and B and initial probabilities as a vector Π. Hereby, we denote parameters for HMM as $\lambda = (A, B, \Pi)$. In modelling via HMM, we are interested in basically solving three problems:

- Finding likelihood of an observation sequence given an HMM with parameters λ,

- Finding the most probable state sequence given the model parameters and the observation sequence,
- Estimating the model parameters given sequences of states and observations.

In the following part, we represent each step with detail by using a toy example whose description is also presented.

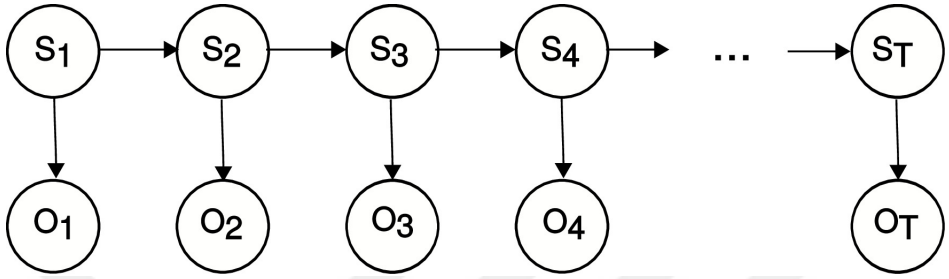

Figure 4.1: Hidden Markov model; nodes represent hidden states and observations while edges indicate the dependencies among them.

4.2 Toy example

For illustrative purposes, we use the following example which is modified from Eisner's paper (Eisner 2002). Let's say the number of ice creams that a person eats every day depends on the weather, which will be taken as either cold or hot. Also, let the number of ice creams she/he eats be from the set $\{1, 2, 3\}$. Here, the weather is the hidden variable where the number of ice creams is observed. Therefore, we have $S = \{H, C\}$ and $O = \{1, 2, 3\}$. Accordingly, the parameters of the model are defined below:

$$A = \begin{bmatrix} P(S_t = H|S_{t-1} = H) & P(S_t = C|S_{t-1} = H) \\ P(S_t = H|S_{t-1} = C) & P(S_t = C|S_{t-1} = C) \end{bmatrix} = \begin{bmatrix} 0.6 & 0.3 \\ 0.4 & 0.5 \end{bmatrix},$$

$$B = \begin{bmatrix} P(O_t = 1|S_t = H) & P(O_t = 2|S_t = H) & P(O_t = 3|S_t = H) \\ P(O_t = 1|S_t = C) & P(O_t = 2|S_t = C) & P(O_t = 3|S_t = C) \end{bmatrix}$$
$$= \begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0.5 & 0.4 & 0.1 \end{bmatrix},$$

$$\Pi = \begin{bmatrix} P(S_1 = H) \\ P(S_1 = C) \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix}.$$

We use this example to elaborate on the calculations of the HMM steps.

4.3 Calculation of likelihood

In some problems, we might be interested in finding the likelihood of an observation sequence given the model parameters while the state sequence is hidden. There are three major approaches to this calculation. We first describe the most natural way to solve this problem, which is the naive approach, and continue with faster counterparts: forward and backward algorithms.

4.3.1 Naive approach

In this computation, we initially find the likelihood given a specific state as shown earlier and then, we sum over all possible states as below.

$$P(O|\lambda) = \sum_{S} P(O, S|\lambda), \tag{4.2}$$

where λ is the model parameter. In Equation (4.2) there are N^T possible states. Therefore, when N and T are large, this approach becomes computationally demanding in the order of $\mathcal{O}(N^T)$ to calculate the likelihood. If we apply this to our toy example, assuming the observations for 3 days to be $O = (2, 1, 3)$, the likelihood of this observation sequence is found by:

$$P(O = (2, 1, 3)|\lambda) = \sum_{S} P(O = (2, 1, 3), S = (s_1, s_2, s_3)|\lambda). \tag{4.3}$$

To find the sum in Equation (4.3), let's first calculate one specific element in the sum, such as $S = (H, H, C)$, by using Equation (4.1). Here, we obtain

$$\begin{aligned} & P(O = (2, 1, 3), S = (H, H, C)|\lambda) \\ = \; & P(O_1|S_1)P(O_2|S_2)P(O_3|S_3)P(S_1)P(S_2|S_1)P(S_3|S_2) \\ = \; & P(2|H)P(1|H)P(3|C)P(H)P(H|H)P(C|H) \\ = \; & 0.4 \times 0.2 \times 0.1 \times 0.8 \times 0.6 \times 0.3. \end{aligned}$$

We have to repeat this calculation $2^3 = 8$ times for different state sequences. Then, we need to compute their sum in order to obtain the likelihood. But, in real-life examples, the number of state sequences can be very high. Thereby, another method which is faster than this naive approach is necessary.

4.3.2 Forward algorithm

The forward algorithm (Kouemou 2011) is a dynamic programming example where we break the problem into sub-problems and use the earlier results in a recursion. In this way, we can solve the inference problem faster than the naive approach. Hereby, the likelihood of the observation sequence and a specific state at the last position of the state sequence given model parameters summed over all possible states are presented as below.

$$P(O|\lambda) = P_\lambda(O) = \sum_{i=1}^{N} P(S_T = s_i, O|\lambda).$$

Thus, in order to find the term in the summation conditional on the model parameters λ, we define

$$\alpha_k(S_k) = P_\lambda(S_k = s_i, O_1, \ldots, O_k). \tag{4.4}$$

The value in the sum is simply equal to $\alpha_T(s_i)$. To be able to find this term in Equation (4.4), we can write it recursively via:

$$\begin{aligned}
\alpha_k(S_k) &= \sum_{S_{k-1}=s_1}^{s_N} P_\lambda(S_k, S_{k-1}, O_1, \ldots, O_k) \\
&= \sum_{S_{k-1}=s_1}^{s_N} P_\lambda(O_k|S_k, S_{k-1}, O_1, \ldots, O_{k-1}) \\
&\quad P_\lambda(S_k|S_{k-1}, O_1, \ldots, O_{k-1}) P_\lambda(S_{k-1}, O_1, \ldots, O_{k-1}) \\
&= \sum_{S_{k-1}=s_1}^{s_N} P_\lambda(O_k|S_k) P_\lambda(S_k|S_{k-1}) P_\lambda(S_{k-1}, O_1, \ldots, O_{k-1}) \\
&= \sum_{S_{k-1}=s_1}^{s_N} b_{s_k,o_k} a_{s_{k-1},s_k} \alpha_{k-1}(S_{k-1})
\end{aligned}$$

for $2 \leq k \leq T$, and for $k = 1$ we have

$$\alpha_1(S_1) = P_\lambda(S_1, O_1) = P_\lambda(S_1) P_\lambda(O_1|S_1) = \Pi(S_1) b_{S_1,O_1}.$$

Here, we recursively find $\alpha_1(S_1)$,...,$\alpha_T(S_T)$ and sum $\alpha_T(S_T)$ over all possible values of S_T to get the likelihood of interest. We complete the forward algorithm with the complexity $\mathcal{O}(N^2T)$. For large values of N and T, this complexity is lower than the complexity of the naive approach. Accordingly, let's see how the algorithm works on our toy example for $O = (2, 1, 3)$:

$$\alpha_1(S_1) = P_\lambda(S_1)P_\lambda(O_1|S_1) = \begin{cases} P_\lambda(H)P_\lambda(2|H) & \text{for } s_1 = H \\ P_\lambda(C)P_\lambda(2|C) & \text{for } s_1 = C \end{cases},$$

$$\begin{aligned} \alpha_2(S_2) &= \sum_{S_1=s_1}^{s_N} P_\lambda(O_2|S_2)P_\lambda(S_2|S_1)\alpha_1(S_1) \\ &= P_\lambda(1|S_2)P_\lambda(S_2|H)\alpha_1(H) + P_\lambda(1|S_2)P_\lambda(S_2|C)\alpha_1(C) \\ &= \begin{cases} P_\lambda(1|H)P_\lambda(H|H)\alpha_1(H) + P_\lambda(1|H)P_\lambda(H|C)\alpha_1(C) & \text{for } s_2 = H \\ P_\lambda(1|C)P_\lambda(C|H)\alpha_1(H) + P_\lambda(1|C)P_\lambda(C|C)\alpha_1(C) & \text{for } s_2 = C \end{cases}, \end{aligned}$$

$$\begin{aligned} \alpha_3(S_3) &= \sum_{S_2=s_1}^{s_N} P_\lambda(O_3|S_3)P_\lambda(S_3|S_2)\alpha_2(S_2) \\ &= P_\lambda(3|S_3)P_\lambda(S_3|H)\alpha_2(H) + P_\lambda(3|S_3)P_\lambda(S_3|C)\alpha_2(C) \\ &= \begin{cases} P_\lambda(3|H)P_\lambda(H|H)\alpha_2(H) + P_\lambda(3|H)P_\lambda(H|C)\alpha_2(C) & \text{for } s_3 = H \\ P_\lambda(3|C)P_\lambda(C|H)\alpha_2(H) + P_\lambda(3|C)P_\lambda(C|C)\alpha_2(C) & \text{for } s_3 = C \end{cases}. \end{aligned}$$

