Gaussian Process in Computational Biology: Covariance Functions for Transcriptomics

Muhammad Arifur Rahman

Department of Computer Science

University of Sheffield

A thesis submitted in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy

September 2017

To my loving family

Declaration

This thesis describes the work carried out between October 2011 and December 2016 in the ML@SITraN Group at The University of Sheffield, UK under the supervision of **Prof. Neil D. Lawrence**.

This thesis has been composed by myself and has not, nor any similar dissertation been submitted in any previous application for a degree.

Muhammad Arifur Rahman
September 2017

The University Of Sheffield.

Access To Thesis.

- A bound copy of every thesis which is accepted as worthy for a higher degree, must be deposited in the University of Sheffield Library, where it will be made available for borrowing or consultation in accordance with University Regulations.
- All students registering from 2008–09 onwards are also required to submit an electronic copy of their final, approved thesis. Students who registered prior to 2008–09 may also submit electronically, but this is not required.

Author: Muhammad Arifur Rahman Dept: Computer Science

Thesis Title: Gaussian Process in Computational Biology: Covariance Functions for Transcriptomics Registration No: 110121714

For completion by all students:

Submit in print form only (for deposit in the University Library): ☑

Submit in print form and also upload to the *White Rose eTheses Online* server: In full ☑

Edited eThesis ☑

Please indicate if there are any embargo restrictions on this thesis. Please note that if no boxes are ticked, you will have consented to your thesis being made available without any restrictions.

Embargo details: (complete only if requesting an embargo to either your print and/or eThesis)	Embargo required?		Length of embargo (in years)
Print Thesis	Yes ☐	No ☑	____
eThesis	Yes ☐	No ☑	____

Supervisor: I, the supervisor, agree to the named thesis being made available under the conditions specified above.

Name: Prof. Neil D. Lawrence Dept: Computer Science

Signed:  Date: 12.12.2016

Student: I, the author, agree to the named thesis being made available under the conditions specified above.

I give permission to the University of Sheffield to reproduce the print thesis in whole or in part in order to supply single copies for the purpose of research or private study for a non-commercial purpose.

I confirm that this thesis is my own work, and where materials owned by a third party have been used copyright clearance has been obtained. I am aware of the University's *Guidance on the Use of Unfair Means* (www.sheffield.ac.uk/lets/design/unfair)

I confirm that all copies of the thesis submitted to the University (including electronic copies on CD/DVD) are identical in content.

Name: Muhammad Arifur Rahman Dept: Computer Science

Signed:  Date: 12.12.2016

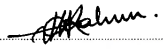
For completion by students also submitting an electronic thesis (eThesis):

I, the author, agree that the University of Sheffield's eThesis repository (currently WREO) will make my eThesis available over the internet via an entirely non-exclusive agreement and that, without changing content, WREO may convert my thesis to any medium or format for the purpose of future preservation and accessibility.

I, the author, agree that the metadata relating to the eThesis will normally appear on both the University's eThesis server and the British Library's EThOS service, even if the thesis is subject to an embargo. I agree that a copy of the eThesis may be supplied to the British Library.

I confirm that the upload is identical to the final, examined and awarded version of the thesis as submitted in print to the University for deposit in the Library (unless edited as indicated above).

Name: Muhammad Arifur Rahman Dept: Computer Science

Signed:  Date: 12.12.2016

THIS SHEET MUST BE BOUND IN THE FRONT OF THE PRINTED THESIS BEFORE IT IS SUBMITTED

Acknowledgements

I would like to express my deepest gratitude to my mentor, Prof. Neil D. Lawrence for accepting me as a post-graduate research student. Neil initiated me into the discipline of Machine Learning in Computational Biology and continuously taught me the most important scientific skills: showed me the connection between data, model and reality. Very few have these skills and I was privileged to learn from a true master. Neil's contribution in this thesis is fundamental, from the basic idea of Gaussian process to little comments that made my work presentable with confidence and clarity.

I wish to thank Prof. Guy J. Brown and Prof. Richard H. Clayton, the members of my supervisory committee at the University of Sheffield for their advice, valuable suggestions and insightful comments what guided me through all these years. I would like to thank Paul R. Heath for his collaboration and providing me with biological insights, which guided me in a better direction. I also wish to thank Professor Nick Monk and Dr. Simon Rogers - my thesis examiners, for their helpful comments and precious suggestions.

Good science is always the joint effort of many people and the research in this thesis is no exception. This thesis could never have happened without my fellow colleagues, with their constructive discussions and spontaneous assistance whenever or wherever I was in need. Alphabetically: Teo de Campos, Mike Croucher, Luisa Cutillo, Zhenwen Dai, Andreas Damianou, Nicolas Durrande, Nicoló Fusi, James Hensman, Javier Gonzalez Hernandez, Alfredo Kalaitzis, Ciira Maina, Jens D Nielson, Mu Niu, Ricardo Andrade Pacheco, Alan Saul, Michael Smith, Alessandra Tosi, Bei Wang, Fariba Yousefi, Sura Zaki, Max Zwiessele. I felt like this was a family and Neil was our academic Dad. There are simply no words to express my gratitude to them.

I would also like to thank the many people who gave me the support and technical backing so that I could focus on my research. I gratefully acknowledge the Ministry of Science and Technology, Bangladesh, for the fellowship awarded to me and the Department of Physics, Jahangirnagar University, Bangladesh for issuing me the study leave. I thank the Department of Computer Science, University of Sheffield for consistently providing the best and most reliable computer support possible. I thank all

the administrative staff at the Sheffield Institute for Translational Neuroscience for all their help and support, most importantly providing me the best working environment ever possible where I could enjoy every bit of weather change from rain to sunshine, hail to snow.

I am grateful to my parents, for being so wonderful and supportive. They nurtured my curiosity, creativity and passion for understanding from the earliest age. Abba, thank you for teaching me the value of life and giving me the strength to chase my dreams. I aspire to be like you. You were and will always be my role model though you no longer with us. Amma, words cannot express how grateful I am to you for all of the sacrifices that you've made for me. Your blessings were and will be the most valuable asset of my life. You are my very inspiration to excel.

I thank my brothers, sisters, brother-in-laws and sister-in-laws specially Ezabul Hossain and Shamim Reza for caring me so much and for their constant effort to step forward in my life since my childhood to writing up this thesis. Without their love, support and inspiration it would never possible to reach this stage of my life.

I apologize to my son Adib, who paid the heaviest price for this thesis, during the endless hours I worked away from him. I hope when you will grow up, you will understand and forgive me. I thank my father-in-law and mother-in-law for their mental support and countless hours to take care of Adib when he was in Bangladesh without me, and met up his every requirement with the best of love and care.

My special regards to my teachers, whose teaching at different levels of education has made it possible for me to reach a stage where I could write this thesis. My students are always the best source of my inspiration to continue my walk toward learning. I would like to thank them all. I like to thank M. Shamim Kaiser for staying beside me and playing roles of teacher, brother and above all a friend depending on my needs. I would also like to thank all my old friends who do not need to be named to know their importance to me!

I owe my deepest gratitude towards my better half for her eternal support and understanding of my goals and aspirations. Her infallible love and support has always been my strength. Her patience and sacrifice will remain my inspiration throughout my life. Without her help, I would not have been able to complete much of what I have done and become who I am. I am endlessly indebted and it would be ungrateful on my part if I thank Zehan in these few words.

Muhammad Arifur Rahman
The University of Sheffield, UK

Abstract

In the field of machine learning, Gaussian process models are widely used families of stochastic process for modelling data observed over time, space or both. Gaussian processes models are nonparametric, meaning that the models are developed on an infinite-dimensional parameter space. The parameter space is then typically learnt as the set of all possible solutions for a given learning problem. Gaussian process distributions are distribution over functions. The covariance function determines the properties of functions samples drawn from the process. Once the decision to model with a Gaussian process has been made the choice of the covariance function is a central step in modelling.

In molecular biology and genetics, a transcription factor is a protein that binds to specific DNA sequences and controls the flow of genetic information from DNA to mRNA. To develop models of cellular processes, quantitative estimation of the regulatory relationship between transcription factors and genes is a basic requirement. Quantitative estimation is complex due to various reasons. Many of the transcription factors' activities and their own transcription level are post transcriptionally modified; very often the levels of the transcription factors' expressions are low and noisy. So, from the expression levels of their target genes, it is useful to infer the activity of the transcription factors. Here we developed a Gaussian process based nonparametric regression model to infer the exact transcription factor activities from a combination of mRNA expression levels and DNA-protein binding measurements.

Clustering of gene expression time series gives insight into which genes may be coregulated, allowing us to discern the activity of pathways in a given microarray experiment. Of particular interest is how a given group of genes varies with different conditions or genetic backgrounds. In this thesis, we developed a new clustering method that allows each cluster to be parametrized according to the behaviour of the genes across conditions whether they are correlated or anti-correlated. By specifying the correlation between such genes, we gain more information within the cluster about how the genes interrelate. Our study shows the effectiveness of sharing information between replicates and different model conditions while modelling gene expression time series.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Machine learning is a joint field of artificial intelligence and modern statistics, predominantly focused on the design and development of models, algorithms and techniques that allow computers to extract information automatically, by some learning process, from data. The structure learned from data can be described by a statistical model. Gaussian process models are well-known families of stochastic processes for modelling data observed over time, space or both. Data modelling with Gaussian process is a state-of-the-art technique in the wider community, from robotics (Deisenroth et al. (2014)) to genomics (Topa et al. (2015)), from astronomy (Rajpaul et al. (2015)) to meteorology (Chen et al. (2014)). Gaussian process models are nonparametric, which means the models are developed on an infinite-dimensional parameter space. For a particular learning problem, the parameter space is typically learnt as a set of possible solutions. There are different ways to learn functions. Probabilistic inference is one of the elegant and widely accepted way among them. In the field of machine learning regression is a supervised learning problem, while clustering is an unsupervised learning problem. A regression task is related to making predictions of a continuous output variable at any desired input location, given an input-output training set. A clustering task groups a set of observations into subsets (also known as clusters) so that observations in the same cluster shows similarity in some particular sense. Here we set two generic goals for this thesis

Generic goal 1: We will develop a tool to analyse transcription factor activities. This tool will target the gene expression time series data which is sampled across continuous time.

Generic goal 2: Our second goal is to develop an approach for gene expression clustering that handles structure in the experimental conditions as part of the cluster analysis.

Our primary focus of this thesis is to achieve these goals by building Gaussian process models from transcriptomic data.

1.1 System Biology

The prime goal of Biology is to gain insight of various principles and details of biological systems. More than six decades ago, Watson and Crick discovered the structure of DNA (Watson and Crick (1953)) and changed our approach to study and development of biology and biological systems. They explained the biological phenomena with the help of molecular basis. This concept helps to explain different aspects of biology like heredity, different diseases, various evolutionary aspects as well as development with a firmer theoretical ground. Since then, biology has become a framework of knowledge governed by some basic and fundamental laws of physics.

Due to the enormous advances in molecular biology, at present, we have in-depth knowledge of elementary processes like evolution, heredity, disease, development etc. These mechanisms also include other biological features like replication, transcription and translation. The accomplishment of symbolic DNA sequencing helped to reveal large numbers of genes and their transcriptional products. DNA sequences for many organisms like *Mycoplasma, Plasmodium falciparum, Saccharomyces cerevisiae, Caenorhabditis elegans, Drosophila melanogaster, Homo sapiens* and many more have been fully identified. Due to the advancement of different methods, gene expression profiles are available at the mRNA level. Even measurement of protein levels and their different subsequent actions are also making progress.

Undoubtedly understanding at the molecular level will accelerate understanding of biological systems, but this knowledge is not sufficient to understand biological systems, as systems. Genes and protein are a few components of a whole system. It is necessary to understand what constitutes the system, but even then just this knowledge is not sufficient to understand the complete system. Systems biology is a new field of biology that aims to understand every detail and principles of the biological system (Kitano (2000)).

The extent of the area of system biology is very broad, and various techniques may be required for each individual research target. Very often it demands combined effort from multiple discipline research areas like molecular biology, high-precision measurement technology, mathematics, computer science, control theory and other engineering and scientific fields. Kitano (2002) mentioned the main four key areas to be carried out for further research: (1) genomic and other molecular biology

research, (2) diverse technology for comprehensive and high-precision measurements, (3) computational studies, such as bioinformatics, modelling and simulation, software tools, and (4) analysis of the dynamics of the systems. This depicts the requirement of multidisciplinary research efforts to get the knowledge of biological systems as systems. Indeed the abstract concept of a system is more than a collection of multi-disciplinary research components. Besides the detailed description of the components to acquire the proper insight of system it is also essential to know what happens during the period or processes when any stimuli and/or disruptions take place.

The primary requirement to understand biological systems is the identification of the system structure. Some of the key structures might have different regulatory relationships of genes and interactions with proteins that show the metabolism pathway and signal transduction, the physical structure of chromatin, cells, organisms and other components. It is very critical to monitor biological processes in bulk. High-throughput DNA microarray, real-time polymerase chain reaction (RT-PCR), protein chips and other methods are essential to identify genes and metabolism network. Once a system structure is established up to a certain degree, we need to **u**npick its behaviour. A number of analysis methods can be used to understand this behaviour correctly. For example, consider that we are interested to know the sensitivity of a specified behaviour against some external perturbations and its time to return its normal state since the stimuli took place. This type of analysis provides the system level characteristics as well as uncover valuable insights of medical treatments by revealing cell responses to certain chemical affinities.

To understand the behaviour of the system and to control the state of the biological systems further research is required, with the knowledge previously obtained from the system structure. All these phases lead toward the establishment of technologies that allow us to design a biological system which can provide cures for different diseases. Some futuristic examples could be organ cloning techniques for the treatment of diseases that require organ transplants or building biological materials for engineering, especially robotics, with self-sustaining and self-repairing capabilities.

1.2 Dynamic Mathematical Model: What and Why in System Biology?

Any models are abstractions of reality. Models are designed to focus on specific aspects of the objects for a particular kind of study. Loosely speaking, during these modelling processes other aspects of less interest are abstracted away. Biologists are almost

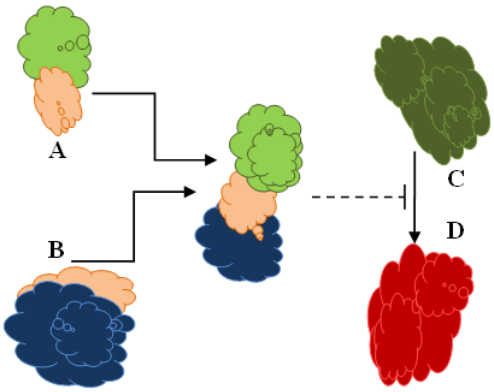


Fig. 1.1 A 'cartoon' model of protein protein interaction. Two different molecular species A and B bind to form a complex molecular. The newly formed complex hinder the rate at which molecules of species C are transformed to species D.

regularly making use of tangible 'real world' models. Some of them are very simple, like molecular ball-and-stick, again some of them are highly complex such as animal disease models or model organisms. They also use 'conceptual models'. These conceptual models usually take the form of verbal descriptions of the system and are communicated by diagrams. These diagrams are usually constructed with a set of components and the ways they interact with each other. These interaction diagrams or 'cartoon models' play a central role in representing knowledge of cellular or different other processes (Ingalls (2012)).

A major drawback of these 'cartoon models' is that, while considering system behaviour, they could be significantly ambiguous. Furthermore, if there is any interaction network related to feedback. Complexity increases even further when the number of components and their corresponding interactions in the network grow. Sometimes it becomes challenging to get the intuitive understanding of the system's behaviour. A mathematical model or description of the same model can eliminate the uncertainty of the model behaviour. The mathematical model will consider the quantitative representation of the individual interaction of the cartoon model. In Figure 1.1 species A and B bind to form a new complex. The newly formed complex hinder the rate at which molecules of species C are transformed into species D. A numerical description of the process is required to quantify the interaction. Though for simple cases only equilibrium condition is enough, in many other cases binding and unbinding rates might also be required. The cartoon model or traditional knowledge cannot provide a quantitative description rather than a qualitative explanation of the molecular interac-

tion. A well-studied mechanism with sufficient data might be capable of showing the quantitative characteristics. The interaction diagram with related quantitative data can be used to develop a dynamic mathematical model. This kind of model consists of a number of equations that describe the system's behaviour over time. This behaviour is termed as "system's dynamic behaviour". These models are usually *mechanistic*, as they explain the mechanisms of molecular interaction with some laws of physics and chemistry as well as mathematics. Any of the parts of the mechanistic model actually represent the real observed system. Any change in the mechanistic model's component will also mimic to the real system. So, model simulation (*in silico* experiments) can be used to predict system behaviour. Some numerical software built with different programming languages are used for this simulation purposes.