Thus, finally we can obtain

$$P_\lambda(O = (2, 1, 3)) = \alpha_3(H) + \alpha_3(C).$$

4.3.3 Backward algorithm

The Backward algorithm (Kouemou 2011) is similar to the forward algorithm, except for the starting point of the calculation. Hereby, we find the likelihood by the

following expression.

$$\begin{aligned} P_\lambda(O) &= \sum_{i=1}^{N} P_\lambda(S_1 = s_i, O) \\ &= \sum_{i=1}^{N} P_\lambda(S_1 = s_i) P_\lambda(O_1 | O_2, \ldots, O_T, S_1 = s_i) P_\lambda(O_2, \ldots, O_T | S_1 = s_i) \\ &= \sum_{i=1}^{N} P_\lambda(S_1 = s_i) P_\lambda(O_1 | S_1 = s_i) P_\lambda(O_2, \ldots, O_T | S_1 = s_i) \\ &= \sum_{i=1}^{N} \Pi(s_i) b_{s_i, O_1} P_\lambda(O_2, \ldots, O_T | S_1 = s_i). \end{aligned}$$

To obtain the solution, we need to obtain the last term in the sum and we compute it by recursively using the following definition.

$$\begin{aligned} \beta_k(S_k) &= P_\lambda(O_{k+1}, \ldots, O_N | S_k) \\ &= \sum_{S_{k+1}=s_1}^{s_N} P_\lambda(O_{k+1}, \ldots, O_T, S_{k+1} | S_k) \\ &= \sum_{S_{k+1}=s_1}^{s_N} P_\lambda(O_{k+2}, \ldots, O_T | S_{k+1}, S_k, O_{K+1}) P_\lambda(O_{k+1} | S_{k+1}, S_k) P_\lambda(S_{k+1} | S_k) \\ &= \sum_{S_{k+1}=s_1}^{s_N} P_\lambda(O_{k+2}, \ldots, O_T | S_{k+1}) P_\lambda(O_{k+1} | S_{k+1}) P_\lambda(S_{k+1} | S_k) \\ &= \sum_{S_{k+1}=s_1}^{s_N} \beta_{k+1}(S_{k+1}) b_{S_{k+1}, O_{k+1}} a_{S_k, S_{k+1}} \end{aligned}$$

for $1 \leq k \leq N-1$. For $\beta_T(S_T)$, we cannot use the above definition since it involves O_{N+1} which does not exist. So, if we use our recursion formula for $k = T-1$,

$$\begin{aligned} \beta_{T-1}(S_{T-1}) &= \sum_{S_T=s_1}^{s_N} P_\lambda(O_T, S_T | S_{T-1}) \\ &= \sum_{S_T=s_1}^{s_N} \beta_T(S_T) P_\lambda(O_T | S_T) P_\lambda(S_T | S_{T-1}). \end{aligned}$$

But $P_\lambda(O_T, S_T | S_{T-1})$ can be also written as,

$$\begin{aligned} P_\lambda(O_T, S_T | S_{T-1}) &= P_\lambda(O_T | S_T, S_{T-1}) P_\lambda(S_T | S_{T-1}) \\ &= P_\lambda(O_T | S_T) P_\lambda(S_T | S_{T-1}). \end{aligned} \tag{4.5}$$

Therefore, for Equation (4.5) to hold, $\beta_T(S_T) = 1$. Now by using Equation (4.1) and the definition of β, we can get

$$P_\lambda(O) = \sum_{i=1}^{N} \Pi(s_i) b_{s_i,O_1} \beta_1(S_1 = s_i).$$

4.4 Viterbi algorithm: Inference of the most probable path

The Viterbi algorithm (Kouemou 2011) is a recursive algorithm that is used to find the most probable sequence, also called *path*, given the observation sequence and parameters. In the calculation, after initialization of the state, at each step, we use the earlier paths which we find. More formally, our aim is to find

$$S^* = \arg\max_S P(S|O).$$

Note that;
If $f(a) \geq 0$ for all a and $g(a,b) \geq 0$ for all a, b, we have,

$$\max_{a,b} f(a)g(a,b) = \max_a \left\{ f(a) \max_b g(a,b) \right\}, \tag{4.6}$$

and we have

$$\arg\max_S P(S|O) = \arg\max_S P(S,O)$$

since $P(O)$ does not contain any element from hidden states. Now let us define the function μ and the recursion by using Equation (4.6) as below.

$$\begin{aligned}
\mu_k(S_k) &= \max_{S_1,\ldots,S_{k-1}} P(S_1,\ldots,S_k,O_1,\ldots,O_k) \\
&= \max_{S_1,\ldots,S_{k-1}} P(O_k|S_k)P(S_k|S_{k-1})P(S_1,\ldots,S_{k-1},O_1,\ldots,O_{k-1}) \\
&= \max_{S_{k-1}} P(O_k|S_k)P(S_k|S_{k-1}) \max_{S_1,\ldots,S_{k-2}} P(S_1,\ldots,S_{k-1},O_1,\ldots,O_{k-1}) \\
&= \max_{S_{k-1}} P(O_k|S_k)P(S_k|S_{k-1})\mu_{k-1}(S_{k-1})
\end{aligned}$$

for $2 \leq k \leq T$, and by definition $\mu_1(S_1) = P(S_1, O_1) = P(S_1)P(O_1|S_1)$. So, we find the sequence of the state which leads to

$$\max_{S_T} \mu_T(S_T) = \max_{S_1,\ldots,S_T} P(S_1,\ldots,S_T,O_1,\ldots,O_T).$$

For this purpose, at each iteration, we note the most probable state and the path which satisfies these conditions. We can explain the application of this search process via

our toy example. Let $O = (2, 1, 3)$, then,

$$\mu_1 S_1 = P(S_1)P(O_1|S_1) = \begin{cases} P(H)P(2|H) & \text{for } s_1 = H \\ P(C)P(2|C) & \text{for } s_1 = C \end{cases} = \begin{cases} \mathbf{0.32} & \text{for } s_1 = H \\ 0.08 & \text{for } s_1 = C \end{cases}. \tag{4.7}$$

Since the maximum is achieved when $S_1 = H$, we have $\arg\max P(S_1|O_1) = H$. By using Equation (4.7), we calculate μ as follows.

$$\begin{aligned} \mu_2(S_2) &= \max_{S_1} P(O_2|S_2)P(S_2|S_1)\mu_1(S_1) \qquad (4.8) \\ &= \begin{cases} P(1|H)P(H|H)\mu_1(H) \text{ for } s_1 = H, s_2 = H \\ P(1|H)P(H|C)\mu_1(C) \text{ for } s_1 = C, s_2 = H \\ P(1|C)P(C|H)\mu_1(H) \text{ for } s_1 = H, s_2 = C \\ P(1|C)P(C|C)\mu_1(C) \text{ for } s_1 = C, s_2 = C \end{cases} \\ &= \begin{cases} \mathbf{0.0384} \text{ for } s_1 = H, s_2 = H \\ 0.0016 \text{ for } s_1 = C, s_2 = H \\ \mathbf{0.0480} \text{ for } s_1 = H, s_2 = C \\ 0.0050 \text{ for } s_1 = C, s_2 = C \end{cases}. \end{aligned}$$

Therefore, we obtain the most probable paths as $S_1 = H, S_2 = H$ and $S_1 = H, S_2 = C$. If we continue the iteration in the same way via Equation (4.8),

$$\begin{aligned} \mu_3(S_3) &= \max_{S_2} P(O_3|S_3)P(S_3|S_2)\mu_2(S_2) \qquad (4.9) \\ &= \begin{cases} P(3|H)P(H|H)\mu_2(H) \text{ for } s_2 = H, s_3 = H \\ P(3|H)P(H|C)\mu_2(C) \text{ for } s_2 = C, s_3 = H \\ P(3|C)P(C|H)\mu_2(H) \text{ for } s_2 = H, s_3 = C \\ P(3|C)P(C|C)\mu_2(C) \text{ for } s_2 = C, s_3 = C \end{cases} \\ &= \begin{cases} \mathbf{0.009216} & \text{for } s_1 = H, s_2 = H \\ 0.007680 & \text{for } s_1 = C, s_2 = H \\ 0.001152 & \text{for } s_1 = H, s_2 = C \\ \mathbf{0.002400} & \text{for } s_1 = C, s_2 = C \end{cases} \end{aligned}$$

Hence, we get the most probable paths as $S_1 = H, S_2 = H, S_3 = C$ and $S_1 = H, S_2 = H, S_3 = H$. Finally, by using the results in Equation (4.9), we reach

$$\max_{S_3} \mu_3(S_3) = \max_{S_3} \{\mu_3(H), \mu_3(C)\} = \max\{0.009216, 0.002400\} = 0.009216.$$

As a result, we conclude that the path, which presents $S_1 = H, S_2 = H, S_3 = H$, is the most probable path if the sequence of observations is $O = (2, 1, 3)$.

4.5 Baum-Welch algorithm: Estimating the model parameters

The Baum-Welch forward-backward method (Durbin et al. 1998) is an iterative algorithm which is also a special case of the expectation-maximization approach. Here, we start the calculation with an initial guess of the parameters and by using data in hand, we aim to make better estimates for the model parameters λ iteratively until λ converges. In these computations, we use the following expression for the estimator of the transition probability between the ith and the jth variables, i.e., states.

$$\hat{a}_{ij} = \frac{\text{Expected number of transitions from } i \text{ to } j}{\text{Expected number of transitions from } i}.$$

To find these expectations, we apply the following equation.

$$\begin{aligned} \xi_t(i,j) &= P(S_t = i, S_{t+1} = j|O, \lambda), \\ &= \frac{P(S_t = i, S_{t+1} = j, O|\lambda)}{P(O|\lambda)}, \\ &= \frac{\alpha_t(S_t = s_i) a_{ij} b_{S_{t+1}=s_j, o_{t+1}} \beta_{t+1}(S_{t+1} = s_j)}{\sum_{j=1}^{N} \alpha_t(S_t = s_j)\beta_t(S_t = s_j)}. \end{aligned} \tag{4.10}$$

In Equation (4.10), we can find the denominator by using only the forward or only the backward algorithm too. Then, by computing the function ξ, we can write the estimator for a_{ij} as

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i,k)}.$$

Similarly, to estimate the emission probability matrix B we can use,

$$\hat{b}_j(o_k) = \frac{\text{Expected number of times being in state } s_j \text{ and observing } o_k}{\text{Expected number of times being in state } s_j}.$$

Accordingly, the meaning of γ_t is

$$\gamma_t(j) = P(S_t = j|O, \lambda) = \frac{P(S_t = j, O|\lambda)}{P(O|\lambda)} = \frac{\alpha_t(S_t = s_j)\beta_t(S_t = s_j)}{\sum_{j=1}^{N} \alpha_t(S_t = s_j)\beta_t S_t = s_j}.$$

Finally, we can write our estimate for b_j as follows.

$$\hat{b}_j(o_k) = \frac{\sum_{t=1 \text{ st } O_k = o_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}.$$

Also, we can state the estimate of the initial probability π as

$$\hat{\pi}_i = \gamma_1(i).$$

4.6 Application on HIV-1 Protease Cleavage Sites

4.6.1 Data description

In our work, we use the HIV-1 protease cleavage 746 dataset (Rögnvaldsson, You and Garwicz 2015). The data contain the lists of octamers (8 amino acids) and a flag depending on whether the HIV-1 protease will cleave in the central position (between amino acids 4 and 5). There are 401 cleaved and 345 non-cleaved octamers. We also use the physicochemical properties of amino acids from the AAIndex database (Nakai, Kidera and Kanehisa 1988). In this database, there are 544 properties taken as continuous variables for each amino acid. We discard 14 of them since they contain null values.

4.6.2 Creation of states

In modelling our data via HMM, there are 8-bit observation sequences where each observation is from a set of 20 standard amino acids, namely, A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y and V. Each observation has a hidden state behind it, which we form by using physicochemical properties of amino acids. Furthermore, we accept that if we replace an amino acid of a cleaved sequence with another amino acid having similar properties, it is more likely that the new sequence will also be a cleaved sequence. Therefore, we group amino acids according to the similarities based on physicochemical properties and use that information as their hidden states in the model.

After discarding features with null values, we have 530 features that can be taken for the analyses. On the other hand, when a clustering algorithm is used with a large

number of features, typically, it can perform poorly due to outliers or highly correlated variables. Therefore, in our calculation, we implement some feature selection methods to decrease the number of features. For this purpose, initially, we group the features via the same AAIndex data by treating features as instances. Here, we use the k-means (Hartigan and Wong 1979), k-medoids (Park and Jun 2009) and hierarchical clustering (Murtagh 1985) techniques. In the k-means and hierarchical clustering approaches, we form a subset of features by choosing a variable randomly from each cluster. On the other side, in the k-medoids, we select the cluster medoids as the cluster representatives. Furthermore, we try different numbers of feature subsets that change from 30 to 60 with an increment of 5 in order to detect the optimal number of subsets. Additionally, we construct the model without performing any feature selection and compare it with the models with the feature selection in order to observe the effect of these clusterings in modelling.

Thereby, by using the underlying subsets of features, we group amino acids to create the states. Here, we accept that an amino acid can share many different properties with multiple groups. Thus, we prefer fuzzy clustering (Bezdek 1981), rather than classical clustering approaches, in our analyses. In this way, an amino acid can belong to more than one cluster with a membership degree between 0 and 1, and the sum of the membership degrees adding to 1. Among alternative fuzzy approaches, we select the most well-known ones, namely, the fuzzy k-means (Bezdek 1981), Gustafson and Kessel-like fuzzy k-means (Gustafson and Kessel 1978) and the fuzzy k-medoids (Krishnapuram et al. 2001). The fuzzy k-means is similar to usual the k-means method, where the Gustafson Kessel-like fuzzy k-means considers non-spherical clusters too. In the fuzzy k-medoids, the medoids are taken as cluster representatives instead of artificial means. Finally, we assign amino acids to a state if the membership degree is greater than 0.1.

On the other hand, in our calculations, since there are 20 amino acids to cluster, the number of clusters cannot be more than 10. Whereas, we see that when the number of clusters is less than 5, too much information is lost. Therefore, we try and compare the number of states from 5 to 10 in order to detect the optimal number. Also, we standardize the features before clustering to avoid any bias caused by the variance of the features. In Table 4.1, we present 9 states created by the fuzzy k-medoids method using 60 features chosen by the hierarchical clustering.

Table 4.1: States created by the fuzzy k-medoids approach with 9 number of states using 60 features which are determined via the hierarchical clustering.

States	**Amino acids**
1	R, N, D, C, Q, M, P, W, Y
2	C, I, M, F, P, W, Y
3	R, N, D, C, G, P, S, T, Y
4	A, N, C, G, M, P, S
5	I, L, M, W
6	C, G, I, M, V
7	R, N, K
8	D, E, P
9	N, C, H, W, Y

4.6.3 Initialization of the EM algorithm

On the other side, while doing the inference on emission, starting and transition probabilities, we use the Baum-Welch EM algorithm, which can converge to a local maximum instead of the global maximum (Durbin et al. 1998). Therefore, we give a clever starting point to the algorithm in order to increase the probability of reaching the global maximum in our calculation. Hence, we use the data in hand to make a good prediction in the following way:

1. Calculation of initial probabilities: To calculate the starting probability of a state, we count all the sequences in the training data which start with amino acids that this state includes. Then, for all states, we divide them into the sum of these counts to turn these counts into probabilities. For example, let's say we have 15 sequences in the training data where 5 of them starting with A, 2 starting with S and 8 starting with V. State 3 contains A, therefore its count is counted by 2, State 4 contains both A and S, therefore its count is set to $2 + 5$, State 6 contains V, hence, its count equals to 8. Thus, the probability of the first state being State 3 is found as $2/17$, the first state being State 4 is equated to $7/17$ and the first state being State 6 is computed as $8/17$ while all other values

in the vector Π being 0.

2. Calculation of emission probabilities: To estimate the probability of observing an amino acid given a state, we apply the following procedure. With the 0.9 probability, we observe one of the amino acids that this state includes, and with the 0.1 probability, other amino acids that this state does not include. As an example, State 1 includes 9 amino acids where the probability 0.9 is equally distributed among them, each having probability $0.9/9$, and the rest of the amino acids have the 0.1 probability equally distributed among them, each having the probability $0.1/11$.

3. Calculation of transition probabilities: We know the corresponding states for each amino acid of our sequences in the data. For example, Table 4.2 shows the corresponding states for amino acids of the sequence AIMALKMR. For example, as seen in Table 4.2, there are 2 transitions from State 5 to State 5 and there is a 1 transition from State 5 to State 2 given only the sequence AIMALKMR. In this way, we count all transitions coming from all sequences in the training set. Afterwards, for each state, we sum all transitions from this state to all states including itself and divide counts of all transitions from this state to this number. Accordingly, we can turn it into a probability distribution. In the case of the sum is 0, the probability of this row is taken equally distributed as $1/N$ for each state.

Sequence	A	I	M	A	L	K	M	R
States	4	2 5 6	1 2 4 5 6	4	5	7	1 2 4 5 6	1 3 7

Table 4.2: Corresponding states of the amino acids in the sequence AIMALKMR.

Figures 4.2 and 4.3 show the count matrix and the transition matrix produced by using only the sequence AIMALKMR.

	$State1$	$State2$	$State3$	$State4$	$State5$	$State6$	$State7$	$State8$	$State9$
$State1$	1	0	1	1	0	0	1	0	0
$State2$	2	1	1	1	1	1	1	0	0
$State3$	0	0	0	0	0	0	0	0	0
$State4$	1	1	1	1	2	1	1	0	0
$State5$	2	1	1	2	1	1	2	0	0
$State6$	2	1	1	2	1	1	1	0	0
$State7$	1	1	0	1	1	1	0	0	0
$State8$	0	0	0	0	0	0	0	0	0
$State9$	0	0	0	0	0	0	0	0	0

Figure 4.2: Counts of the transitions between states produced from the sequence AIMALKMR.