As a mathematical model is a hypothesis, so the outcome or result of the model hypothesis is also a hypothesis. Though the real cellular behaviour cannot be predicted by simulation, it can be invaluable for further experimental design by showing the promising paths for further investigation, or by showing the inconsistencies between the real laboratory observations and our understanding of the models or systems.

1.3 The Systeome Project

'Systeome' is a collection of system profiles for all genetic variations and environmental stimuli responses. A system profile consists of a set of information about the properties of the system including structure, behaviour, analysis of results such as bifurcation diagram or phase portfolio. The structure of the system should include the structure of genes and metabolic networks and its physical structure, associated constants, and their properties (Kitano (2002)).

Systeome is not just a simple cascade map; rather it assumes different active and dynamic solutions, simulations as well as profiling of various system status. The Systeome project might be established with dealing all aspects for profiling the Systeome of yeast, *C. elegans, Drosophila,* mouse and finally human. The primary goal of the Human Systeome project is defined as - "To complete a detailed and comprehensive simulation model of the human cell at an estimated error margin of 20 percent by the year 2020, and to finish identifying the system profile for all genetic variations, drug responses, and environmental stimuli by the year 2030"(Kitano (2002)).

This is a highly ambitious project and requires several milestones. Some pilot projects will lead toward the final goal. Initial pilot projects are using yeast for the simplicity of structure and subsequent behaviour. *C. elegans* have comparatively

complex system structure and so is their behaviour. Besides such pilot projects, concurrently the Human Systeome project shall be commenced.

The futuristic impact of this project will be very wide spread as well as far-reaching. These will be the baseline and standard asset for any further biological research to provide fundamental diagnostics and prediction for a variety of medical practices. This Systeome project involves many other major engineering projects for developing the measurements, as well as software platforms.

1.4 Biological Background Related with this Thesis

In modern molecular biology, the biological systems like cells are treated as complex systems. The usual conception of the complex system is a very large number of simple but identical elements interact to generate the complex behaviour. The actual behaviour of biological systems is different from this conception. A vast number of functionally different and multifunctional group of elements act with each other selectively, perhaps non-linearly, to generate coherent behaviour. Mostly, functions of biological systems depend on a combination of the network and specific elements involved.

Development of molecular biology has discovered a large number of biological facts like sequencing genome, protein properties etc. To explain the biological system's behaviour only these developments are never sufficient. Study of cell tissues, organs, organisms also might be the systems' components to be considered. Their specific interaction which is defined by the evolution could be more supportive of reaching the prime goal of biology. Though advancement in more accurate quantitative experimental approach will continue, the detailed functional insights of biological systems may not provide the exact results from purely intuitive basis due to the intrinsic complexity of biological systems. A proper combination of experimental and computational approaches is more likely to solve this problem. In modern molecular biology, the organisational and functional activity of gene regulatory network is a key experimental and computational challenge.

1.4.1 Transcriptome and Transcriptomics

A *transcriptome* is the complete set of messenger RNA (mRNA) produced by the genome, in a specific cell or tissue type expressed by an organism under specific circumstances. One of the key characteristics of a genome is it's stability, while the activity of transcriptome is dynamic. Transcriptomic activities change over time

depending on many factors, such as change of the environmental conditions, stage of the development. Gene activity is the count the number of transcripts, which is also known as gene expression. Differentially expressed genes are identified by juxtaposition of transcriptomes in response to different treatments or in distinct cell populations. In any organisms, almost every cell contains the same genes, but the gene expression patterns might be different depending on different properties of cells. These differences are responsible for the different behaviours of divergent cells and tissues (Adams (2008)). Transcriptomic data helps to explore different gene functions. For an example, in a breast cancer cell study, an unknown gene's expressions are significantly higher than in healthy cells. It is more likely that the unknown gene is playing role in cell growth. Thus the transcriptomic data may assist the researcher by reducing the search space. *Transcriptomics* is the study of the transcriptome using high-throughput methods, such as microarray analysis, Reverse Transcription-Polymerase Chain Reaction (RT-PCR) analysis, Chromatin immunoprecipitation (ChIP) experiments and many more.

1.4.2 ChIP-chip, Microarray and Gene Expression Data for Genomics

Living cells contain thousands of genes. These genes code for one or more proteins. Expressions of these genes are regulated by many of these proteins through a very complex regulatory pathway. Usually regulation occurs to accommodate the changes of the environment, as well as at the cell cycle of the development process. In the process gene expression, information contained in the gene, synthesise to a functional gene product. The genetic code stored in the DNA is usually expressed or interpreted by gene expression which represents the phenotype. Gene expression data is usually stored in a DNA microarray or DNA chip which is also known as a biochip.

In the field of transcriptomics, ChIP (Chromatin immunoprecipitation) is a technique applied to determine the location of DNA binding sites on the genome for a particular protein of interest. Chromatin immunoprecipitation provides a broader view of the protein and DNA interactions which occur inside the nucleus. ChIP-chip (also known as ChIP-on-chip) is a technology that brings together chromatin immunoprecipitation (ChIP) with DNA microarray chip. The ChIP-chip technology is used for isolation and identification of the genome-wide location by specific DNA binding proteins (Ren et al. (2000)). ChIP-chip technology facilitated researchers to annotate functional elements. Mapping the location of protein markers with associated cite this technology also provides a better understanding of the functionality of promoters,

enhancers, repressor, insulators, etc. Chromatin immunoprecipitation is a microarray technology which isolates DNA fragments bounded by specific DNA binding proteins. Lockhart et al. (1996) used this technology to measure the concentration of DNA fragments. Tiling Arrays are a subcategory of the microarray that differs from the traditional microarrays by the nature of the probes. Tiling arrays probe intensively for sequences that exist in contiguous or adjacent regions. The ChIP-chip technology facilitates to represent the whole genome in a one dimensional series of signals. In these signals, protein-binding sites are usually expressed by peaks. Therefore, from the signals, protein-binding sites are detected by systematically recognising the peaks. Peak recognition or peak detection is a mathematical modelling challenge. A process called sonication is used to snip long genomic sequences into smaller DNA fragments. In nature this snipping process is probabilistic. Therefore, special mathematics models or tools with probabilistic assumptions are required to deal the snipping process and also the peak recognition carefully.

The main steps of ChIP-chip process are-

- Let bound transcription factor and other associated proteins bind to DNA.
- Chop the DNA sequences into small fragments by sonication.
- Isolate the DNA fragments bound by proteins by chromatin immunoprecipitation (ChIP).
- Cross-linking between DNA and protein is reversed and DNA is released, amplified by ligation-mediated polymerase (LM-PCR) chain reaction and labelled with a fluorescent dye.
- Both IP-enriched and -unenriched DNA pools of labelled DNA are hybridised to the same high-density oligonucleotide arrays (chip).

Figure 1.2 shows two Affymetrix chips which contain DNA microarray. Two Matchsticks are shown at the bottom and alongside for the purpose of size reference of a microarray. The solid-phase DNA microarray is usually a collection of ordered microscopic spots called features. Figure 1.3 shows the schema of the gene expression microarray data. On a typical Affymetrix microarray, there are 6.5 million locations (represented by columns) with millions of identical DNA strands in every location. Every strand constructs with 25 probes or bases. The microarray is rinsed and washed with a fluorescent stain. To accomplish a DNA test, two types of samples are used: one is the controlled sample and another one is the test sample. After extracting

Fig. 1.2 Gene expression data are extracted from Affymetrix microarray. Two Matchsticks are placed for reference purpose. Images at the left side and right side are the top-view and middle one is the bottom view of the microarray. Left one contains mouse genome with 430 2.0 array, while right one contains human genome with U133 Plus 2.0 array. A special scanner (i.e. GeneChip scanner) is required to scan this high-density arrays.

mRNA from DNA, copies are made from mRNA by reverse transcription. Two different fluorescents tagged with cyanide are used to differentiate between the control sample and a test sample. In general, green is used for control copy and red for test copy. Then the tagged samples are washed on the microarray. DNA is analysed based on matching with the probes on the microarray. A laser is used to glow the fluorescent molecules. After the hybridization process, a green spot represents a hybridization with the control targets only, a red spot indicates hybridization with the test targets only, yellow represent hybridization both with the control targets and test targets, while black represents no hybridization with the samples. Over the last couple of decades, these gene expression data became one of the key resources of the biologists to diagnose diseases and drug discovery, gene discovery and determining genetic variations, aligning and comparing genetic codes, biomerker development, forensic application, functional analysis and computational biology.

Using a dynamic Bayesian network Ong et al. (2002) modelled the regulatory pathway in E.coli from the time series gene expression microarray data by modelling causality, feedback loops or hidden variables. By analysing gene expression data Friedman et al. (2000) were the first to determine the transcriptional properties for Baker's yeast using a Bayesian network.

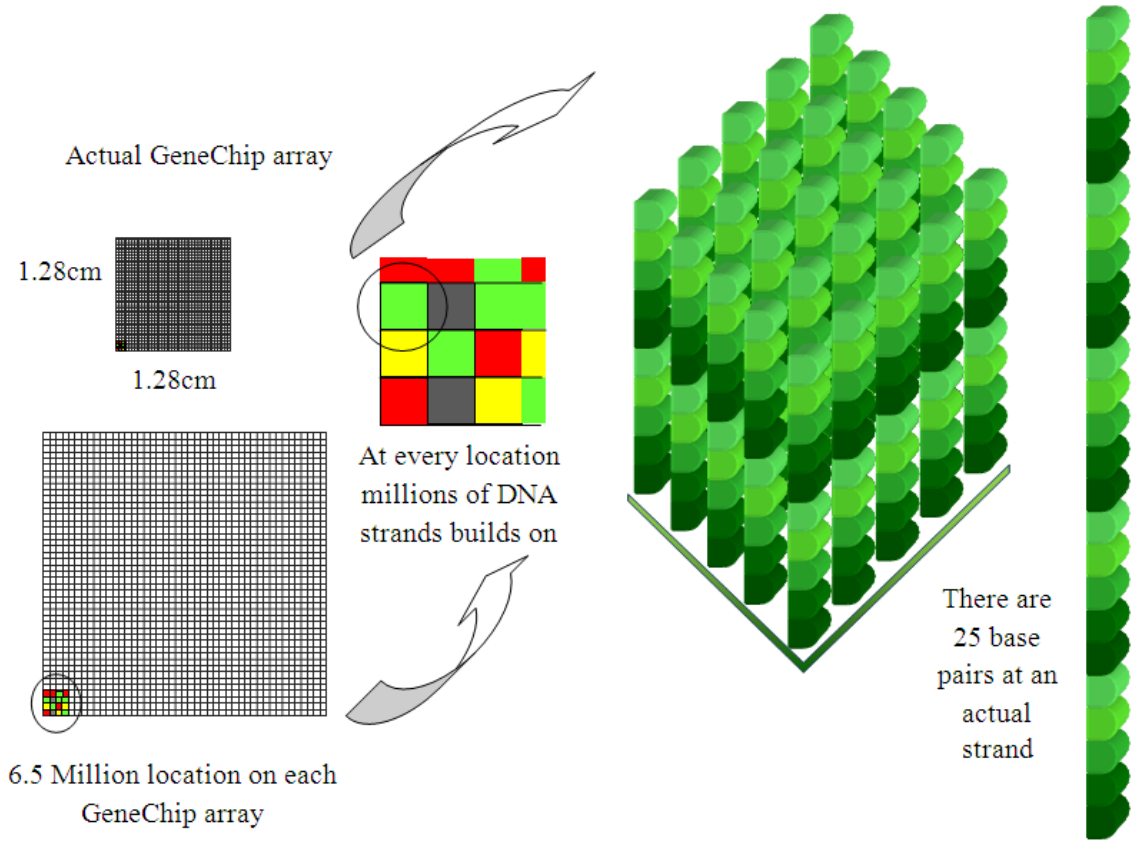


Fig. 1.3 Gene expression data: Affymetrix microarray. The dimension of a typical Affymetrix microarray is 1.28cm by 1.28cm where there are 6.5 million locations on each GeneChip array. At every location millions of identical DNA strands builds on. Every strand constructs with 25 probes or bases.

Many of the recent studies already established the fact that the gene function of the regulatory network depends on qualitative as well as quantitative aspects of the organisation of the network like high-throughput data, including genomic sequence, expression profiles and transcription factor. Among them, one of the major challenges is the quantitative measurement and analysis of the mechanisms which regulate mRNA transcription. Though using high throughput techniques it is comparatively easier to measure the output of transcription; it is experimentally very complicated to measure the protein concentration levels of transcription factors and chemical affinity to the genes. Very often transcription factors are post-transcriptionally modified. So, the actual protein concentration levels and binding affinities could be an unreliable proxy for the mRNA expression levels of transcription factors (Sanguinetti et al. (2006)).

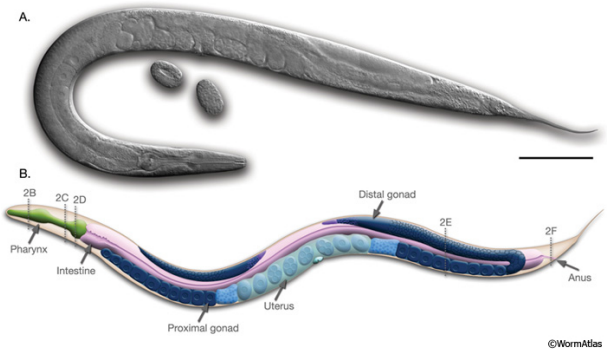


Fig. 1.4 Anatomy of an adult hermaphrodite(*C. elegans*). (a) Differential interference contrast (DIC) image of an adult hermaphrodite, left lateral side. Scale bar 0.1 mm. (b) Schematic drawing of anatomical structures, left lateral side (Courtesy of WormAtlas http://www.wormatlas.org/hermaphrodite/introduction/IMAGES/introfig1leg.htm).

Due to the advancement of the experimental technique, lot of interest in recent years has been growing to infer information about regulatory activity from target genes. Biologists are capable of acquiring the information about the structure of the transcriptional regulatory network. Lee et al. (2002) determined the transcriptional regulatory network of yeast using chromatin immunoprecipitation(ChIP). They tried to figure out how yeast transcriptional regulators bind to promoter sequences across the genome. By calculating a confidence value (P value) and setting up specific threshold, they consider the protein-DNA interactions and artificially imposes a binding or not binding binary decision for each of the protein-DNA pair.

1.4.3 *Caenorhabditis elegans*

Caenorhabditis elegans is a nonparasitic, soil dwelling, a small nematode worm. *C. elegans* and other *Caenorhabditis* species are found through all over the world. It can easily colonise mostly in the rotting materials with other micro-organisms. At the laboratory *C. elegans* is easy to maintain in the Petri dishes. At 25 °C *C. elegans* complete its life cycle in just 2.5 days from fertilised embryos to egg-laying adult through 4 larval stages. Its typical lifespan is 2-3 weeks. In 1965, Sydney Brenner introduced *Caenorhabditis elegans* as a model organism to study the behaviour and development of animal (Brenner (1974)).

C. elegans is a relatively new addition as a model organism but its biological characteristics and property already been studied to an extraordinary level. The anatomical characteristics and detail development of this nematode were facilitated by

its simple body plan. It is a eukaryote and it shares cellular and molecular structures and control pathways with higher organisms. *C. elegans* is multicellular, an adult wild type consists of 959 somatic cells and among these 302 are neurones (Palikaras and Tavernarakis (2013); Sulston and Horvitz (1977)). Its developmental process (e.g. embryogenesis, morphogenesis) goes through a complex process to develop into an adult. Monitoring of the cellular process and recording of cell division pattern is comparatively easier as its body is transparent. *C. elegans*'s complete cell lineage at the electron microscopy level has been completed. It has already been established that the cell lineage is remarkably invariant between animal to animal (Brenner (1974); Byerly et al. (1976); Sulston et al. (1980); Wood (1988)).

To elucidate pathways and processes relevant to human biology and diseases *C. elegans* has been used as a vital model. There are between ∼20,250 to ∼21,700 predicted protein-coding genes in *C. elegans* (Gerstein et al. (2010)). Using four different orthology-prediction methods, Shaye and Greenwald (2011) assayed four methods to compile a list of *C. elegans* orthologs of human genes. A list of 7,663 unique protein-coding genes resulted in that list and this represents around 38% of the 20,250 protein-coding genes of *C. elegans*. When human genes introduced into *C. elegans*, human genes replaced their homologs. On the contrary, many *C. elegans* genes can function with a great deal of similarity to human like mammalian genes. So, the biological insight acquired from *C. elegans* may be applicable to a more complex organism like the human.