	$State1$	$State2$	$State3$	$State4$	$State5$	$State6$	$State7$	$State8$	$State9$
$State1$	$1/4$	0	$1/4$	$1/4$	0	0	$1/4$	0	0
$State2$	$2/8$	$1/8$	$1/8$	$1/8$	$1/8$	$1/8$	$1/8$	0	0
$State3$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$
$State4$	$1/8$	$1/8$	$1/8$	$1/8$	$2/8$	$1/8$	$1/8$	0	0
$State5$	$2/10$	$1/10$	$1/10$	$2/10$	$1/10$	$1/10$	$2/10$	0	0
$State6$	$2/9$	$1/9$	$1/9$	$2/9$	$1/9$	$1/9$	$1/9$	0	0
$State7$	$1/5$	$1/5$	0	$1/5$	$1/5$	$1/5$	0	0	0
$State8$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$
$State9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/9$

Figure 4.3: Transition matrix produced from the sequence AIMALKMR.

4.6.4 Modeling the data via HMM

In our calculation, we initially split cleaved and non-cleaved data into 90% of training and 10% of the test data. Then, we only use the test data after finding the optimal model parameters through training. Using initializations for the model parameters, we apply the Baum-Welch EM algorithm with the 1000 maximum numbers of iterations, and the convergence criteria for the change of the log-likelihood equal to 0.001. Furthermore, in all analyses, we conduct the R programming language and we utilize the `aphid` R package for the calculation.

Moreover, the optimum values of the hyper-parameters are selected by using the 10-fold cross-validation on the training data. Cross-validation is used to reduce the bias that stems from the random selection of data. Accordingly, the training data are divided into 10 folds and 9 of them are used for the training data as well as the last one is used for the validation data. Finally, we repeat this process 10 times until we utilize all 10 folds as the validation data.

To classify the sequence as cleaved or non-cleaved, two separate HMMs are trained on the cleaved and non-cleaved datasets, respectively. We declare these sequences as cleaved if the likelihood of belonging to cleaved HMM is greater than the non-cleaved HMM and vice versa. This way, we calculate the false positive (FP), false negative (FN), true positive (TP) and true negative (TN) values. To measure the quality of our classification, we compute the precision(pre), recall (rec), accuracy (acc), Matthews correlation coefficient (MCC) and the F-measure (F). The formulas of these measures are also represented below:

$$
\begin{aligned}
\text{Precision} &= \frac{\text{TP}}{\text{TP+FP}}, \\
\text{Recall} &= \frac{\text{TP}}{\text{TP+FN}}, \\
\text{Accuracy} &= \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}, \\
\text{MCC} &= \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP+FP}) \times (\text{TP+FN}) \times (\text{TN+FP}) \times (\text{TN+FN})}}, \\
\text{F-measure} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}.
\end{aligned}
$$

Lastly, after deciding on the final model, to avoid over-optimism caused by overfitting, we declare results by using the test data whose final model has not been seen yet.

4.6.5 Results

In this section, for brevity, we refer to the fuzzy k-means method as fkm, Gustafson-Kessel like fuzzy k-means as GKfkm and the fuzzy k-medoids approach as fkmed.

1. Effect of the number of states when other hyperparameters are fixed: The number of states do not have a linear effect on the accuracy values when other parameters are fixed. Figure 4.4 shows the accuracy as a function of the number of states and the feature selection methods with the number of features used as 60 (Figure 4.4 (a-c)) or no feature selection used (Figure 4.4 (d)). As seen in Figure 4.4, when GKfkm is used, the accuracy increases with the number of states except in the case when the number of states changes from 5 to 6. In that case, there is a slight decrease. On the other hand, there is no common pattern when other techniques are used. In our analyses, in total, we perform 3 feature selection techniques, 7 different number of features and 3 state selection techniques, which makes a total of $3 \times 7 \times 3 = 63$ possible cases. This number becomes 66 when we include cases when we do not implement any feature selection. Out of these 66 cases, 40 of them give the best accuracy when the number of states is 10, 14 of them give the best accuracy when the number of states is 9, 5 of them give the optimal accuracy when the number of states is 8, followed by the number of states 5, 6 and 7, respectively. Hence, we conclude that the large numbers of states produce more accurate results.

2. Effect of the number of features on the accuracy: Figure 4.5, 4.6 and 4.7 show the accuracy as a function of the number of features for different feature selection and state selection methods. As seen in the figures, the effect of the number of features highly depends on the methods used. Moreover, the change in the number of features does not have an effect on the accuracy when GKfkm is applied. Also, fkm is very robust to the changes in the number of features only when the hierarchical feature selection method is performed. Finally, we observe that there is no common pattern for the other methods in the analyses.

3. Effect of the feature selection methods on the accuracy: As seen in Figure 4.8, when the fkm state selection method is implemented, the k-medoids method gives the best results almost all the time except in a few cases. Whereas the

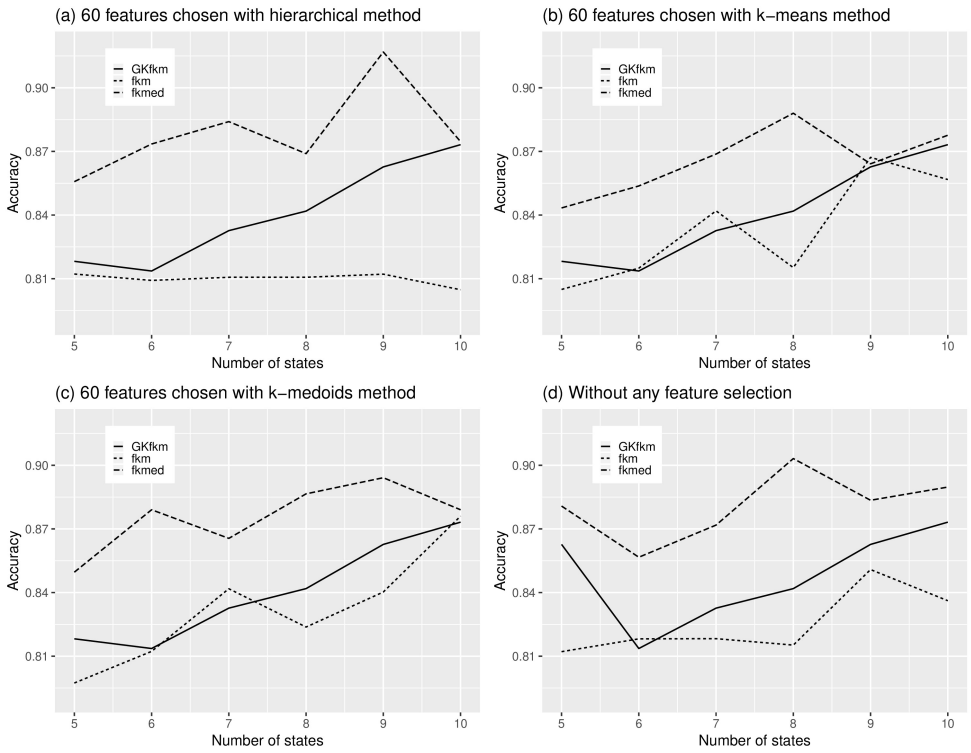

Figure 4.4: Effect of the number of states on the accuracy values.

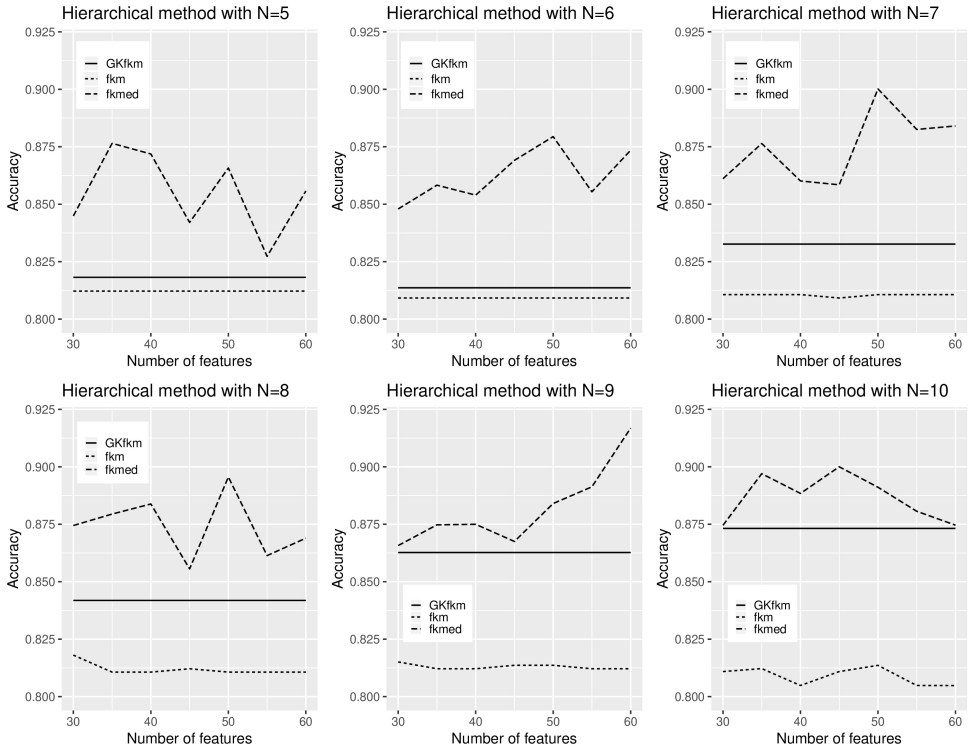

Figure 4.5: Accuracy values for the hierarchical feature selection.

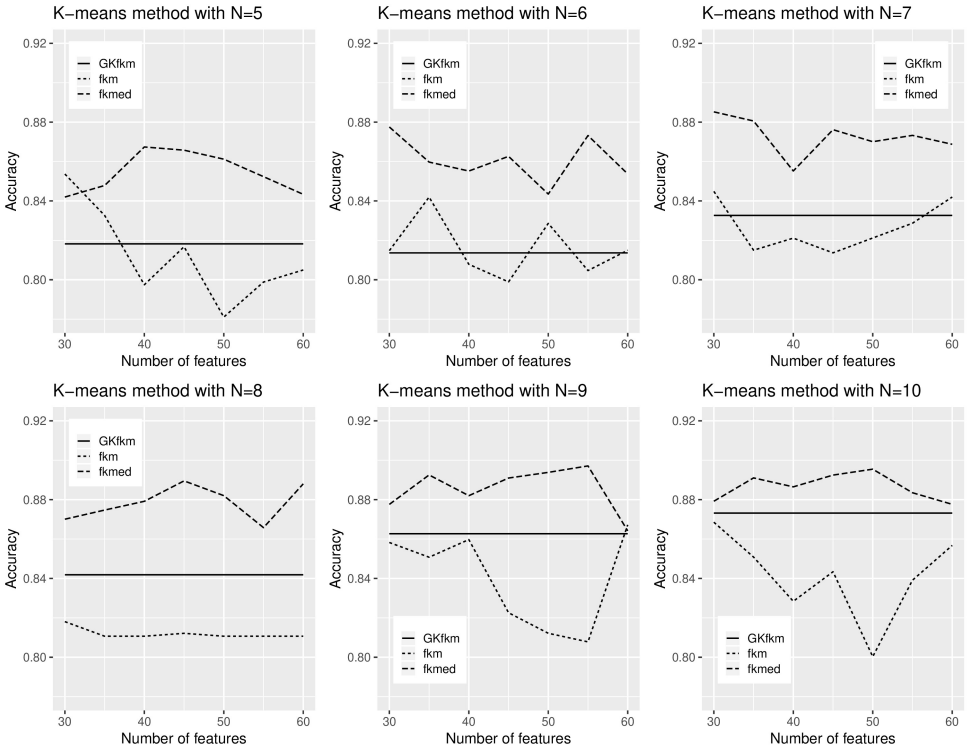

Figure 4.6: Accuracy values for the k-means feature selection.

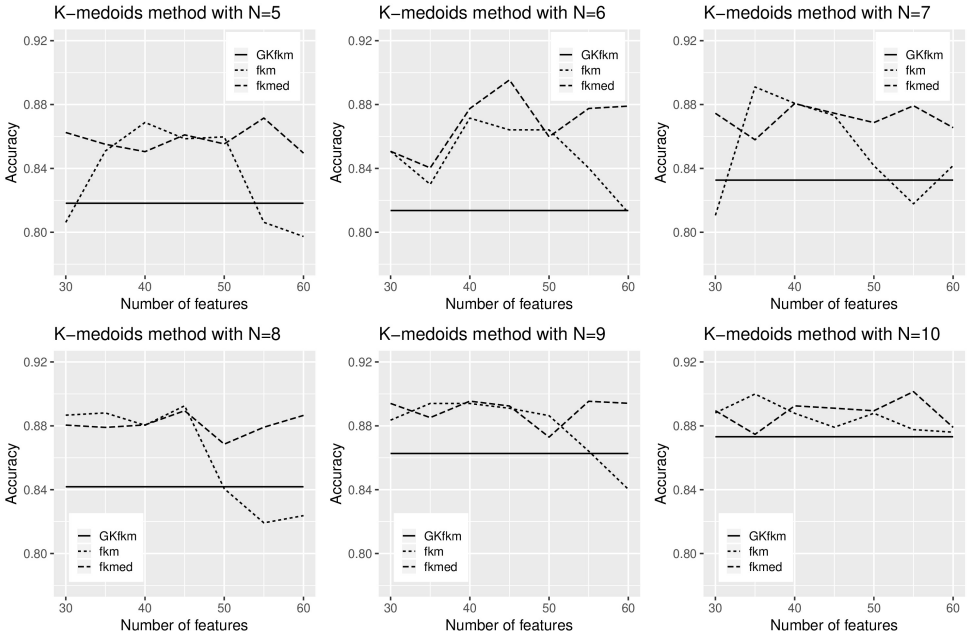

Figure 4.7: Accuracy values for the k-medoids feature selection.

hierarchical method gives poor results and this result does not change with the number of features. The k-means method, however, doesn't follow a common pattern, being worse than the k-medoids approach most of the time, but having higher accuracy for a few cases. As seen in Figure 4.9, the feature selection methods are more robust and none of them is particularly better than the other when the fkmed method is applied. Additionally, the change in the feature selection method does not have an effect on the accuracy when GKfkm is used. The only exception is seen when the number of states is 5 in such a way that the accuracy values for this method do not change within our range for the number of features, but change when we do not implement any feature selection. Lastly, when the hierarchical feature selection method is used with the fkm state selection, the number of features affects the accuracy slightly and the accuracy value is very poor.

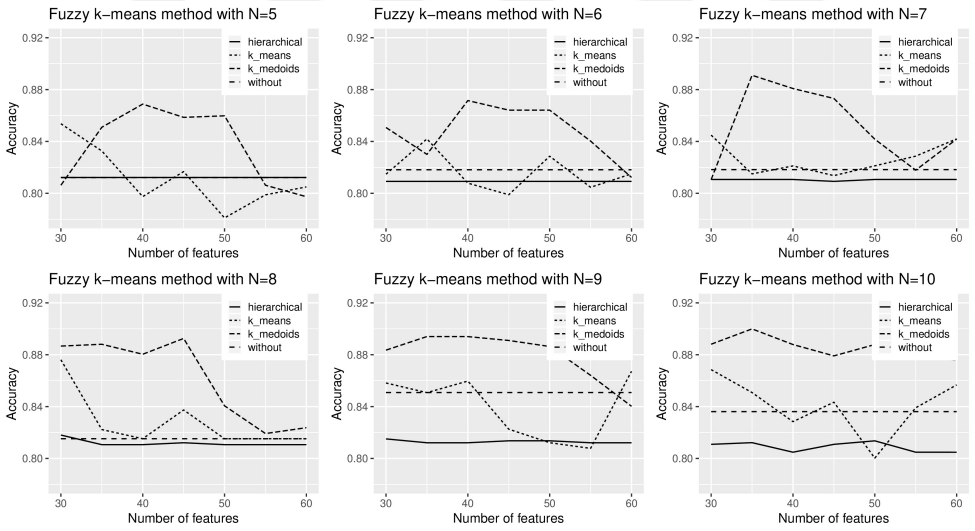

Figure 4.8: Accuracy values for the fuzzy k-means state selection.

4. Effect of the state selection methods on the accuracy: When we compare the state selection methods, we see that GKfkm is very robust to the number of features and the feature selection methods, but the accuracy changes when the number of states changes. fkm is also very robust to the changes in the number of features when the hierarchical feature selection method is performed. As seen in Figure 4.5, when the hierarchical feature selection method is applied, fkmed always gives the best results, GKfkm produces worse results and fkm

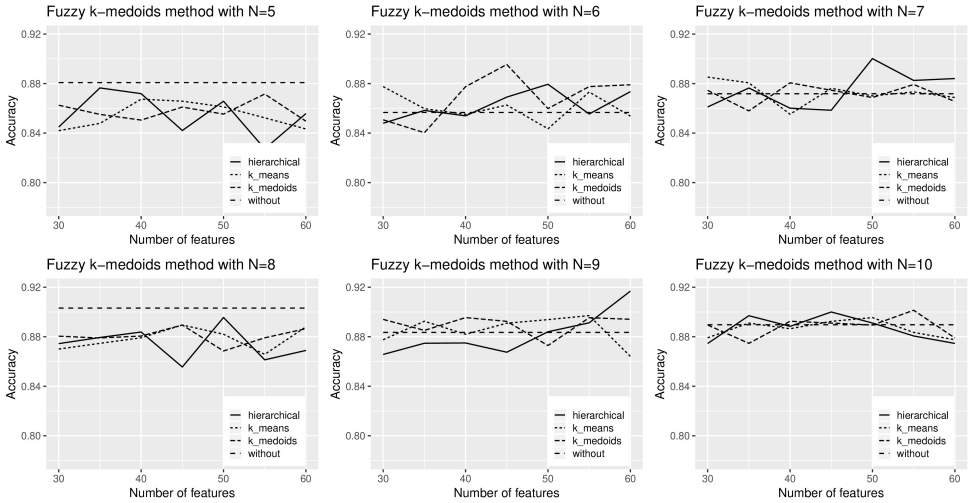

Figure 4.9: Accuracy values for the fuzzy k-medoids state selection.

shows the worst outcomes. This is an expected finding since the fkmed method is more robust to outliers in the data and the GKfkm method captures non-spherical patterns, unlike fkm. As seen in Figure 4.6, a similar pattern appears when the k-means feature selection is implemented, except in some cases fkm surpasses fkmed and GKfkm. Moreover, fkm works more efficiently when it is used with the k-medoids feature selection method. As seen in Figure 4.7, in some cases fkm gives better results than fkmed and in many cases, it shows better results than GKfkm. Overall, there are 132 different cases when all other hyperparameters are fixed except state selection methods. The fkmed method is the best among state selection methods 118 times out of 132 cases, followed by fkm which is the best 14 times and GKfkm is never the best among other methods.