1.4.4 Transcription

A number of biological functions like development, maintenance and repair of body tissues, production of energy, creation of hormones and enzymes, transportation of certain molecules and formation of antibodies take place using proteins. To perform any of the above activities or functions cells need to generate protein continuously. Inside the cell, proteins are manufactured from the DNA. When the cells are in need of protein production, a special signal is sent to the DNA using transcription factors. Then proteins reside in DNA start to manufacture depending on the received signals. The way that the enzymes find the information required for protein construction is extremely complicated.

DNA (Deoxyribonucleic acid) transcription is a process that transcribes genetic information from DNA to a complementary RNA (Ribonucleic acid). By the transcription process protein is produced from a copy of DNA. This production of proteins and enzymes are controlled by the coding of cellular activity. Even the conversion of

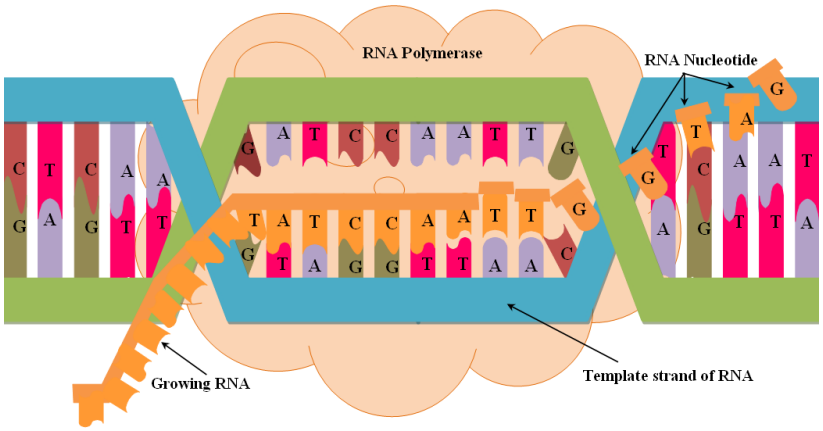


Fig. 1.5 A 'cartoon model' of DNA transcription. DNA consists of four nucleotide bases named adenine (A), guanine (G), cytosine (C) and thymine (T) that are paired together (A-T and C-G) to give DNA its double helical shape. By DNA transcription genetic information from DNA transcribe to a complementary RNA.

DNA to proteins is not straight forward. At the initial stage of transcription, an RNA polymerase reads the sequence of DNA and finally produces a complementary RNA at the end of the transcription process. DNA consists of four nucleotide bases named adenine (A), guanine (G), cytosine (C) and thymine (T) that are paired together (A-T and C-G) to give DNA its double helical shape. The major steps of DNA transcription are

RNA polymerase binding to DNA: In order to initiate the DNA transcription, RNA polymerase and sigma factor[1] form a holoenzyme[2], which binds to a specific area of the DNA named promoter region. Transcription process starts at the promoter region of a double-stranded DNA. Sigma factor can recognise the DNA and its specific promoter region.

Elongation: A sequence-specific DNA binding factor, called transcription factor unwind the DNA strand. Elongation of the transcript then continues by the RNA polymerase and a sequence of the chain is opened up. A messenger RNA (mRNA) is formed when RNA polymerase transcribes into a single-stranded RNA polymer from a single-strand of DNA.

Termination: RNA polymerase moves along the DNA unwinding its double helical form until it reaches the terminator sequence. At that point, RNA polymerase detaches from the DNA and releases the mRNA polymer. In this way, DNA double helix is

[1]Sigma factor (σ factor) is a special type of protein needed for the initiation of RNA synthesis.
[2]an active compound biochemically formed by the combination of an enzyme with a co-enzyme.

opened, transcribed and reclosed with minimum stress on the DNA molecule. At any certain time, many RNA polymerases can transcribe a single DNA sequence, which can manufacture a large quantity of protein at once.

1.4.5 Transcription Factor

A transcription factor is a protein that binds to DNA sequences and controls the flow of genetic information coding from DNA to mRNA (Karin (1990); Latchman (1997)). Transcription factors can both promote or block the transcription process and act as an activator or repressor respectively (Lee and Young (2000); Nikolov and Burley (1997); Roeder (1996)). A transcription factor may contain one or more DNA-binding domains. These binding domains attach to specific sequences of DNA adjacent to the genes that they regulate. Though some other proteins such as coactivators, deacetylases, chromatin remodelers, kinases, histone acetylases, and methylases also play crucial roles in gene regulation, yet they are not classified as transcription factors due to lack of DNA-binding domains (Brivanlou and Darnell (2002); Mitchell and Tjian (1989); Ptashne and Gann (1997)). Figure 1.7 describes the mapping (we can also say 'cartoon' mapping) between the environmental signal, transcription factors inside the cell, and the gene that they regulate. The environmental signal activates specific transcription factor. After the activation, the transcription factors bind DNA to change the transcription rate (the rate at which mRNA is produced) of specific target genes. The mRNA is then translated into protein by the process named translation (Alon (2006)).

1.4.6 Amyotrophic Lateral Sclerosis and Mouse Model

Amyotrophic lateral sclerosis (ALS), also known as Lou Gehrig's disease or Motor neurone disease (MND) is a diverse neurodegenerative disorder. The median survival of this lethal disorder is less than 5 years. The disease is heterogeneous with variable severity in terms of speed of progression of the disease course (Brockington et al. (2013); Peviani et al. (2010)). From the biological aspect, the disease progression speed is not clear yet. For experimental purpose, many of the pathological and clinical features of human ALS can be replicated very well by transgenic mice. These murine models also show the heterogeneity in the disease progression for the clinical phenotype. In a study Pizzasegola et al. (2009) reported that disease progression is much faster in $129Sv$ than $C57$ mouse strain. Genomic analysis with gene expression time series data from these murine models could be interesting to examine the speed of propagation of ALS.

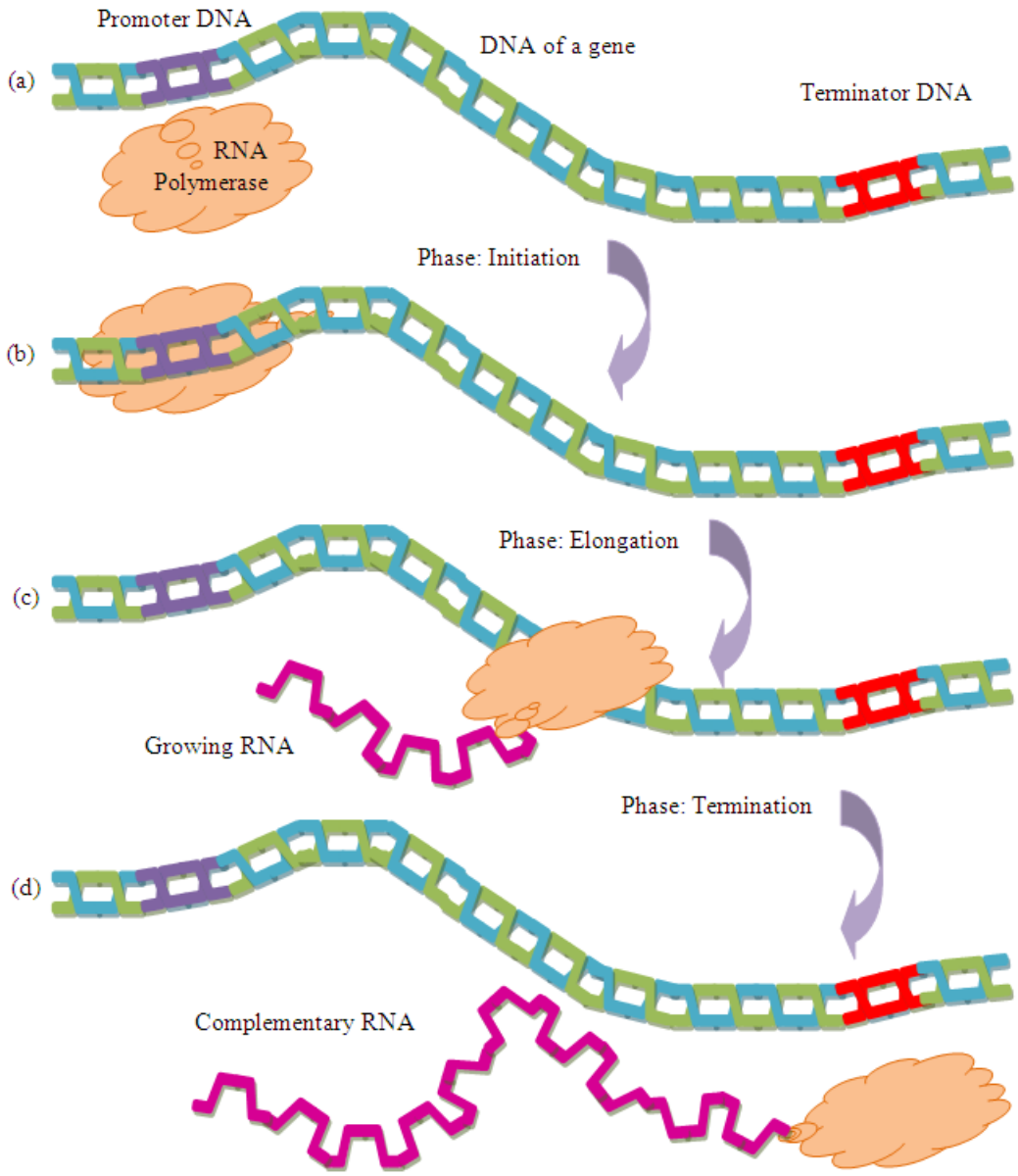


Fig. 1.6 A 'cartoon model' of transcription process: DNA transcribed in mRNA. RNA polymerase binds to a promoter region of DNA. Transcription factor unwind the DNA strand and after the elongation phase a sequence of chain is opened up. A messenger RNA (mRNA) is formed when RNA polymerase transcribes into a single stranded RNA polymer from a single strand of DNA.

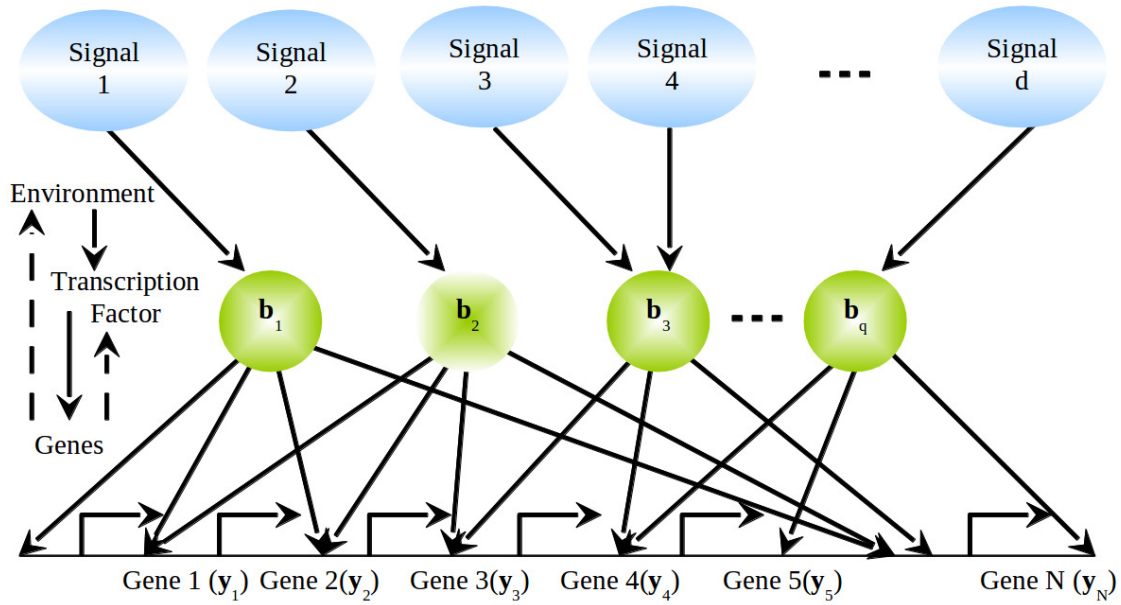


Fig. 1.7 The mapping between environmental signal, transcription factors and the genes that they regulate (Alon (2006)). Some excitation or environmental changes activates transcription factors. After the activation, the transcription factors bind to DNA and regulates the transcription rate of the target gene.

1.5 Gaussian Processes

Gaussian processes (GPs) are a general class of models of functions. GPs are one of the most widely used families of stochastic processes. As a general setting, the Gaussian process of many types has been studied and incorporated in research for decades, especially for modelling dependent data observed over time, or space, or time and space together. In GPs observations in the input space are random variables from Gaussian distributions. We included the introductory concepts of the Gaussian process in Chapter 3.

1.6 Publication Related with this Thesis

The work detailed in this thesis has been presented (as a form of poster and talk) at different International Conferences, Workshops and Summer Schools as listed below

- **Muhammad Arifur Rahman** and Neil D. Lawrence, "A Gaussian Process Model for Inferring the Dynamic Transcription Factor Activity", *International Conference on Bioinformatics, Computational Biology, and Health Informatics*, USA, October 2016.

- Sura Zaki Alrashid, **Muhammad Arifur Rahman**[3], Nabeel H. Al-Aaraji, Paul R. Heath and Neil D. Lawrence, "Clustering Gene Expression Time Series of Mouse Model for Speed Progression of ALS", *Workshop on Mathematical and Statistical Aspects of Molecular Biology*, University of Helsinki, Finland, April 2015.

- **Muhammad Arifur Rahman** and Neil D. Lawrence, "A Probabilistic Dynamic Model for Transcription Factor Activity of C. elegans", *Machine Learning Summer School and International Conference on Artificial Intelligence and Statistics*, Iceland, April 2014.

At the time of writing more developed work from these chapters is currently under consideration for publication in a peer-reviewed journal.

[3]This is a collaborative work between Sura Zaki Alrashid and Muhammad Arifur Rahman where the first authorship is shared.

1.7 Road Map

The thesis is structured in the following chapters

Chapter 1: This document starts with some basic concepts and general terminology to the field of interest to address some key issues which will be tackled or achieved later on this work.

Chapter 2: This chapter starts with the basic concepts of a latent variable model (LVM) and a probabilistic model. After describing the connectivity information between genes and transcription factors we briefly describe the probabilistic model for transcription factor activities. Earlier this problem has been solved for a unicellular microorganism (yeast), we have forwarded the mathematical model of transcription factors activity for a multicellular eukaryote (*C. elegans*) building our own connectivity information.

Chapter 3: This is a technical background chapter where we briefly describe the Gaussian process, regression problem and regression with the Gaussian process. Choice of an appropriate kernel is one of the key issues while modelling with the Gaussian process. In this chapter, we briefly describe some commonly used kernels. We also mentioned about hyperparameter learning. Why and which kernel could be an appropriate choice while modelling the transcription factor activity using Gaussian process will be justified at the later section of this chapter.

Chapter 4: We note that the TFA model with Markov property proposed by Sanguinetti et al. (2006) is a linear Gaussian model which is equivalent to a Gaussian process model with a particular covariance function. We, therefore, build a model directly from the Gaussian process perspective to achieve the same effect. In this chapter, we design a covariance function for reconstructing transcription factor activities given gene expression profiles and a connectivity information between genes and transcription factors. We introduce a computational trick, based on a judicious application of singular value decomposition, to enable us to efficiently fit the Gaussian process in a reduced 'TF activity' space.

Chapter 5: Amyotrophic lateral sclerosis is an irreversible neurodegenerative disorder that kills the motor neurones and results in death within 2 to 3 years from the symptom onset. The speed of progression for different patients is heterogeneous with significant variability. Transgenic mice from different backgrounds showed consistent phenotypic differences for disease progression. We used a hierarchy of Gaussian processes to model condition-specific and gene-specific temporal covariances. In this chapter, we develop a new clustering method that allows each cluster to be parametrized according to whether the behaviour of the genes across conditions is correlated or anti-

correlated. By specifying the correlation between such genes, we gain more information within the cluster about how the genes interrelate. This chapter also includes the gene enrichment score analysis and KEGG pathway analysis that we used for our clustering analysis results for biological validation.

Chapter 6 The final chapter concludes this thesis by summarising the achievements highlighting its novelties. It also raises some important questions that need to be considered in the future.

1.8 Notation, Symbols and Acronyms

1.8.1 Notation

The matrix $\mathbf{X} \in \mathbb{R}^{N \times m}$ represent the data space, where each row corresponds to a observed data points to a data feature or dimension. Unless otherwise defined, we denote $\mathbf{y}_i$ is the row of the data matrix and $\mathbf{y}_{:,j}$ is the column of the data matrix. $\mathbf{y}_{i,j}$ is the single scalar element from i^{th} row and j^{th} column.

$$\mathbf{X} = \overbrace{\left.\begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,m} \end{pmatrix}\right\}}^{\text{data features}} \text{data points}$$

In the above matrix there are N rows and m columns. Similarly, $\mathbf{y}_i$ represent the i^{th} row and $\mathbf{y}_{:,j}$ represent the j^{th} column of the matrix $\mathbf{Y}$.

1.8.2 Symbols

Throughout this paper, all vectors are represented with boldface lower-case symbols (e.g., x) and matrices with bold upper-case symbols (e.g., K) unless otherwise specified

$\mathbb{R}$	The set of real numbers
$\mathbf{x}$	A vector
$\mathbf{x}_i$	The i^{th} element of the vector $\mathbf{x}$
$\mathbf{x}^\top$	The transpose of the vector $\mathbf{x}$
$\boldsymbol{\theta}$	A set of hyperparameters
$\mathbf{0}$	A vector of zeros
$\text{diag}\left(\mathbf{x}^\top\right)$	A diagonal square matrix with the elements of the vector $\mathbf{x}$ along its main diagonal
$\mathbf{A}_{ij}$	The element from i^{th} row and j^{th} column of the matrix $\mathbf{x}$
$\mathbf{I}$	The identity matrix
$\lvert\mathbf{A}\rvert$	Determinant of the matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	The inverse of the matrix $\mathbf{A}$
$\mathbf{A}^\top$	The transpose of the matrix $\mathbf{A}$
$\mathcal{GP}\left(.,.\right)$	Gaussian process
$\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$	Gaussian distributions with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\mathcal{N}\left(.\lvert\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$	Probability distributions of Gaussian random variables with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\sim$	distributed according to the mentioned probability distribution
$\mathbb{E}\left[x\right]$	expectation of the random variable x
$\Gamma\left(.\right)$	Gamma function

1.8.3 Acronyms

cDNA	Complementary Deoxyribonucleic Acid
C. elegans	Caenorhabditis elegans
ChIP	Chromatin Immunoprecipitation
DIC	Differential Interference Contrast
DMI	Differential Multi Information
DNA	Deoxyribonucleic Acid
EDGEdb	C. elegans Differential Gene Expression Database
GO	Gene Ontology
GP	Gaussian process
GPLVM	Gaussian process Latent Variable Model
GPy	A Gaussian processes framework in python
KEGG	Kyoto Encyclopedia of Genes and Genomes
LLS	Log Likelihood Score
LPF	Local field potential
LVM	Latent Variable Model
mRNA	messenger Transfer Ribonucleic Acid
ORA	Overrepresentation Analysis
PCA	Principal Component Analysis
PLS	Partial Least Squares
PPCA	Probabilistic Principal Component Analysis
RBF	Radial Basis Function
RMI	Renyi Mutual Information
RMSE	Root Mean Square Error
RNA	Ribonucleic Acid
RT-PCR	Reverse Transcription Polymerase Chain Reaction
SE	Squared Exponential
SVD	Singular Value Decomposition
TF	Transcription Factor
TFA	Transcription Factor Activity
TFBS	Transcription Factor Binding Sites

Chapter 2

Probabilistic TFA of *C. elegans*

The data – information – knowledge – wisdom (DIKW) hierarchy is one of the fundamental and widely recognised hierarchy in information and knowledge literature. This hierarchy contextualises data, information, knowledge and wisdom, on one another to identify and describe the processes involved in the transformation of the lower level entity of the hierarchy to a higher level one (Rowley (2007)). The increasing availability of very high-dimensional data, with diverse characteristics and growing complexity, play a vital role in the recent advancement of machine learning techniques. Figure 2.1 shows some example of high-dimensional data from different domains, types and nature.

Data from the real world suffer from quality issues for various reasons. Acquisition errors are very likely to be included even in a controlled environment. Dealing with noise or added uncertainty of the data is troublesome. Within the constraints, probabilistic modelling is the dominant approach with added flexibility and capability to deal with uncertainty.

2.1 Motivation behind the study of TFA

In the consequences of diverse internal and external stimuli, cellular life must respond and recognise appropriately. Gene expression that is the conversion of abstract coded biological information preserved inside the DNA to a concrete physiologically active proteins is tightly regulated. With very few minor exceptions all cell types in a multicellular organism contain the same genetic information. Any individual cell type expresses only a unique subset of the total number of distinct genes for that specific organism. Differentially expressed genes are specified by unique epigenetic

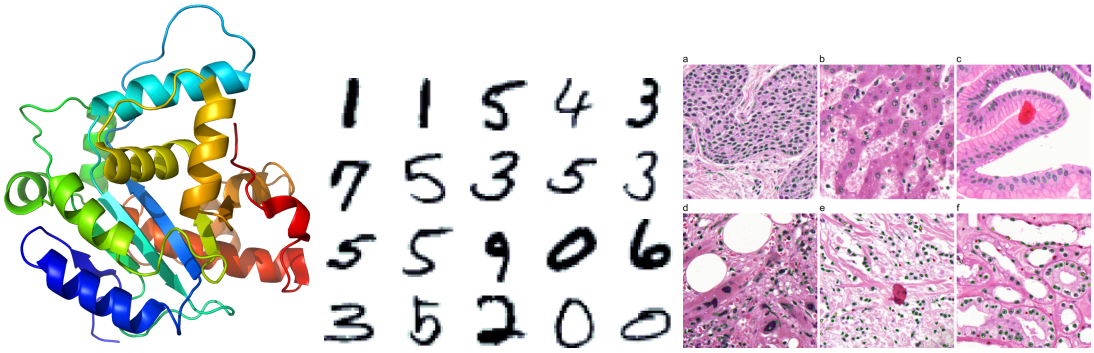


Fig. 2.1 Examples of high dimensional data: different types and nature. Left: 3D model of a protein structure. Centre: Multiple samples of hand written digits from MNIST dataset. available at http://yann.lecun.com/exdb/mnist/. Right: Multiple image patches from breast cancer, liver, gastric mucosa, bone marrow connective tissue, kidney tissue for virtual microscopy (Wienert et al. (2012)).

information. This information is present in the particular cell and which also determines its phenotype Alberts et al. (2002); Keller (1994).

For most of the genes, control at the first step of expression, transcription, is pivotal. The transcription profile is a highly preferable parameter for the recognition of a distinct cell type. In general, a complex biological regulatory network controls the differential gene expression, in which particular transcription factors relay the signals to specific target genes. Among these transcription factors, many of them are DNA-binding proteins, which bind to regulatory DNA elements located cis to the target genes.

Transcription factors play a crucial role in the transcriptional regulation of gene expression, and it allows to alter the cell growth patterns (in eukaryotes) in diverse ways Phillips and Hoopes (2008); Takahashi and Yamanaka (2006). There are two types of transcription factors; the first one is *general* transcription factors and the second one is *sequence-specific* transcription factors. *General* transcription factors, also known as *basal* transcription factors, usually acts in corporation with RNA polymerase II and transcribe a large number of genes Lee and Young (2000). The *sequence-specific* transcription factors bind to specific subsets of target genes, leading to distinct spatiotemporal structures of gene expression. Due to the complexity of transcriptional control, the critical role of this kind of transcription factors has been overlooked in many occasions Kadonaga (2004). Measuring the gene expression level systematic gene expression quantification of DNA microarray has been available for a couple of decades Schena et al. (1995). DNA microarray experiments allowed to describe the

genome-wide expression level changes both in health and disease state. The result of microarray experiments also stimulates the methodological development of diagnosis and prognosis of different diseases. The advancement of genome-wide identification of TF-binding sites instigates the development of chromatin immunoprecipitation (ChIP) followed by microarray (ChIP-on-chip, also known as ChIP-chip) Horak and Snyder (2002); Ren et al. (2000) and ChIP-sequencing (ChIP-seq) technologies Johnson et al. (2007).

The transcriptional regulatory system plays a pivotal role in controlling many biological processes, ranging from cell cycle progression Simon et al. (2001) and maintenance of intracellular metabolic and physiological balance, to cellular differentiation and developmental time courses Dynlacht (1997) by ensuring the correct expression of specific genes. A number of diseases emerge from a breakdown in the regulatory system: transcription factors are over-represented among oncogenes Furney et al. (2006), and almost one-third of human developmental disorders have been ascribed to dysfunctional TFs Boyadjiev and Jabs (2000). Even alterations in the activity and regulatory specificity of transcription factors are likely to be a primary reason for evolutionary adaptation and phenotypic diversity De et al. (2008). Indeed, recent research and study have already proved that increased sophistication of the transcriptional regulatory system seems to have been a key requirement for the emergence of metazoan life Levine and Tjian (2003). So inferring the dynamics of transcription factors activities might play a significant role to obtain a deeper insight into the gene regulatory network.

2.1.1 Why is it Complicated?

The coordinated gene expressions drive a number of cellular processes. This coordination is partially regulated by interactions between transcription factors and sequence-specific DNA elements, also known as transcription factor binding sites (TFBS). Transcriptional regulation is not an isolated process but coregulated in a highly interconnected gene regulatory network consisting of hundreds of transcription factors, their target promoters and also co-regulators Geertz and Maerkl (2010). Transcription factor binding and transcription factors activities are regulated on several stages. The initial and most fundamental order of regulation is achieved by the preferential binding of a transcription factor to specific DNA sequences. Higher orders of regulation and activities are accomplished by post-translational modifications of transcription factor domains or binding of different co-regulators. These alterations, in turn, can modulate the activity and cellular location of a transcription factor Tzamarias and Struhl (1994).

It is the specific binding of transcription factors that determine in large part the connectivity of gene regulatory networks and also the quantitative level of gene expression Gertz et al. (2009). Genetic variations in transcription factor binding sites are frequently associated with differences in transcription among individuals, highlighting the necessity of precise characterization Kasowski et al. (2010). Thus, in-depth characterization of TFA on a genome-wide level is pivotal to understand the transcriptional regulation process.

2.1.2 Why do we need to study TFA?

To build a transcriptional regulatory network, it may appear that knowledge of particular biomedical functions of transcription factors are not important. Even with some naive assumptions, it may appear that promoters or repressors regulate the transcription process similarly under similar condition. However, these assumptions can't be considered as general rules as it already reported that transcription factor DNA binding events might not follow the exact or defined biological regulatory mechanism. Such as, in a study, Turcotte and Guarente (1992) reported that in different mutants of yeast *HAP1* positive control could selectively affect different gene expressions. Using comparative genomics and functional scanning of transgenic mice Menke et al. (2008) showed how the transcription factor *TBX4* plays a pivotal role in hindlimb and vascular development. They showed a group of enhancers control the gene expresses level in different tissues. Genomic analysis also showed the relationship between transcription factor binding events and transcription factors from genes affected by the mutation. Hughes and de Boer (2013) explained even further where they reported about the role of cofactors during the transcriptional mechanism. Only understanding of condition specific activation, noise presence in the data or transcriptional cascades is not enough to grasp the actual understanding of these phenomena. Therefore, to dissect the regulatory mechanisms and gaining a better insight the complete index of transcription factor activities and its interacting partners would be invaluable.

2.2 Latent Variable Model

Latent variable models (LVMs) (Bishop (1999)) explain complex relations between multiple variables providing the connection between the variables and an underlying unobservable, i.e. latent structure. Latent variables are typically included in statistical models for different statistical concepts, including the representation of unobservable

factors/covariates, missing data, random effects, finite mixtures, variations in hierarchical data, clusters and many more. Figure 2.2 shows an analogy of latent variable model where a marionette's different movement and dynamics are observed, whereas these movements are taking place or controlled by the puppeteer. The dynamics of the puppeteer is usually unobserved.

A set of latent (hidden or directly not observable) variables $\mathbf{X}$ that can be related to the observed variables $\mathbf{Y}$ defines by a joint distribution over both. The latent space is controlled by a prior distribution $p(\mathbf{X})$ over the distribution of $\mathbf{Y}$ under the assumption of a probabilistic mapping of the form

$$y_{i,j} = f_j(\mathbf{x}_i) + \epsilon_i, \tag{2.1}$$

where $i = 1 \dots q$ and $j = 1 \dots p$, $\mathbf{x}_i \in \mathbb{R}^q$ is the latent point associated with the i^{th} observation $\mathbf{y}_i \in \mathbb{R}^p$, j is the index of the features of $\mathbf{Y}$. Inaccuracy of the model and the noise of the data is modelled by the additional noise parameter ϵ_i. Typically it is assumed that the noise has a Gaussian distribution $\epsilon_i \sim \mathcal{N}(0, \beta^{-1})$, where the term β is the precision.

We can map f of Equation 2.1 as linear and equal to a matrix $\mathbf{W} \in \mathbb{R}^{p \times q}$. Then we can rewrite as

$$y_{i,j} = \mathbf{w}_j \mathbf{x}_i + \epsilon_i, \tag{2.2}$$

where $\mathbf{w}_j$ are the rows of $\mathbf{W}$. This model recognized as probabilistic version of principal component analysis (PPCA) (Roweis (1998); Tipping and Bishop (1999)).

Given that the prior distribution over the latent variables has a Gaussian distribution, the precision β tends to infinity, PCA is recovered in the limit. The conditional probability of data given the latent space is

$$p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{W}, \beta) = \mathcal{N}\left(\mathbf{y}_i|\mathbf{W}\mathbf{x}_i, \beta^{-1}\mathbf{I}\right). \tag{2.3}$$

If we consider the data points are independent, then the marginal likelihood of the data is obtained by

$$p(\mathbf{Y}|\mathbf{W}, \beta) = \int \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{W}, \beta)\, p(\mathbf{x}_i)\, d\mathbf{x}. \tag{2.4}$$

where N is the total number of data. Even for finite precision β the maximum likelihood solution for $\mathbf{W}$ spans the principal sub-space of the data (Tipping and Bishop (1999)). This approach can be applicable for both linear (e.g. Silva et al. (2005))

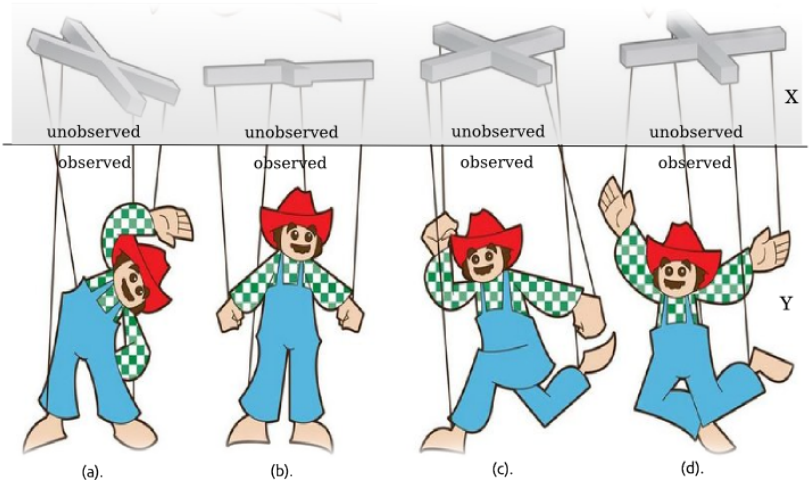


Fig. 2.2 Marionette analogy of latent variable model: Marionette's different dynamics are observed (represented by $\mathbf{Y}$). But these movements are controlled by the puppeteer's control bar which is unobserved (dynamics represented by $\mathbf{X}$).

and non-linear (e.g. GP-LVM by Lawrence (2005)) models. The classical approach while dealing with these latent variables is to marginalised them. Then other parameters are optimized using the maximum likelihood. Lawrence (2005) used an alternative but similar approach by first marginalising the parameters and then optimizing the latent variables.