5. Effect of the feature selection on the accuracy: When fkm is implemented for the state selection, only the k-medoids and sometimes the k-means feature selection give higher accuracy compared to no feature selection. When the GKfkm state selection method is used, we observe that there is no difference between the feature selection and the feature selection findings. On the other hand, when fkmed is applied as the state selection, using all the dataset without any feature selection show either the best or comparable results to the models with the feature selection approach.

As a result, at the end of the training process, we select the hierarchical feature selection method with 60 features and the fuzzy k-medoids state selection with 9 numbers of states as the optimal choices for our analyses. The associated states can be seen in Table 4.1. In the paper of Zhang et al. (2006), a method, called the multiple property grouping, is suggested. In that paper, they built a graph using the number of common features that amino acids share and declare the cliques of this graph as states. We apply this method to our dataset and compare the results on both the 10-fold cross-validation values on the training data and on the test data. The measures taken for the comparison are smaller on the test data than the training data since the model does not see the test data throughout the training process. As presented in Table 4.3, from the outcomes, it is observed that the proposed model gives better results than the multiple property grouping on both training and the test data on almost all measures except a slightly smaller value of the precision on the test data.

Model	**Precision**	**Recall**	**F-score**	**MCC**	**Accuracy**
Proposed model training values	0.908	0.945	0.924	0.837	0.917
Multi property model training values	0.886	0.917	0.900	0.777	0.888
Proposed model test values	0.864	0.950	0.905	0.789	0.893
Multi property model test values	0.875	0.875	0.875	0.732	0.867

Table 4.3: Comparison of the proposed model with the multiple property grouping the state selection method.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In conclusion, there are various fields where networks based on circular data emerge. The first part of this thesis deals with the problem of constructing a network model with as little information as possible. Up to date, circular network models have either been parametric and therefore, work under strict distributional assumptions or they have been built only for certain network structures such as trees or HMMs. Unlike these methods, the proposed model is working without distributional assumptions in a more general setting. Hence, we consider that our model can be promising in this field and hopefully will lead to more complicated models for specific applications in each field where circular network structures emerge. Furthermore, we constructed a novel simulation setting for circular networks which can be used to compare the efficiencies of different network models in the circular field, which was not available to date.

Although the model can be used on various applications as it is, there exist issues that should be addressed in future work to improve. First, the model needs a large sample size. Second, as the network size gets bigger, the search space for the optimal regression coefficients gets intractable. Therefore, for larger networks, we need to consider refinements of the optimization algorithm used or applications of more powerful new algorithms. However, intensive simulations showed that our model can be safely deployed when the sample size is at least 5 times bigger than the number of nodes and when the network size is not above 20, given the underlying distribution follows a wrapped normal distribution. For the circadian gene networks, utilizing only the circular nature of the data causes a loss of information. To avoid this, the height of the data points can be added to the models together with the waveforms. Models, where circular and real data are integrated, can be created for this purpose. Additionally,

finding circadian gene data is difficult in public databases, we will keep searching for this data type.

Furthermore, in the second part of this thesis, we tried to utilize known network structures for the betterment of the models. We tried to solve a classification problem with the help of the known network structure behind. Here, different from the first part, we already know the structure. When the structure or the distribution behind is known, this information can be used to provide additional leverage to the methods applied. We used the hidden Markov model (HMM) in order to detect the lock-and-key relationship in the Chip-seq data. In the application, we have initially explained the mathematical details of HMM in different stages of the estimation of the model parameters via the expectation-maximization method. Furthermore, we have investigated the effect of the clustering approaches in different aspects of the selection of the observations which are the sequence of amino acids, and the states which are biophysical features of amino acids. In these analyses, we have conducted various methods from k-means and hierarchical techniques to fuzzy clustering. The application of fuzzy clustering instead of hard clustering shows an alternative way to utilize the physio-chemical features of amino acids, which is more reliable when applicable. From the findings, we have observed that HMM is promising to describe the selected benchmark Chiq-seq dataset and the proposed clustering approaches have improved its accuracy. The proposed model gives better results for classifying octamers as cleaved or non-cleaved because of the clever starting point that we create. This model can be used to narrow down the search grid in experiments investigating cleavage sites of amino acid chains regarding any illness. Additionally, we presented a toy example in detail. It clarifies how the HMM model works for inexperienced researchers or practitioners. Following this example, they can easily adapt the HMM model to their use cases. In the future, same strategy can be adapted for a range of bioinformatics problems including amino acids. When we work with the amino acids, instead of hard clustering methods fuzzy clustering can be used to extract more information. Furthermore, in the future, we will keep looking for opportunities to leverage the integration of data coming from different sources for further analyses.

REFERENCES

[1] Alonso-Pena, M., Ameijeiras-Alonso, J., and Crujeiras, R. M. (2021). Nonparametric tests for circular regression. *Journal of Statistical Computation and Simulation* **91(3),** 477–500.

[2] Athar, A., Füllgrabe, A., George, N., Iqbal, H., Huerta, L., Ali, A., Snow, C., Fonseca, N. A., Petryszak, R., Papatheodorou, I., Sarkans, U., and Brazma, A. (2019). ArrayExpress update – from bulk to single-cell expression data. *Nucleic Acids Research* **47(D1),** D711–D715.

[3] Baggs, J. E., Price, T. S., DiTacchio, L., Satchidananda, P., FitzGerald, G. A., and Hogenesch, J. B. (2009). Network Features of the Mammalian Circadian Clock. *PLoS Biol* **7(3),** 1–13.

[4] Bezdek J. (1981). Pattern Recognition With Fuzzy Objective Function Algorithms. Plenum Press, New York.

[5] Boomsma, W., Mardia, K. V., Taylor, C. C., Ferkinghoff-Borg, J., Krogh, A., and Hamelryck, T. (2008). A generative, probabilistic model of local protein structure. *Proceedings of the National Academy of Sciences of the United States of America* **105(26),** 8932–8937.

[6] Braun, R., Kath, W. L., Iwanaszko, M., Kula-Eversole, E., Abbott, S. M., Reid, K. J., Zee, P. C., and Allada, R. (2018). Universal method for robust detection of circadian state from gene expression. *Proceedings of the National Academy of Sciences of the United States of America* **115(39),** E9247–E9256.

[7] Cai, Y. D., and Chou, K. C. (1998). Artificial neural network model for predicting HIV protease cleavage sites in protein. Adv. Eng. Softw. **29(2),** 119–128.

[8] Cai, Y. , Liu, X. , Xu, X. and Chou, K. (2002). Support vector machines for predicting HIV protease cleavage sites in protein. *J. Comput. Chem.* **23,** 267–274.

[9] Chen, B., Ding, Y., and Wild, D. J. (2012). Assessing drug target association using semantic linked data. *PLoS Comput. Biol.* **8(7).**

[10] Chormungea, S., and Jenab, S. (2018). Correlation based feature selection with clustering for high dimensional data. *J. Electr. Syst. Inf. Technol.*

[11] Chudova, D., Ihler, A., Lin, K. K., Andersen, B., and Smyth, P. (2009). Bayesian detection of non-sinusoidal periodic patterns in circadian expression data. *Bioinformatics* **25(23),** 3114–3120.

[12] Czeisler, C. A., Duffy, J. F., Shanahan, T. L., Brown, E. N., Mitchell, J. F., Rimmer, D. W., Ronda, J., Silva, E. J., Allan, J. S., Emens, J. S., Dijk, D. R., and Kronauer, R. E. (1999). Stability, precision, and near-24-hour period of the human circadian pacemaker. *Science* **284,** 2177–2181.

[13] Dar, E. D., Purutçuoğlu, V., Purutçuoğlu, E. (2020). Detection of HIV-1 Protease Cleavage Sites via Hidden Markov Model and Physicochemical Properties of Amino Acids. *In: Machado, J., Özdemir, N., Baleanu, D. (eds) Numerical Solutions of Realistic Nonlinear Phenomena.* **Nonlinear Systems and Complexity, vol 31. Springer, Cham.** 171–193.

[14] Di Marzio, M., Panzera, A., and Taylor, C. C. (2013). Non-parametric Regression for Circular Responses. *Scandinavian Journal of Statistics* **40(2),** 238–255.

[15] Downs, T. D., and Mardia, K. V. (2002). Circular regression. *Biometrica* **89,** 683–697.