2.3 Bayesian Modelling

Many facets of Bayesian modelling are widely used in machine learning for various kind of problem solutions. Bayesian models are very much dependent on two elementary probability operations. One is 'conditioning'and another is 'marginalization'. Bayes' formula has a 'double use'of the joint probability density as the product of conditional and marginal densities. Let $\mathbf{m}_1$ and $\mathbf{m}_2$ be two continuous variables and their normalized probability density is $p(\mathbf{m}_1, \mathbf{m}_2)$. By definition, the marginal probability density for $\mathbf{m}_1$ is obtained by integrating $\mathbf{m}_2$ out, we have

$$p(\mathbf{m}_1) = \int p(\mathbf{m}_1, \mathbf{m}_2)\, \mathrm{d}\mathbf{m}_2 \tag{2.5}$$

and the marginal probability density for $\mathbf{m}_2$ is obtained by integrating $\mathbf{m}_1$ out

$$p(\mathbf{m}_2) = \int p(\mathbf{m}_1, \mathbf{m}_2)\, \mathrm{d}\mathbf{m}_1. \tag{2.6}$$

The conditional probability of $\mathbf{m}_1$ for given $\mathbf{m}_2$ is

$$p(\mathbf{m}_1|\mathbf{m}_2) = \frac{p(\mathbf{m}_1, \mathbf{m}_2)}{\int p(\mathbf{m}_1, \mathbf{m}_2)\, \mathrm{d}\mathbf{m}_1} = \frac{p(\mathbf{m}_1, \mathbf{m}_2)}{p(\mathbf{m}_2)} \tag{2.7}$$

and the conditional probability of $\mathbf{m}_2$ for given $\mathbf{m}_1$ is

$$p(\mathbf{m}_2|\mathbf{m}_1) = \frac{p(\mathbf{m}_1, \mathbf{m}_2)}{\int p(\mathbf{m}_1, \mathbf{m}_2)\, \mathrm{d}\mathbf{m}_2} = \frac{p(\mathbf{m}_1, \mathbf{m}_2)}{p(\mathbf{m}_1)}. \tag{2.8}$$

From Equation 2.7 we have

$$p(\mathbf{m}_1, \mathbf{m}_2) = p(\mathbf{m}_1|\mathbf{m}_2)\, p(\mathbf{m}_2) \tag{2.9}$$

and From Equation 2.8 we have

$$p(\mathbf{m}_1, \mathbf{m}_2) = p(\mathbf{m}_2|\mathbf{m}_1)\, p(\mathbf{m}_1)\,. \tag{2.10}$$

Now from Equation 2.9 and Equation 2.10 we can write

$$p(\mathbf{m}_1|\mathbf{m}_2)\, p(\mathbf{m}_2) = p(\mathbf{m}_2|\mathbf{m}_1)\, p(\mathbf{m}_1)\,. \tag{2.11}$$

Then Bayes' rule can be obtained by

$$p(\mathbf{m}_1|\mathbf{m}_2) = \frac{p(\mathbf{m}_2|\mathbf{m}_1)\, p(\mathbf{m}_1)}{p(\mathbf{m}_2)} = \frac{p(\mathbf{m}_2|\mathbf{m}_1)\, p(\mathbf{m}_1)}{\int p(\mathbf{m}_1, \mathbf{m}_2)\, \mathrm{d}\mathbf{m}_1}. \tag{2.12}$$

where the unconditioned $p(\mathbf{m}_1)$ is called the *prior* to get the idea even before the observation of $\mathbf{m}_2$. The conditional density $p(\mathbf{m}_2|\mathbf{m}_1)$ is the *likelihood*, $p(\mathbf{m}_2)$ is the *marginal likelihood* and the conditioned density $p(\mathbf{m}_1|\mathbf{m}_2)$ is the *posterior*. The *marginal likelihood* $p(\mathbf{m}_2)$ is independent of $\mathbf{m}_1$ and used as a normalizing constant.

2.4 Modelling Transcription Factor Activities

Modelling transcription factor activities can be seen as latent variable modelling. We can observe the gene expression level, but these expression levels are regulated by protein-coding genes that bind to a specific DNA sequence and controls the production rate of mRNA. These gene expression levels are analogous to the movement of the marionette (Figure 2.2), but the transcription factor's activity is unobservable like the puppeteer's control bar which controls the gene expression levels.

In recent years an idea that has gained a lot of interest to infer the regulatory mechanism from the expression levels of genes. There has been a wealth of research on microarray data. A number of methods (Alter and Golub (2004); Gao et al. (2004); Liao et al. (2003)) aim to infer a matrix of transcription factor activities (TFAs). These TFAs can be summed up in a single number at a certain experimental point to find the concentration of the transcription factor and its binding affinity to its target genes. A variety of approaches has been proposed to infer these TFAs. For example, Liao et al. (2003) developed a data decomposition technique with dimension reduction and introduced the concept of 'network component analysis'. This method takes account of the connectivity information by imposing algebraic constraints on the factors. They argued that classical statistical methods such as principal component analysis and independent component analysis, do not consider the underlying network structure while computing low dimensional or hidden representation of a high-dimensional data sets like DNA microarray.

Alter and Golub (2004) used a dimension reduction technique (singular value decomposition) to figure out TFAs and also the correlation between DNA replication initiation and RNA transcription during the yeast cell cycle. Using multivariate regression and backward variable selection to identify active transcription factors Gao et al. (2004) targeted the same; Boulesteix and Strimmer (2005) used the partial least squares (PLS) regression to infer the true TFAs from a combination of tRNA expression and DNA-protein binding measurement. A major drawback of the methods mentioned above is that transcription factor activities do not hold any information regarding the strength of the regulators' interactivity between the transcription factors and its different target genes. It is expected that depending on the experimental conditions the transcription factor activities can vary from gene to gene. It is also expected that different transcription factors may bind to the same gene. In most cases, realistic information about the intervals may not be true as they were not based on the fully probabilistic model. Moreover, false positives are always a problem for connectivity data, typically a significant portion of ChIP data suffers from it (Boulesteix and Strimmer (2005)). Furthermore, due to the various cellular process or changes in environmental conditions the structure of the regulatory network of the cell can change considerably. Using regression-based methods, it is difficult to track these changes. Nachman et al. (2004) built a probabilistic model, using the basic framework of dynamic Bayesian networks based on discrete random variables for protein concentrations and binding affinities. Though the model was more realistic, the computational complexity for genome-wide analysis can be expensive.

2.5 Our Goal

In this chapter we will build a dynamical model that extends the linear regression model of Liao et al. (2003) and probabilistic model of Sanguinetti et al. (2006) to model the distribution of each transcription factor acting on each gene from a multicellular eukaryote (*C.elegans*). By nature, this model will be a latent variable model which will be developed based on probabilistic approach. Our first target is to construct an open source tool by implementing this approach in *R* programming language and made available by GitHub[1]. Then in Chapter 4 we will generalize the approach using Gaussian process (A Gaussian process is a collection of random variables and where the random variables have a normal distribution and it is associated with every single point in a range of times or of space. In Chapter 3 we introduce a formal description of Gaussian process.) to model the temporal changes from time-series gene expression data. The covariance structure of the transcription factors will be shared among all genes. This will lead to a manageable parameter space and will figure out useful information about the correlation of TFAs.

2.6 Probabilistic TFAs

We developed our *R* programming language based tools *ChipDyno* using the probabilistic approach of Sanguinetti et al. (2006). In the following section, we will briefly describe this approach.

The logged gene expression measurements are collected in a design matrix, $\mathbf{Y} \in \mathbb{R}^{N\times d}$ where N is the number of genes and d is the time points or number of experiments. The binary matrix $\mathbf{X} \in \mathbb{R}^{N\times q}$ is the connectivity measurements, where q is the number of transcription factors. We assume that $\mathbf{X}_{i,j}$ is '1'if transcription factor j can bind gene i, '0'otherwise.

Sanguinetti et al. (2006) used a latent variable model (as we described in Section: 2.2). TFAs were obtained by regression from the gene expressions using the connectivity information, given the following linear model

$$\mathbf{y}_n = \mathbf{B}_n \mathbf{x}_n + \boldsymbol{\epsilon_n} \tag{2.13}$$

Here $n = 1, ..., N$ indexes the gene, $\mathbf{y}_n = \mathbf{Y}(n,:)^\top$, $\mathbf{x}_n = \mathbf{X}(n,:)^\top$ and $\boldsymbol{\epsilon_n}$ is an error term. The matrix $\mathbf{B}_n$ has d rows and q columns, and models the gene specific TFAs.

[1]GitHub is a Web-based repository hosting service and source code management platform.

Different TFAs for every individual gene will increase number of model parameters drastically. This huge parameter space can be dealt through marginalization by prior distribution on the rows of $\mathbf{B}_n$. Yet, two physically plausible assumptions for selecting the prior distribution will be helpful to determine the gene specific TFAs.

- The first assumption: $\mathbf{b}_{nt}$ has the Markov property (Appendix A, Section A.4) and hence gene specific TFA $\mathbf{b}_{nt}$ at time t depends exclusively on the gene specific TFA at time $(t-1)$.

- The second assumption: the prior distribution to be stationary in time.

In order to support these assumptions, there will be two limiting cases for prior distributions. Let's first assume all the $\mathbf{b}_{nt}$ are identical for all t. Then the first limiting case is

$$\mathbf{b}_{n1} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2.14}$$

and

$$\mathbf{b}_{n(t+1)} \sim \mathcal{N}(\mathbf{b}_{nt}, \mathbf{0}). \tag{2.15}$$

If the experimental dataset comes by replicating a condition then this model is an appropriate one. The second limiting case appears when all the $\mathbf{b}_{nt}$ are independent and identically distributed (iid)

$$\mathbf{b}_{nt} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{2.16}$$

This is the case when experimental dataset comes from independent samples drawn without any temporal order.

Sanguinetti et al. (2006) expected a realistic model of time series data to be somewhere in between these two extremes (Equation 2.14, Equation 2.15 and Equation 2.16)

$$\mathbf{b}_{n(t+1)} \sim \mathcal{N}(\gamma \mathbf{b}_{nt} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}) \tag{2.17}$$

for $t = 1, ..., (d-1)$ and $\mathbf{b}_{n1} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where γ is a parameter measuring the degree of temporal continuity of the TFAs. If genes are independent for a given TFA then the likelihood function is given by

$$p\left(\mathbf{Y}|\mathbf{B},\mathbf{X}\right) = \prod_{n=1}^{N} p\left(\mathbf{y}_n|\mathbf{B}_n, \mathbf{x}_n\right). \tag{2.18}$$

Considering Gaussian noise $\boldsymbol{\epsilon}_n \sim \mathcal{N}(0, \boldsymbol{\sigma}^2\mathbf{I})$ we have

$$p(\mathbf{y}_n|\mathbf{B}_n, \mathbf{x}_n) = \mathcal{N}\left(\mathbf{y}_n|\mathbf{B}_n\mathbf{x}_n, \boldsymbol{\sigma}^2\mathbf{I}\right). \tag{2.19}$$

Factorizing the likelihood along the experiments with the assumption of spherical Gaussian noise distribution, we can rewrite the Equation 2.18 as

$$p(\mathbf{Y}|\mathbf{B}, \mathbf{X}) = \prod_{t=1}^{d}\prod_{n=1}^{N} p(\mathbf{y}_{nt}|\mathbf{b}_{nt}, \mathbf{x}_n) \tag{2.20}$$

where

$$p(\mathbf{y}_{nt}|\mathbf{b}_{nt}, \mathbf{x}_n) = \mathcal{N}\left(\mathbf{y}_{nt}|\mathbf{b}_{nt}^\top\mathbf{x}_n, \sigma^2\right). \tag{2.21}$$

Using the classical approach of latent variable model analysis, a marginal likelihood for the observations can be obtained by

$$\begin{aligned} p(\mathbf{y}_n|\sigma, \boldsymbol{\Sigma}, \boldsymbol{\mu}, \gamma, \mathbf{x}_n) = \int \prod_{t=1}^{d} \mathcal{N}\left(\mathbf{y}_{nt}|\mathbf{b}_{nt}^\top\mathbf{x}_n, \sigma^2\right) \\ \times \left(\prod_{t=2}^{d} p\left(\mathbf{b}_{nt}|\mathbf{b}_{n(t-1)}\right)\right) \mathcal{N}(\mathbf{b}_{n1}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, \mathrm{d}\mathbf{b}_{nt}. \end{aligned} \tag{2.22}$$

TFAs can be estimated a posteriori using Bayes' theorem (we briefly described Bayesian modelling using marginalization and conditioning of probability density in Section 2.3)

$$p\left(\left[\mathbf{b}_{n1}^\top, \mathbf{b}_{n2}^\top, \ldots, \mathbf{b}_{nd}^\top\right]^\top|\sigma, \boldsymbol{\Sigma}, \boldsymbol{\mu}, \gamma, \mathbf{X}, \mathbf{Y}\right) = \mathcal{N}\left(\bar{\mathbf{b}}_n, \boldsymbol{\Sigma}_{\mathbf{b}_n}\right) \tag{2.23}$$

where the posterior covariance is given by

$$\boldsymbol{\Sigma}_{\mathbf{b}_n} = \begin{pmatrix} A_1 & B & 0 & 0 \\ B & A & \ldots & 0 \\ 0 & B & \ldots & B \\ 0 & 0 & \ldots & A_d \end{pmatrix}^{-1}, \tag{2.24}$$

where

$$A_1 = A_d = \sigma^{-2}\mathbf{x}_n\mathbf{x}_n^\top + \left(1-\gamma^2\right)^{-1}\boldsymbol{\Sigma}^{-1}$$

$$A = \sigma^{-2}\mathbf{x}_n\mathbf{x}_n^\top + \left(1+\gamma^2\right)\left(1-\gamma^2\right)^{-1}\boldsymbol{\Sigma}^{-1}$$

$$B = -\gamma \left(1 - \gamma^2\right)^{-1} \boldsymbol{\Sigma}^{-1},$$

and posterior mean is given by

$$\bar{\mathbf{b}}_n = \boldsymbol{\Sigma}_{\mathbf{b}_n} \begin{pmatrix} \sigma^{-2} y_1 \mathbf{x} + \frac{1}{1+\gamma} \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \\ \sigma^{-2} y_2 \mathbf{x} + \frac{1-\gamma}{1+\gamma} \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \\ \vdots \\ \sigma^{-2} y_d \mathbf{x} + \frac{1}{1+\gamma} \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \end{pmatrix}. \tag{2.25}$$

The detail explanation of this model is available at Sanguinetti et al. (2006).

2.7 Dataset

Sanguinetti et al. (2006) did their experiments on yeast cell cycle data of Spellman et al. (1998) which is a unicellular microorganism. Our first research question is "can we step forward to find out the transcription factor activities from a unicellular microorganism to a multicellular eukaryote?".

In 1965, Sydney Brenner introduced *C. elegans* as a model organism to study the behaviour and development of animal (Brenner (1974)). It is a eukaryote and it shares cellular and molecular structures and control pathways with higher organisms. To elucidate pathways and processes relevant to human biology and disease *C. elegans* play a vital model. We provided introductory information about this model organism in Chapter 1. To find out the TFAs of *C. elegans* we had to work with three type of dataset

1. Gene expression time series data

2. A list of transcription factors

3. Connectivity information between genes and transcription factors.

2.7.1 Gene Expression Time series data

The gene expression Affymetrix single colour GeneChip data[2] on point estimate of expression level is the source of our gene expression time series data.

[2]We would like to acknowledge Professor Andrew Cossins, Institute of Integrative Biology, University of Liverpool, UK for providing us the data set with valuable information and also for the permission for further analysis of the data.

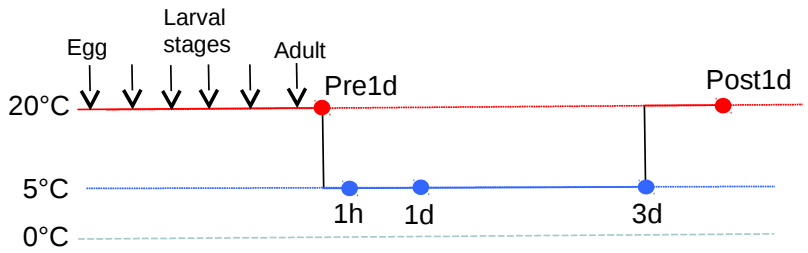


Fig. 2.3 Temperature and time settings for the gene response to chill exposure experiments: 'Pre1d '- The first experimental data within one day of *C. elegans*'s adulthood at the temperature 20 °C. '1h '- second experimental data were collected after one hour of the reduction of the temperature to 5 °C. '1d '- Third experimental data were collected after 24 hours of the temperature reduction to 5 °C. '3d '- Fourth experimental data were collected after 72 hours of the temperature reduction. 'Post1d '- The fifth experimental data were collected after one day of the rise of the temperature to 20 °C.