[16] Durbin, R., Eddy, S., Krogh, A., Mitchison, G. (1998). Biological Sequence Analysis. Cambridge, UK.

[17] Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research* **30(1),** 207–210.

[18] Eisner, J. (2002). An interactive spreadsheet for teaching the forward-backward algorithm. *Proc. of the ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL* 10–18.

[19] Faber, J., and Fonseca, L. (2014). How sample size influences research outcomes. *Dental press journal of orthodontics* **19(4),** 27–29.

[20] Fan, J., Feng, Yang., and Wu, Y. (2009). Network exploration via the adaptive LASSO and SCAD penalties. *Ann. Appl. Stat.* **3(2),** 521–541.

[21] Fisher, N. I., and Lee, A. J. (1982). Nonparametric Measures of Angular-Angular Association. *Biometrika* **69,** 315–321.

[22] Fisher, N. I. (1993). Statistical analysis of circular data. *Cambridge University Press*.

[23] Friedman J., Hastie T., and Tibshirani R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **Jul;9(3),** 432–441.

[24] Gallo, R. C., Salahuddin, S. Z., Popovic, M., Shearer, G. M., Kaplan, M., Haynes, B. F., Palker, T. J., Redfield, R., Oleske, J., Safai, B., White, Cl., Foster, P., and Markham, P. D. (1984). Frequent detect on and isolation of cytopathic retroviruses (HTLV-III) from patients with AIDS and at risk for AIDS. *Science* **224(4648),** 500–503.

[25] Glynn, E. F., Chen, J., and Mushegian, A. R. (2006). Detecting periodic patterns in unevenly spaced gene expression time series using Lomb-Scargle periodograms. *Bioinformatics* **22(3),** 310–316.

[26] Gottard, A., and Panzera, A. (2021). Graphical models for circular variables.

[27] Gould, A. L. (1969). A Regression Technique for Angular Variates. *Biometrics* **25(4),** 683–700.

[28] Gustafson, D. E., and Kessel, W. C. (1978). Fuzzy clustering with a fuzzy covariance matrix. *Proc. IEEE CDC* 761–766.

[29] Hartigan, J. A., and Wong, M. A. (1979). Algorithm AS 136: A K-means clustering algorithm. *J. Royal Stat. Soc. Series C (Applied Statistics)* **28,** 100–108.

[30] Hickey, D. S., Kirkland, J. L., Lucas, S. B., and Lye, M. (1984). Analysis of circadian rhythms by fitting a least squares sine curve. *Comput. Biol. Med.* **14(2),** 217–223.

[31] Jaeger, S., and Chen, S.-S. (2010). Information fusion for biological prediction. *J. Data Sci.* **8,** 269–288.

[32] Jammalamadaka, S. R., and Sarma, Y. R. (1988). A Correlation Coefficient for Angular Variables. *Stat. Theory Data Anal. II (Ed. Matusita)* **115(39),** 349–364.

[33] Jayavardhana, Rama, G.L., and Palaniswami, M. (2005). Cleavage knowledge extraction in HIV-1 protease using hidden Markov model. *Proc. 2nd International Conference on Intelligent Sensing and Information Processing*, 469–473.

[34] Jianying, H., Brown, M.K., and Turin, W. (1996). HMM Based Online Handwriting Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**, 1039–1045.

[35] Johnson, R. A., and Wehrly, T. E. (1978). Some angular-linear distributions and related regression models. *Journal of the American Statistical Association* **73(363),** 602–606.

[36] Juang, B., and Rabiner, L. (1991). Hidden Markov Models for Speech Recognition. *Technometrics* **33(3)**, 251–272.

[37] Jupp, P. E., and Mardia, K. V. (1981). A general correlation coefficient for directional data and related regression problems. *Biometrika* **68,** 738.

[38] Kato, S., Shimizu, K., and Shieh, G. (2008). A circular-circular regression model. *Statistica Sinica* **18,** 633–645.

[39] Kato, S., and Jones, M. C. (2010). A family of distributions on the circle with links to, and applications arising from, Möbius transformation. *Journal of the American Statistical Association* **105(489),** 249–262.

[40] Kim, G., Kim, Y., Lim, H., and Kim, H. (2010). An MLP-based feature subset selection for HIV-1 protease cleavage site analysis. *Artif. Intell. Med.* **48,** 83–89.

[41] Kim, S., SenGupta, A., and Arnold, B. (2016). A multivariate circular distribution with applications to the protein structure prediction problem. *Journal of Multivariate Analysis* **143.**

[42] Kim, S., and SenGupta, A. (2016). Multivariate-multiple circular regression. *Journal of Statistical Computation and Simulation* **87,** 1–15.

[43] Kohl, N. E., Emini, E. A., Schlief, W. A., Davis, L. J., Heimbach, J., Dixon, R. A.F., Scolnik, E. M., and Sigal, I. S. (1988). Active human immunodeficiency virus protease is required for viral infectivity. *Proc. Nutl. Sci. USA*. **85(15)**, 4686–4690.

[44] Kouemou, G. L. (2011). History and Theoretical Basics of Hidden Markov Models. *In (Ed.), Hidden Markov Models Theory and Applications, IntechOpen*.

[45] Krishnapuram, R., Joshi, A., Nasraoui, O., and Yi, L. (2001). Low-complexity fuzzy relational clustering algorithms for Web mining. *IEEE Trans. Fuzzy Syst.* **9(4),** 595–607.

[46] Kubiak, T., and Jonas, C. (2007). Applying circular statistics to the analysis of monitoring data: Patterns of social interactions and mood. *European Journal of Psychological Assessment* **23,** 227–237.

[47] Kurz, G., Gilitschenski, I., and Hanebeck, U. (2014). Efficient Evaluation of the Probability Density Function of a Wrapped Normal Distribution. *Proceedings of IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2014)*.

[48] Laycock, P. J. (1975). Optimal design: Regression models for directions. *Biometrika* **62(2),** 305–311.

[49] Leguey, I., Larrañaga, P., Bielza, C., and Kato, S. (2019). A circular-linear dependence measure under Johnson – Wehrly distributions and its application in Bayesian networks. *Information Sciences* **486,** 240–253.

[50] Li, H., and Gui, J. (2006). Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics* **7(2),** 302–317.

[51] Li, S., Shui, K., Zhang, Y., Lv, Y., Deng, W., Ullah, S., Zhang, L., and Xue, Y. (2017). CGDB: a database of circadian genes in eukaryotes. *Nucleic Acids Research* **45(D1),** D397–D403.

[52] Liang, K. C., Wang, X., and Li, T. H. (2009). Robust discovery of periodically expressed genes using the laplace periodogram. *BMC Bioinformatics* **10(15).**

[53] Luan, Y., and Li, H. (2004). Model-based methods for identifying periodically expressed genes based on time course microarray gene expression data. *Bioinformatics* **20(3),** 332–339.

[54] Lund, U. (1999). Least circular distance regression for directional data. *Journal of Applied Statistics* **26(6),** 723–733.

[55] Lund, U. J. (2002). Tree-based regression for a circular response. *Communications in Statistics - Theory and Methods* **31,** 1549–1560.

[56] Mardia, K., and Sutton, T. (1978). A Model for Cylindrical Variables with Applications. *Journal of the Royal Statistical Society: Series B (Methodological)* **40,** 229–233.

[57] Mardia, K. V., Hughes, G., Taylor, C. C., and Singh, H. (2008). A multivariate von Mises distribution with applications to bioinformatics. *Canadian Journal of Statistics* **36,** 99–109.

[58] McDonald, M. J., and Rosbash, M. (2001). Microarray analysis and organization of circadian gene expression in Drosophila. *Cell* **107(5),** 567–578.

[59] Meinshausen, N., and Bühlmann, P. (2006). High-Dimensional Graphs and Variable Selection with the Lasso. *The Annals of Statistics* **34(3),** 1436–1462.

[60] Miller, M., Schneider, J., Sathyanarayana, B. K., Toth, M. V., Marshall, G. R., Clawson, L., Selk, L., Kent, S. B. H. and Wlodawer, A. (1989). Structure of complex of synthetic HIV-l protease with a substrate-based inhibitor at 2.3 A resolution. *Science.* **246,** 1149–1152.

[61] Mitra, P., Murthy, C. A., and Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **24(3),** 301–312.

[62] Munch, K., and Krogh, A. (2006). Automatic generation of gene finders for eukaryotic species. *BMC Bioinform.* **7,** 263.

[63] Murtagh, F. (1985). Multidimensional Clustering Algorithms. *Physica-Verlag*.

[64] Nakai, K., Kidera, A., and Kanehisa, M. (1988). Cluster analysis of amino acid indices for prediction of protein structure and function. *Protein Eng.* **2,** 93–100.

[65] Nanni, L. (2006). Comparison among feature extraction methods for HIV-1 protease cleavage site prediction. *Pattern Recognit.* **39(4),** 711–713.

[66] Niu, B., Yuan, X. C., Roeper, P., Su, Q., Peng, C. R., Yin, J. Y., Ding, J., Li, H., and Lu, W. C. (2013). HIV-1 protease cleavage site prediction based on two-stage feature selection method. *Protein Pept. Lett.*, **20,** 290–298.