To extract this data we used a Bioconductor[3] package *puma* (Pearson et al. (2009)). *puma* can extract the gene expression level with estimates of uncertainty. In the wet laboratory, the experiments were done at five different stages (i.e. our gene expression time series dataset will have five time points). The main goal of the experiments in the wet laboratory was to investigate the chilling effect and identify the cold tolerance phenotype of *C. elegans*. We used the gene response to chill exposure as gene expressions. Figure 2.3 shows the temperature and time settings. In the experimental setup, all the environmental conditions apart from the temperature were same with the target of consistent result. The first experimental data was collected within one day of *C. elegans*'s adulthood at the temperature 20 °C. To measure the gene response to chill exposure the temperature was reduced to 5 °C and second experimental data was collected after one hour of the reduction of the temperature. Third experimental data was collected after 24 hours (1 day) of the temperature reduction. Fourth experimental data was collected after 72 hours (3 days) of the temperature reduction. After the fourth data collection the temperature was brought back to 20 °C. The fifth or final experimental data was collected after one day of the rise of the temperature. In the wet lab, the full experiment was repeated twice maintaining the similar setup. So, three independent replicates of the experiments were available.

Figure 2.4 shows few examples of gene expression time series data extracted from Affymetrix Microarray data. We annotated the Affymetrix ProbeID and found the

[3]The Bioconductor project is an open source software framework to assist biologists and statisticians working in bioinformatics

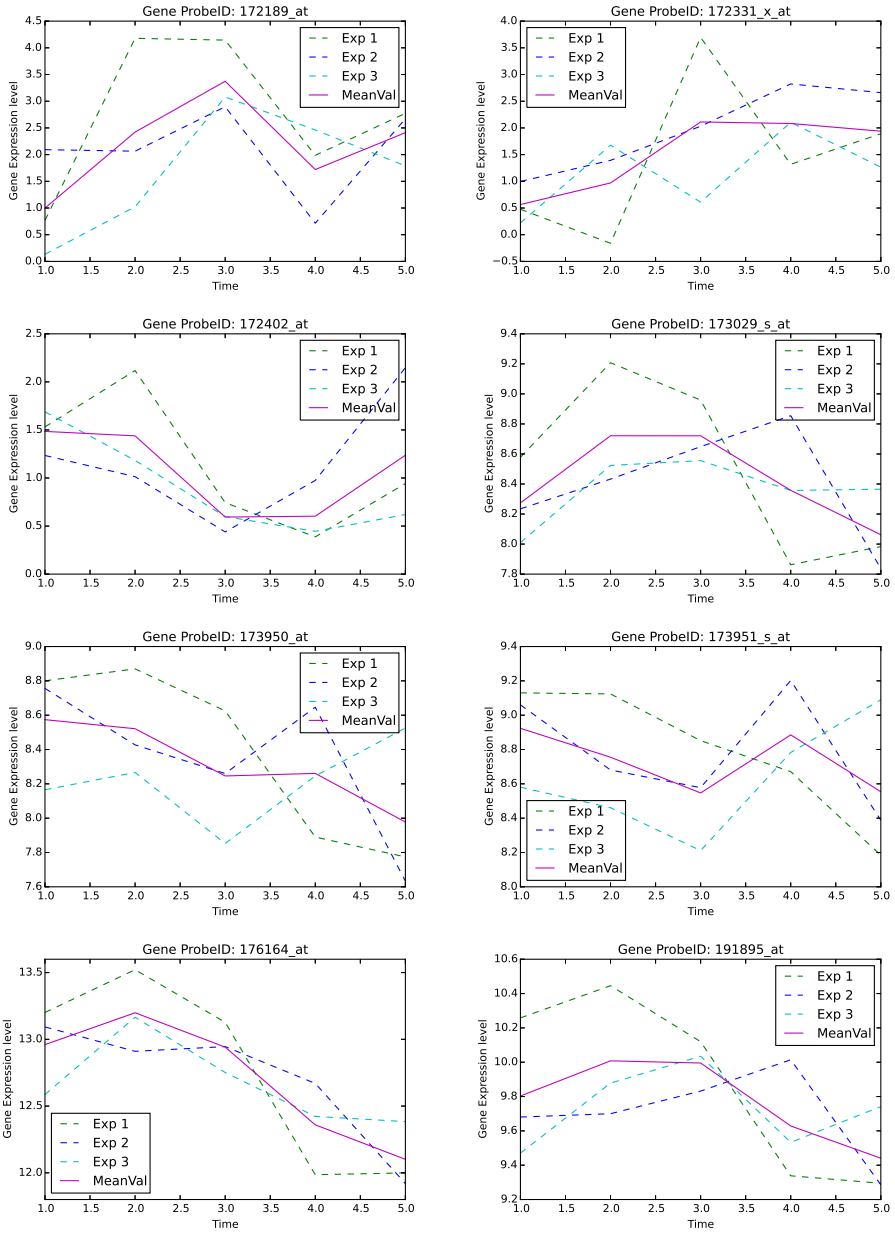


Fig. 2.4 Gene expression time series data extracted from Affymetrix Microarray data. We can annotate the Affymetrix ProbeID and find the related Genes symbols. Here in any individual plot, the title of each figure shows the Affymetrix ProbeID, x-axis represents time and y-axis represents the extracted gene expression level. The dotted lines represent the gene expression over time. We had three replications of data obtained from three separate experiments with the same experimental condition setup. The solid line represents the mean value of the gene expression level.

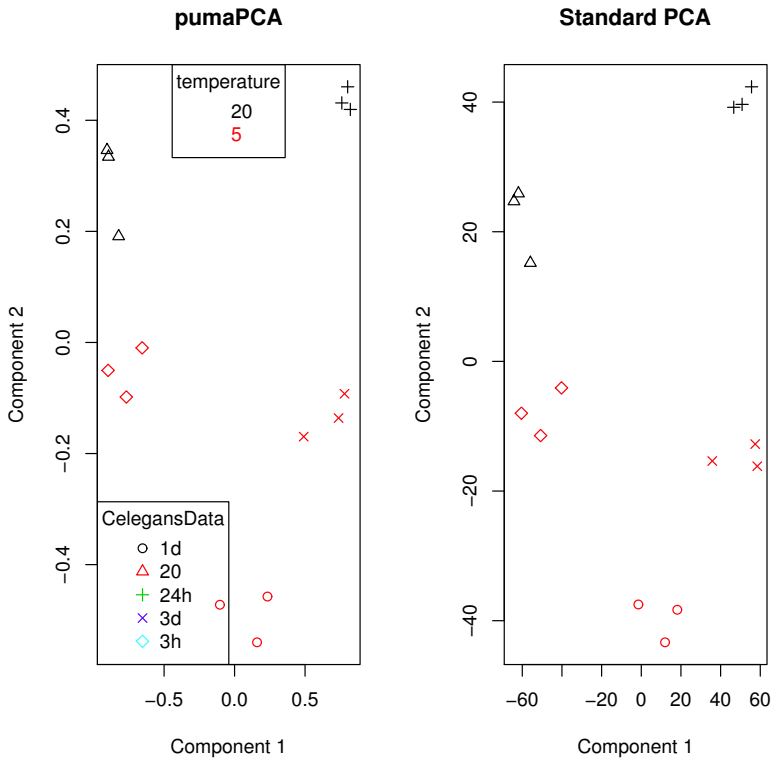


Fig. 2.5 Principal component analysis of gene expression time series data. Figure on the left shows the *puma* PCA which incorporates the uncertainty of the point expressions. Figure in the right shows the standard PCA which does not include any uncertainty of the point-expressions. It appears from these two figures that both of them have a very similar representation. This means that the uncertainty of the point-expressions does not have a significant effect on point expressions, it also proves the higher quality of the data.

related Genes symbols. As we had three replications of data from three separate experiments, we extracted them separately and termed as experiment 1, 2 and 3. The solid line represents the mean value of the gene expression level.

Figure 2.5 shows the PCA analysis of the gene expression time series data. The *puma* PCA incorporates the uncertainty of the point expressions and the standard PCA does not include any uncertainty of the point expressions. Both figures have a very similar representation. This means that the uncertainty of the point expressions does not have a significant effect on point expressions. As the added uncertainty does change the standard PCA a lot, we can say the quality of the data is very high.

2.7.2 Transcription Factors

From different data sources, we found different number/list of transcription factors for *C. elegans*. Inmaculada et al. (2007) built a database named *C. elegans* differential gene expression database (EDGEdb), which contains the sequence information about 934 predicted transcription factors and their DNA binding domains. Initially, we took these 934 transcription factors for our baseline experimental setup, but tool *ChipDyno* can deal with any number of transcription factors depending on the requirement/update of the sequence information of transcription factors.

2.7.3 Connectivity Information

Network motifs are the simplest units of transcriptional regulatory network's architecture. A particular regulatory mechanism such as positive and/or negative feedback loop can be well studied by these network motifs. Network motifs can grow in numbers and complexity based on size and nature. Autoregulation, multi-component loop, single input, multiple inputs, feedforward and regulators chain are some of the simplest and well-known network motifs. Figure 2.6 shows their representation. Xie et al. (2005) used motif conservation information for higher organisms like human, dog, rat and mouse. For promoter analysis, they considered a number of network motif (also known as transcription factor binding sites) and also some new motifs. These type of data, termed as *connectivity data* by Liao et al. (2003), provide information about whether a certain transcription factor can bind the promoter region of a gene or not.

WormNet (2015) is a gene network of protein-encoding genes for *C. elegans* based on probabilistic function and modified Bayesian integration. They have considered 15,139 genes and 999,367 linkages between genes associated with a log-likelihood score (LLS). These measured scores represent a true functional linkage between a pair of

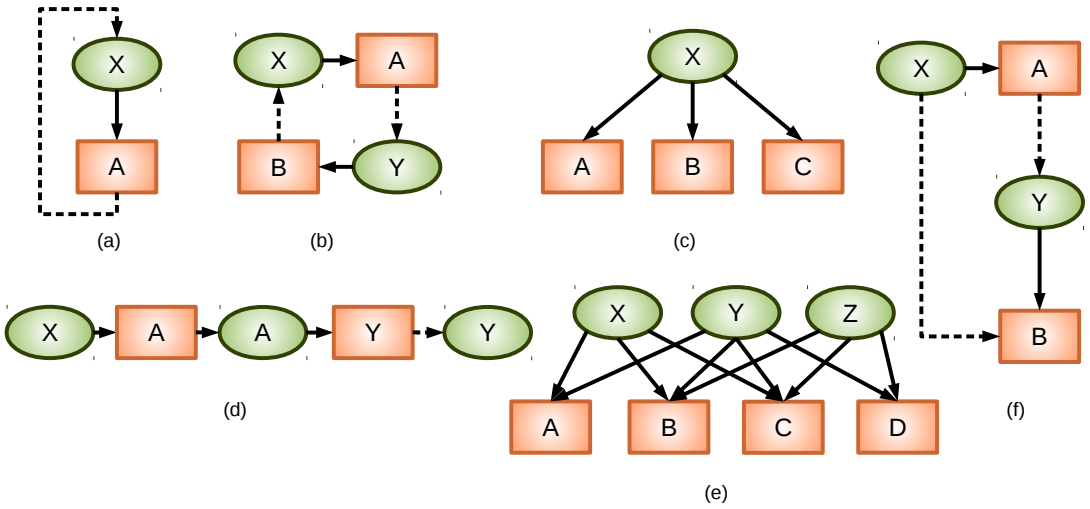


Fig. 2.6 Basic types of network motif: (a) autoregulation, (b) multi-component loop, (c) single input motifs, (d) regulators chain, (e) multiple inputs and (f) feedforward motifs. Here green ovals represents regulators and orange rectangles represents gene promoters. Solid lines represent binding of a regulator to a promoter while dashed arrow represents gene encoding regulators binding their respective regulators.

genes (Lee et al. (2007)). The linkage between two genes was measured based on the evidence codes shown in Table 2.1.

We constructed the connectivity matrix between genes and associated transcription factors from the gene to gene linkage and log-likelihood scores. Initially, we chose co-expression among worm genes (CE-CX), high-throughput yeast 2-hybrid assays among worm genes (CE-YH), literature curated human protein physical interactions (HS-LC) and high-throughput yeast 2-hybrid assays among human genes (HS-YH) to start our experiments. If needed we can consider any of the evidence to reconstruct the connectivity matrix. From the gene list, we picked the protein-coding genes (i.e. transcription factors) and later binarized it. If there is an associated LLS value between a gene and a transcription factor, we set the value '1'and '0'otherwise.

2.8 Result Analysis

We developed *R* programming language based implementation of the *ChipDyno* algorithm to identify the quantitative prediction of regulatory activities of the gene specific TFA through posterior estimation. The *ChipDyno User Guide*[4] explains different

[4] *ChipDyno User Guide* is available at GitHub.

Gene1 - Gene2	Evidence for interaction
CE-CC	Co-citation of worm gene
CE-CX	Co-expression among worm genes
CE-GN	Gene neighbourhoods of bacterial and archaeal orthologs of worm genes
CE-GT	Worm genetic interactions
CE-LC	Literature curated worm protein physical interactions
CE-PG	Co-inheritance of bacterial and archaeal orthologs of worm genes
CE-YH	High-throughput yeast 2-hybrid assays among worm genes
DM-PI	Fly protein physical interactions
HS-CC	Co-citation of human genes
HS-CX	Co-expression among human genes
HS-DC	Co-occurrence of domains among human proteins
HS-LC	Literature curated human protein physical interactions
HS-MS	human protein complexes from affinity purification/mass spectrometry
HS-YH	High-throughput yeast 2-hybrid assays among human genes
SC-CC	Co-citation of yeast genes
SC-CX	Co-expression among yeast genes
SC-DC	Co-occurrence of domains among yeast proteins
SC-GT	Yeast genetic interactions
SC-LC	Literature curated yeast protein physical interactions
SC-MS	Yeast protein complexes from affinity purification/mass spectrometry
SC-TS	Yeast protein interactions inferred from tertiary structures of complexes

Table 2.1 Gene linkage evidence code from WormNet (2015).

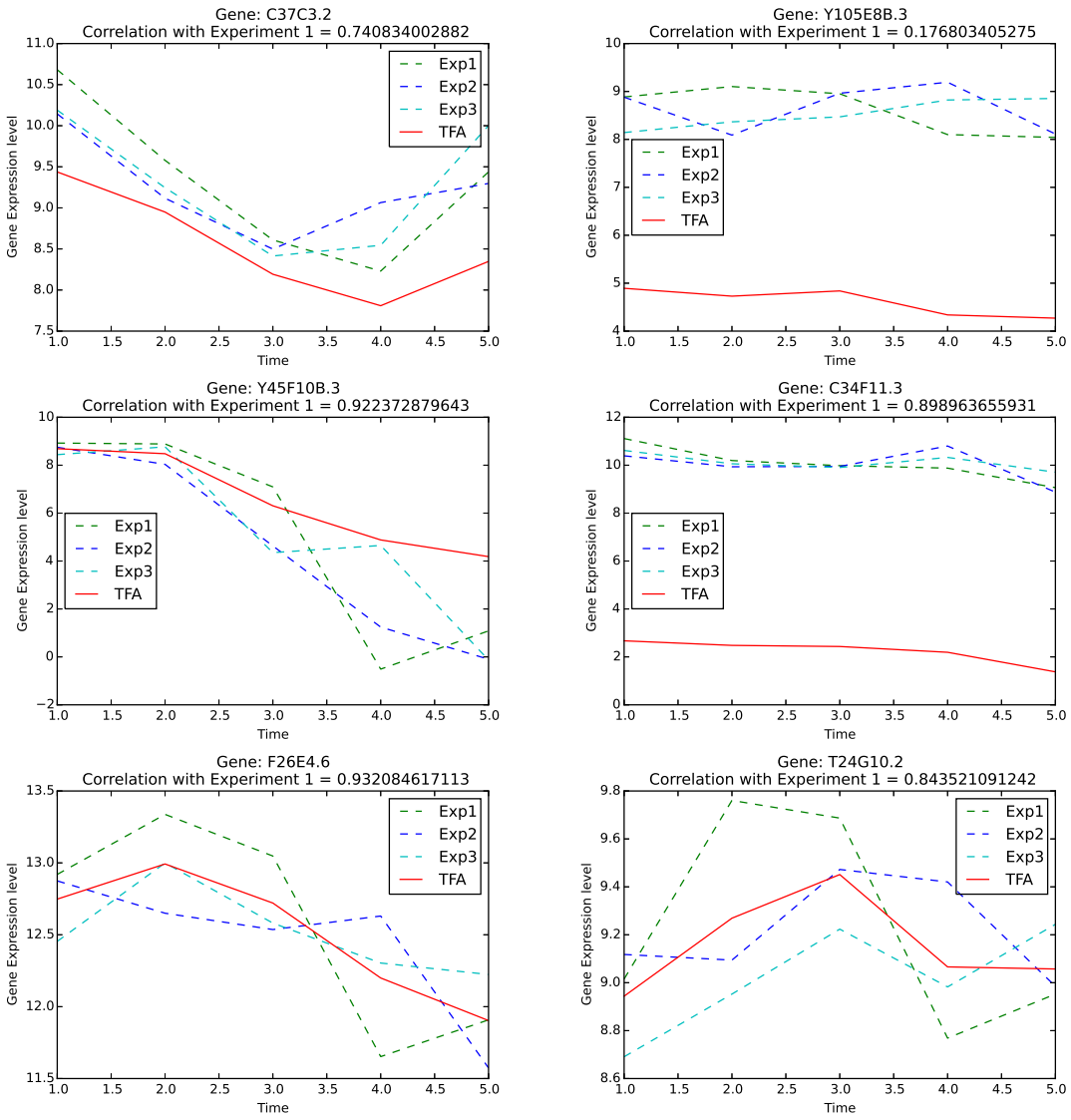


Fig. 2.7 Inferred transcription factor activity of ZK370.2 from gene expression time series data: x-axis represents time and y-axis represents the level of gene expression. The dotted lines represent the gene expressions over time. We had three replications of data obtained from three separate experiments with the same experimental setup. The solid line represents the inferred transcription factor activity. The title of each figure showing the name of the gene and correlation between gene expression level (experiment 1) and transcription factor's activity on that specific gene.

functionality of this tool and working pathway. There is no established benchmark or baseline, nor a known ground truth to compare with our results of gene specific TFA for *C. elegans*.

According to WormNet (2015) the number of genes of *C. elegans* is 15,139 and Inmaculada et al. (2007) presented 934 transcription factors. All the network motifs, i.e. autoregulation, multi-component loop, feed forward loop, single input, multi-input motif and regulator chain were visible for transcription factor activity. So it was a mammoth task to choose all the transcription factors and show their activity. Rather we chose some random transcription factors and find out their activity on different genes.

As a random example, we chose transcription factor ZK370.2 and inferred its activity on different genes. Figure 2.7 shows that transcription factor ZK370.2 can regulate C37C3.2, Y105E8B.3, Y45F10B.3, C34F11.3, F26E4.6 and T24G10.2. We investigated the correlation between the inferred transcription factor activity with gene expression level. The title of each figure (in Figure 2.7) shows the name of the gene and correlation between gene expression level (experiment 1) and transcription factor's activity on that specific gene. We noticed for few cases the correlation was quite high (i.e. 0.932 for gene F26E4.6), but for some genes, the correlation was significantly low (i.e. 0.176 for Y105E8B.3). We had three replicates of same experimental setup and their outcomes. We performed our *in-silico* experiments for individual replicates and collected the results. Later we presented all the outcome together by plots as shown in Figure 2.8. From our experimental results we can say that the dynamics for some of the gene specific regulations (i.e. F26E4.6 and T24G10.2) are very flat and not that much informative, but for some genes, TFAs varies notably over time (i.e. C37C3.2 and Y45F10B.3). These are the genes which are regulated significantly by this transcription factor. For some cases (e.g. gene C34F11.3) the error bar is quite high. These are the examples of bindings where the regulation is insignificant or false positive. The magnitude of TFA also differs from one to another. We picked another random transcription factor T20B12.8.3. Figure 2.9 shows its activity on different genes.

We have also investigated the impact of gene knock-out and how can it play a role to infer the gene-specific transcription factor activity. As we mentioned earlier, transcription factor ZK370.2 can regulate gene C37C3.2, Y105E8B.3, Y45F10B.3, C34F11.3, F26E4.6 and T24G10.2. So, we considered gene expressions of all these genes while inferring the gene-specific transcription factor activity of ZK370.2. Then for our investigation, we knock-out a gene (C37C3.2 in this example) and inferred the

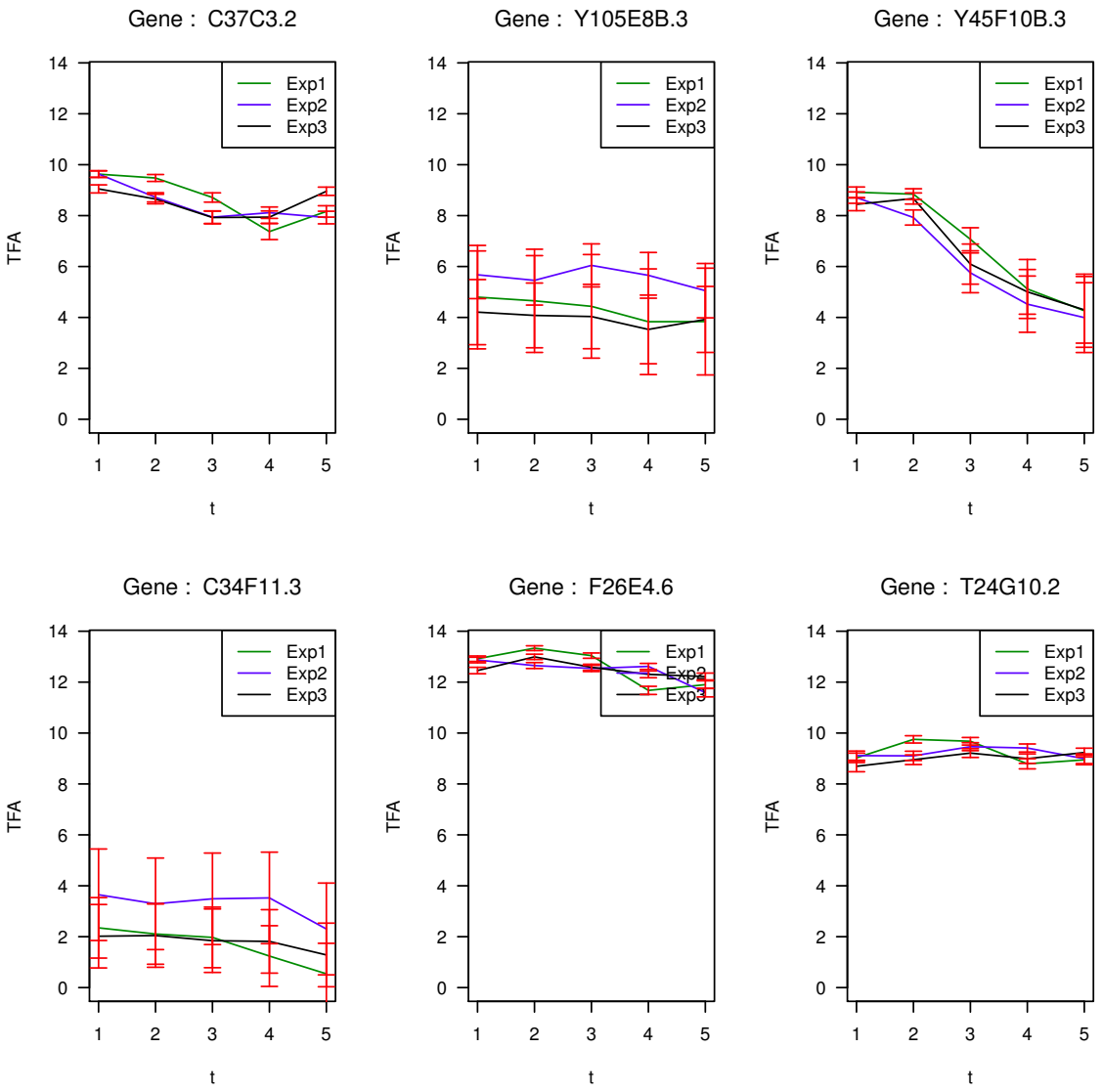


Fig. 2.8 Gene specific transcription factor activity of ZK370.2 on (top left to right) C37C3.2, Y105E8B.3, Y45F10B.3 and (bottom left to right) C34F11.3, F26E4.6, T24G10.2. x-axis represent the time stage of the experiments, and y-axis represent the gene expression level for transcription factor activities. Three different lines represent TFA for three replicates and red vertical lines are error bars.

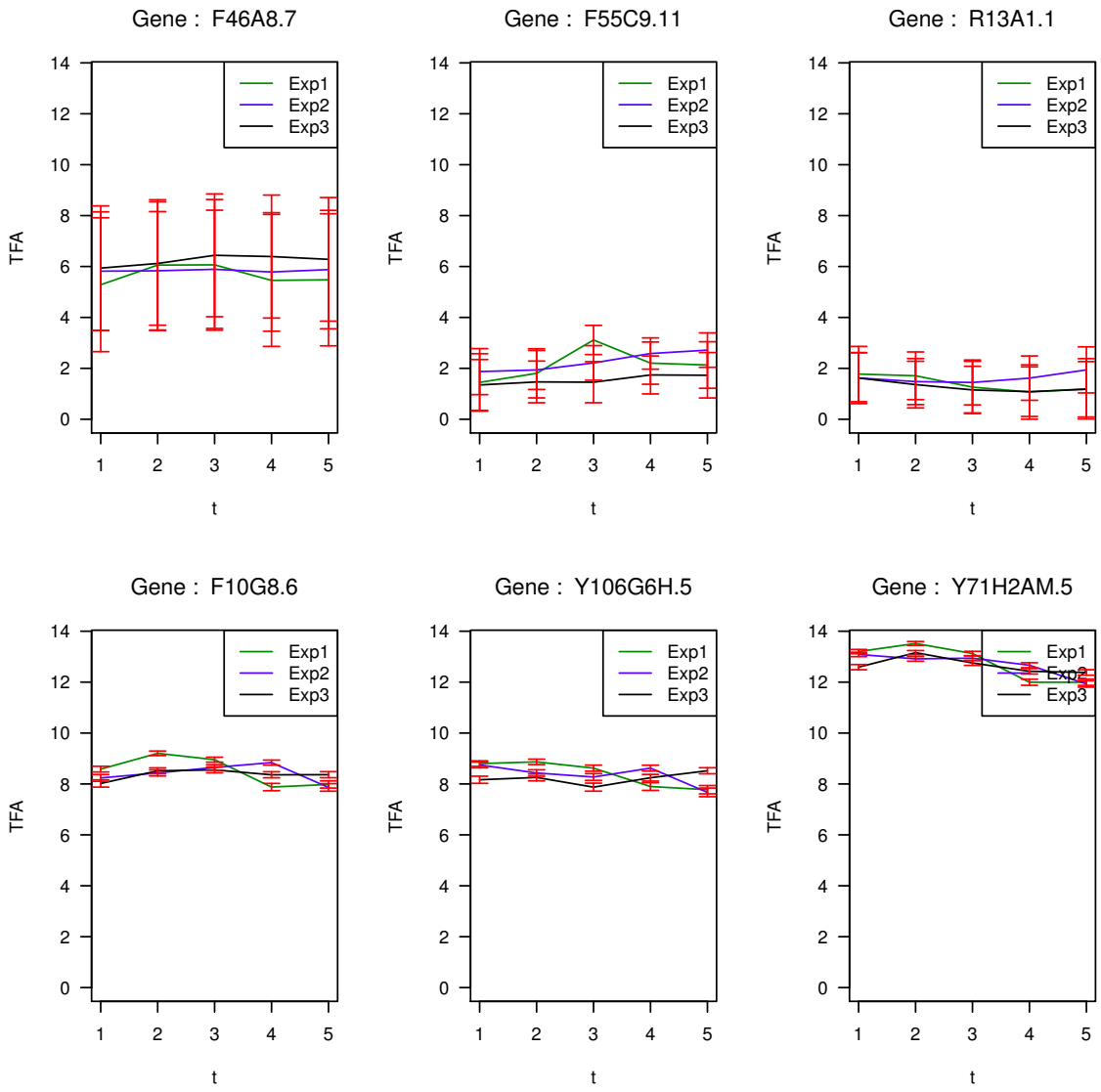


Fig. 2.9 Gene Specific transcription factor activity of T20B12.8.3 on (top left to right) F46A8.7, F55C9.11, R13A1.1 (bottom left to right) F10G8.6, Y106G6H.5 and Y71H2AM.5. x-axis represent the time stage of the experiments, and y-axis represents the gene expression level for transcription factor activities. Three different lines represent TFA for three replicates, and red vertical lines are error bars.

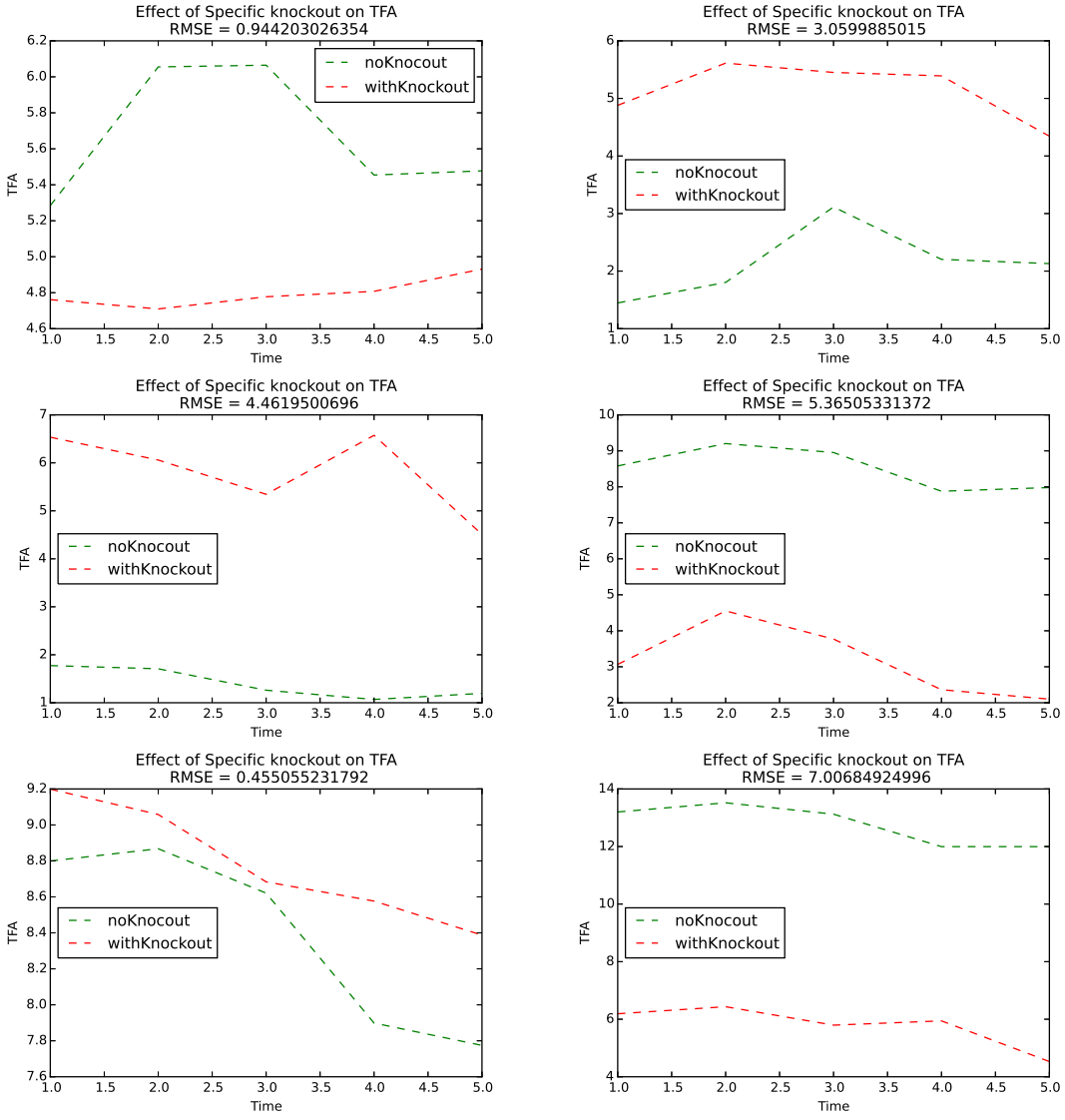


Fig. 2.10 For a specific transcription factor (ZK370.2) effect of gene knockout on other genes. ZK370.2 can transcribe a number of genes, and one of them is C37C3.2. Here we inferred the transcription factor activates of other genes before and after knockout of gene C37C3.2. x-axis represents time, and y-axis represents transcription factor activates. The title of each figure shows root mean square error (RMSE) between the transcription factor activities before and after gene knockout.