[67] Otieno, S., and Anderson-Cook, C. M. (2006). Measures of preferred direction for environmental and ecological circular data. *Environmental and Ecological Statistics* **13,** 311–324.

[68] Pachter, L., Alexandersson, M., and Cawley, S. (2002). Applications of generalized pair hidden Markov models to alignment and gene finding problems. *J. Comput. Biol.* **9,** 389–399.

[69] Park, H., and Jun, C. (2009). A simple and fast algorithm for k-medoids clustering. *Expert Syst. Appl.* **36,** 3336–3341.

[70] Razavian, N. S., Kamisetty, H., and Langmead, C. J. (2011). The von mises graphical model:structure learning. *Technical Report CMU-CS-11-108, Carnegie Mellon University.*

[71] Rivest, L.-P. (1997). A Decentred Predictor for Ciruclar-Circular Regression. *Biometrika* **84(3),** 717–726.

[72] Roberts, N. M., Tikoff, B., Davis, J. R., and Stetson-Lee, T. (2019). The utility of statistical analysis in structural geology. *Journal of Structural Geology* **125,** 64–73.

[73] Rögnvaldsson, T., You, L., and Garwicz, D. (2015). State of the art prediction of HIV-1 protease cleavage sites. *Bioinformatics* **31,** 1204–1210.

[74] Rueda, C., Fernández, M. A., Barragán, S., Mardia, K. V., and Peddada, S. D. (2016). Circular piecewise regression with applications to cell-cycle data. *Biometrics* **Dec;72(4),** 1266–1274.

[75] Sarma, Y., and Jammalamadaka, S. (1993). Circular Regression. *Statistical Science and Data Analysis. Proceedings of the Third Pacific Area Statistical Conference* 109–128.

[76] Schäfer, J., and Strimmer, K. (2005). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics.* **21(6),** 754–764.

[77] Schroff, R. W., Gottlieb, M. S., Prince, H. E., Chai, L. L., and Fahey, J. L. (1983). Immunological studies of homosexual men with immunodeficiency and Kaposi's sarcoma. *Clin. Immunol. Immunopathol.* **27(3),** 300–314.

[78] SenGupta, A., Kim, S., and Arnold, B. C. (2013). Inverse circular-circular regression. *J. Multivar. Anal.* **119,** 200–208.

[79] Sesane, M., and Geyer, S. (2017). The perceptions of community members regarding the role of social workers in enhancing social capital in metropolitan areas to manage HIV and AIDS. *Social Work*, **53(1),** 1–26.

[80] Shieh, G., and Johnson, R. (2005). Inferences based on a bivariate distribution with von Mises marginals. *Annals of the Institute of Statistical Mathematics* **57,** 789–802.

[81] Singh, H., Hnizdo, V., and Demchuk, E. (2002). Probabilistic model for two dependent circular variables. *Biometrika* **89,** 719–723.

[82] Starner, T., and Pentland, A. (1995). Real-time American Sign Language recognition from video using hidden Markov models. *Proc. of International Symposium on Computer Vision - ISCV* 265–270.

[83] Strug, D. L., Grube, B. A., and Beckerman, N. L. (2008). Challenges and changing roles in HIV/AIDS. *Social Work in Health Care.* **35(4),** 1–19.

[84] Stultz, C. M. (1993). Structural analysis based on state-space modeling. *Protein Sci.* **2,** 305–314.

[85] Sukumaran, S., Almon, R. R., DuBois, D. C., and Jusko, W. J. (2010). Circadian rhythms in gene expression: Relationship to physiology, disease, drug disposition and drug action. *Advanced drug delivery reviews* **62(9-10),** 904–917.

[86] Szklarczyk, D., Gable, A. L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., Simonovic, M., Doncheva, N. T., Morris, J. H., Bork, P., Jensen, L. J., and Mering, C. V. (2019). STRING v11: protein-protein association networks with in-

creased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Research* **47(D1),** D607–D613.

[87] Thompson, T. B., Chou, K. C., and Zheng, C. (1995). Neural Network Prediction of the HIV-1 Protease Cleavage Sites. *J. Theor. Biol.* **177(4),** 369–379.

[88] Turhal, U., Gök, M., and Durgut, A. (2015). Comparison among Feature Encoding Techniques for HIV-1 Protease Cleavage Specificity. *Int. J. Intell. Syst. Appl. Eng.* **3(2),** 62–66.

[89] UNAIDS. http://www.unaids.org/en/resources/fact-sheet.

[90] Wallach, T., Schellenberg, K., Maier, B., Kalathur, R. K. R., Porras, P., Wanker, E. E., Futschik, M. E., and Kramer, A. (2013). Dynamic Circadian Protein-Protein Interaction Networks Predict Temporal Organization of Cellular Functions. *PLoS Genetics* **Mar;9**.

[91] Wang, M., Zhou, Z., Khan, M. J., Gao, J., and Loor, J. J. (2015). Clock circadian regulator (CLOCK) gene network expression patterns in bovine adipose, liver, and mammary gland at 3 time points during the transition from pregnancy into lactation. *J. Dairy Sci.* **98(7),** 4601–4612.

[92] White, J. V. (1994). Protein classification by stochastic modeling and optimal filtering of amino-acid sequences. *Math. Biosci.* **119,** 35–75 .

[93] Zhang, C., Bickis, M.G., Wu, F.X., and Kusalik, A. J. (2006). Optimally-connected hidden markov models for predicting MHC-binding peptides. *J. Bioinform. Comput. Biol.* **4(5),** 959–980.

[94] Zhao, T., Liu, H., Roeder, K., Lafferty, J. D., and Wasserman, L. A. (2012). The huge package for high-dimensional undirected graph estimation in R. *Journal of machine learning research : JMLR* **13,** 1059–1062.

CURRICULUM VITAE

ELİF DOĞAN DAR

EDUCATION

- **Ph.D. in Statistics** 2016-2023
  Middle East Technical University (METU)
  Council of Higher Education (YÖK) 100/2000 PhD Scholarship
- **M.S. in Mathematics** 2012-2015
  Bilkent University
  Bilkent University Mathematics Department Scholarship
- **B.S. in Mathematics** 2007-2012
  Boğaziçi University
  TUBITAK BIDEB Undergraduate Scholarship

PUBLICATIONS AND CONFERENCES

- Doğan Dar, E., Purutçuoğlu, V. (2023) "A Non- Parametric Circular-Circular Graphical Network Construction via Simulated Data" (Manuscript under review.)
- Doğan Dar, E., Purutçuoğlu, V. (2020) "Circular-Circular Graphical Network via Simulated and Real Data" CA15109: Final COSTNET Meeting 2020. Slovenia.
- Doğan Dar, E., Purutçuoğlu, V. (2019) "Circadian Gene Interactions: A Circular Approach" 2nd Eurasia Biochemical Approaches and Technologies (EBAT 2019).
- Doğan Dar, E., Purutçuoğlu, V. and Purutçuoğlu, E. (2019) "Detection of HIV-1 protease cleavages sites via hidden Markov model and physicochemical properties of aminoacids" Chapter in: Numerical Solutions of Real-Life Nonlinear Phenomena. Springer.
- Doğan Dar, E., Purutçuoğlu, V. (2018) "Detection of HIV-1 Cleavage Proteins via Hidden Markov Model" International Conference on Applied Mathematics in Engineering (ICAME18).

SELECTED PROJECTS

Detection of HIV-1 Protease Cleavage Sites via Hidden Markov Models (HMM) in R

- HMMs for the classification of aminoacid chains. Hidden states were clusters of aminoacids, which I built using fuzzy clustering. I used packages ppclust, cluster, fclust, kmed, and aphid.

Circular-Circular Graphical Networks via Simulated and Real Data in R

- I created a novel model for the network structure estimation for circular data, and applied on circadian gene interactions. I used packages Huge and matrix-calc.

Structural Bioinformatics in Python

- Solutions to some structural bioinformatics problems, which deal with the three-dimensional structure of proteins. I used numpy, pandas and matplotlib.

Classification for NHANES Data in R

- Random forest model for classification with exploratory data analysis and missing data treatment. I used packages mice, caret, ggplot2 and randomForest.

Time Series Analysis and Forecasting in R

- Time series analysis and forecasting for house pricing data. I applied ARIMA, dynamic harmonic regression, vector autoregression and Holt- Winters models. I utilized packages lubridate, forecast, zoo, ggfortify, tidyr, dplyr and ggplot2.

Panel Data Analysis in R

- Panel data analysis for a clinical trial of patients with epilepsy. I applied marginal, transition and random effect models to assess the difference in treatments as well as effects of various covariates. I used packages epiDisplay, lattice, QICpack, geepack and lme4.

SKILLS

- **R**: ggplot2, tidyr, dplyr, lme4, caret, forecast, lubridate, Amelia, mice, Bioconductor, affy
- **Python**: pandas, numpy, matplotlib
- **SQL**: SQLite

EXPERIENCE

- **Ph.D. Intern** 2023
  Saez-Rodriguez Group, Heidelberg, Germany
- **Teaching Assistant in Mathematics** 2012-2015
  Bilkent University: Calculus, Linear Algebra and Statistics courses