Gene Name	Regulators activity
C44B12.5	Y116A8C.35 = 1.719797 ± 3.493205, F33A8.3 = 1.415785 ± 3.492985
Y105E8B.3	Y54G2A.1 = 0.07157665 ± 1.2222137 F33D11.12 = 0.03861905 ± 0.7252534 ZK370.2 = $-1.20157055 \pm 2.0318513$
Y105E8B.3	T20B12.8 = 0.25474933 ± 2.5665869 F33A8.3 = 0.11619828 ± 3.5107742 Y116A8C.35 = 0.03289664 ± 3.8071374 F11A10.2 = 0.03016348 ± 1.7737585 C16A3.7 = 0.01883489 ± 0.9431105

Table 2.2 Example of genes regulated by multiple transcription factor.

transcription factor activity again. Figure 2.10 shows the impact of gene knock-out on gene-specific transcription factor activates. We have also measured the root mean square error (RMSE) between the transcription factor activities for a specific gene before and after gene knockout. Titles of the Figure 2.10 shows the value. Coregulated gene expression might be the main reason for varying these root mean square errors from a very low value to high one. So, if two genes are coregulated, and we knock out one of them, then we can expect a higher root mean square error. While if there is no relation between two genes then we can expect a lower root mean square error.

2.8.1 Gene With Multiple Regulators

For the case of a multi-input motif, a single gene could be regulated by multiple transcription factors. Our developed tool can determine a posteriori the relative weight for the different transcription factors regulating the genes. Table 2.2 shows some examples. Such as, gene C44B12.5 can be regulated by transcription factor Y116A8C.35 and F33A8.3. While gene Y105E8B.3 is regulated by T20B12.8, F33A8.3, Y116A8C.35, F11A10.2 and C16A3.7. The expression level can be determined by using the posterior variance. Though for some cases the expression level is quite low and noise margin is significantly high (examples of binding with insignificant regulation), we can find the most significant one by ranking the transcription factor activities. We can also rank these genes using the ranking method proposed by Kalaitzis and Lawrence (2011) to rank the differentially expressed gene expressions.

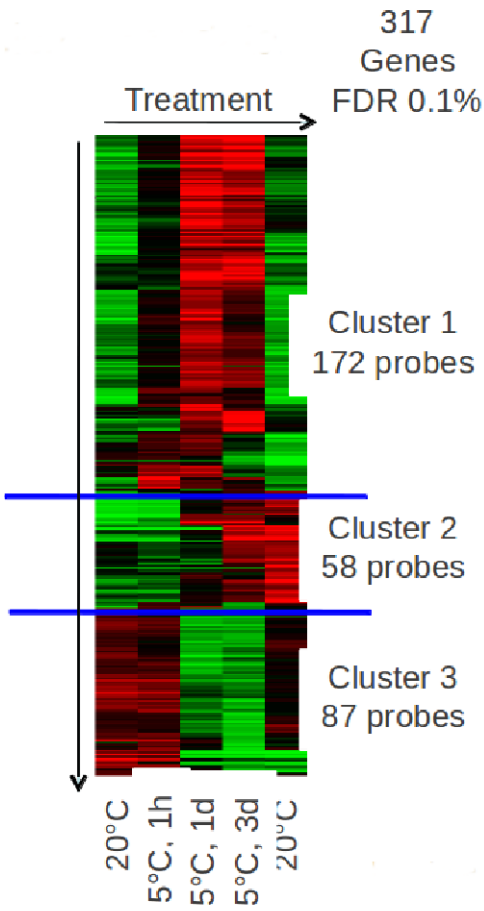


Fig. 2.11 Clustering gene expression data from microarray sample: Each row corresponds to a single gene and each column corresponds to a microarray sample. This is an ordered representation of rows and columns.

2.8.2 Different Clusters And Related Active TF

Clustering of genes is used to identify a set of genes with similar behaviour (i.e. similar expression level or pattern) over a set of experiments (Eisen et al. (1998)). Clusters provide an intuitive way to visualise the data and also help to facilitate the functional annotation of the not yet characterised genes. If an uncharacterized gene belongs to a cluster, then it could possibly have similar function and may be dominated by genes of same the function (Pe'er (2003)). Cossins et al. (2007) performed cluster analysis of genes based on different phenotypes. They constructed the basic clusters with the following phenotype

Cluster 1 - Chill upregulated: related with cell morphogenesis, cell growth, regulation of cell size, electron transport regulation of cell growth, generation of precursor metabolites and energy, anatomical structure morphogenesis, cellular metabolic process, proteolysis, etc.

Cluster 2 - Chill late upregulated: related with chromosome organisation and biogenesis, DNA packaging, chromatin architecture chromatin modification, negative regulation of developmental process, chromatin remodelling, regulation of developmental process, DNA metabolic process larval development, organelle organisation and biogenesis, postembryonic development, etc.

Cluster 3 - Chill downregulated genes: related with amino acid and derivative metabolic process, carboxylic acid metabolic process, organic acid metabolic process, fatty acid metabolic process, amino acid metabolic process, monocarboxylic acid metabolic process, etc.

Rest of the genes were placed in the group 'others'.

Clusters	**Active TF**
1. Chill upregulated	**6**
2. Chill late upregulated	**245**
3. Chill downregulated	**128**
4. Others	**203**

Table 2.3 Active transcription factor on different clusters.

Figure 2.11 shows the heat map generated from DNA microarray data to reflect the gene expression values at different temperatures and their basic clusters (Cossins et al. (2007)). Based on the above clusters, we also find out the active transcription factors active for different clusters. Table 2.3 shows the number of active transcription factors for each clusters. For further analysis, we can present the full list.

2.9 Ranking Differentially Expressed Gene Expressions

Kalaitzis and Lawrence (2011) analysed time series of gene expressions and filtered the quiet or inactive genes from the differentially expressed genes. They developed their model considering the temporal nature of data using Gaussian processes. We have used their model to rank our time series gene expressions and ranked them accordingly.

We ranked the three replicates of our data separately and later determine the Pearson correlation between the ranking score of different samples.

Figure 2.12 shows the Pearson correlation between ranking scores from different samples. The correlation coefficient for all three relations (between sample 1 and sample 2, sample 2 and sample 3 and sample 3 and sample 1) was quite high. This indicates the similarity of differentially expressed genes and quiet genes of different samples or replication of time series data. So, if required, based on these ranking we can easily filter out some of the quiet genes and keep the other genes for further experiments.

2.10 Discussion

In this chapter, in the beginning, we have defined the latent variable model. Then we described the probabilistic model of Sanguinetti et al. (2006) as the basis of our tool *ChipDyno*.

We have reimplemented the tool *ChipDyno* using *R* programming language with the aim to make it publicly available through GitHub. We used the mathematical model of Sanguinetti et al. (2006) which integrate the connectivity information between genes and transcription factors, and transcriptomic data. The probabilistic nature of the model can determine the significant regulations in a given experimental condition.

Earlier the model was developed for a unicellular microorganism (yeast) but we have successfully determined the gene-specific transcription factor activity for *C. elegans*, a multicellular eukaryote. We were also successful to filtered out the quiet genes from the differentially expressed genes.

To elucidate pathways and processes relevant to human biology and disease *C. elegans* is using as a vital model. Different orthology prediction methods (Shaye and Greenwald (2011)) are using to compile a list of *C. elegans* orthologs of human genes. Already a list of 7,663 unique protein-coding genes has resulted in that list and this represents 38% of the 20,250 protein-coding genes predicted in *C. elegans*. When human genes introduced into *C. elegans* human genes replaced their homologous. On the contrary, many *C. elegans* genes can function with a great deal of similarity to human like mammalian genes. So, the biological insight acquires from *C. elegans* may be directly applicable to a more complex organism like the human.

Lots of computational approaches on gene expression data for time series analysis are not well suited where time points are irregularly spaced. Even in commonly used state-space model time points must occur at regular intervals. On the other side gene

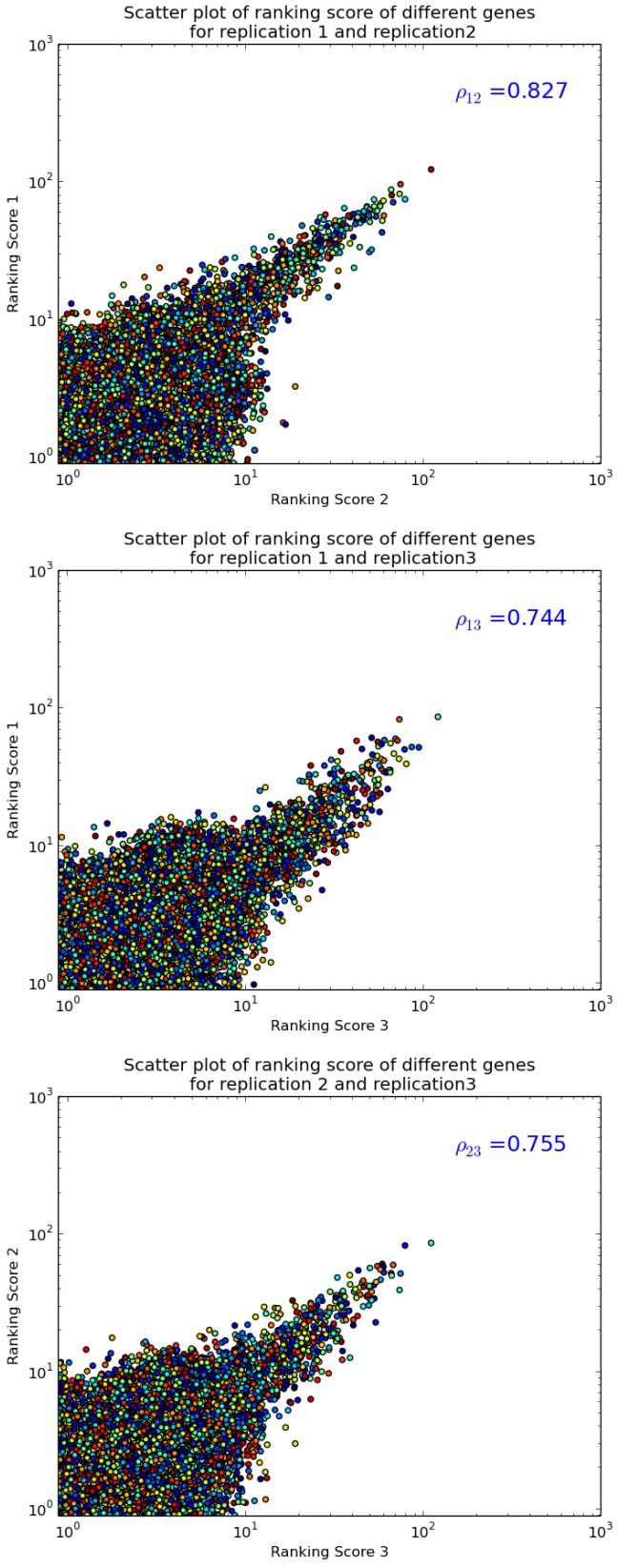


Fig. 2.12 Pearson's correlation between different ranking scores. For each figure number at the top right position represent Pearson's correlation.

expression experiments with regular samples may not be cost effective or optimal from the perspective of statistics. It is expected that models with irregular time points might be more informative if the time points are selected considering some temporal features. The Gaussian process is not restricted to equally spaced time series data. Already Gaussian process regression has been successfully applied to overcome this issue and analyse time series data (Kalaitzis and Lawrence (2011)). So our expected model will overcome the restriction of temporal sampling of equally spaced time intervals.

The probabilistic dynamical model for quantitative inference of TFA that we described in this chapter will be used as our basis for the coming chapters. We will introduce Gaussian process at Chapter 3. In Chapter 4 we will devise a mechanism to overcome the limitations of this mechanistic model using Gaussian process.

Chapter 3

Gaussian Process Regression

3.1 Brief History of Gaussian Process

The Gaussian process is one of the most widely used families of stochastic processes for modelling dependent data observed over time, or space, or both together. As a general setting, Gaussian processes of many types have been studied and incorporated in research for decades. The Wiener process (e.g. Papoulis (1991), one of the best known Lévy processes) is a particular type of Gaussian process. The story of using Gaussian process is still a long one. Kolmogorov (1941) and Wiener (1949) first used Gaussian process for time series prediction in the 1940's, probably the history of the Gaussian process is even older. Indeed the Brownian motion is a Gaussian process too. This is because the distribution of a random vector is a linear combination of vectors which follow a normal distribution. Einstein (1905, 1926) investigated the theory of the Brownian movement. This is Gauss-Markov process. Thorvald N. Thiele was the first to propose the mathematical theory of Brownian motion. He also introduced the 'likelihood function' during the period 1860-1870.

Since the 1970's Gaussian process have been widely adopted in the field of meteorology and geostatistics. Around that time Gaussian process regression was named as kriging and used by Matheron (1973) for prediction in geostatistics. O'Hagan (1978) used Gaussian process in the field of statistics for multivariate input regression problems. For general purpose function approximators, Bishop (1995) reviewed neural networks, Neal (1996) showed the link between Gaussian process and neural networks and in the machine learning context, Williams and Rasmussen (1996) first described Gaussian process regression.

Over the last two decades, the Gaussian process in machine learning has turned to a major interest and much work has been done. Perhaps Rasmussen and Williams

(2006) is the most widely used and cited text on the Gaussian process for machine learning and most of the literature discussed in the beginning part of this chapter can be found there in detailed form.

3.2 The Regression Problem

Machine learning problems can be roughly categorised into three basic classes.

1. Supervised learning: inferring a function from labelled training data;
2. Unsupervised learning: finding the hidden structure of unlabelled data;
3. Reinforcement learning: taking action by maximising the cumulative reward.

Supervised learning may be further sub-categorised in two fundamental tasks: regression and classification. The regression problem deals with estimating the strength of the relationship among some dependent variables with some independent variables, whereas classification identifies the desired discrete output levels. Bishop (2006); MacKay (2003); Rogers and Girolami (2011) described these concepts in detail.

Regression is the task of making a prediction of a continuous output variable at the desired input, based on a training input-output data set. The input data can be any types of objects or real-valued features located in $\mathbb{R}^D$ which have some predictability for an unobserved location.

By the definition of regression, it is obvious that there will be some inference based on a function mapping the outputs from a set of given inputs because by inferring a function we can predict the response to the desired input. In the case of Bayesian inference, a prior distribution over functions is required. Then the model goes through a training process and updates the prior, based on the training dataset. Let's assume the training data $\mathcal{D}$ constructed with N input vectors, such as $\{\mathbf{X}, \mathbf{y}\}$, where $\mathbf{X} \equiv \{\mathbf{x}_n\}_{n=1}^N$, $\mathbf{x}_n \in \mathbb{R}^D$ are the training inputs and $\mathbf{y} \equiv \{y_n\}_{n=1}^N$, $y_n \in \mathbb{R}$ are the training outputs. Now a key question arises, how can we define a distribution over an infinite dimensional object such as a function?

Although using plain and simple statistics, the regression problem can be solved; the Gaussian process has been proven as a better selection to model a more complex and specific learning task with improved reliability and robustness. Theoretically, Gaussian process regression corresponds to Bayesian linear regression with an infinite number of basis functions. In practice, a finite number of basis functions are used and the number increase with the size of the dataset. Therefore, we can say Gaussian

process models are nonparametric. A Gaussian process model can be used as regression model having an object featuring infinite dimensionality. Gaussian processes have advanced beyond the regression model and now are used for classification (Nickisch and Rasmussen (2008); Williams and Barber (1998)), unsupervised learning (Ek et al. (2008)), reinforcement learning (Deisenroth (2012)) and other related fields in machine learning.

We assume the outputs in the training dataset generated from an underlying mapping function $f(\mathbf{x})$ may contain noise. The objective of the regression problem is to construct $f(\mathbf{x})$ from the data $\mathcal{D}$. This task is ill-defined and dealing with noisy data is even harder as the reasoning of the uncertainty is required. Hence, a single estimate of $f(\mathbf{x})$ clearly could be misleading, rather a probability distribution over likely functions could be much more appealing. A regression model based on the Gaussian process is a fully probabilistic Bayesian model and will definitely serve for our purpose. In contrast with other regression models, here we will get the opportunity to choose the best estimate of $f(\mathbf{x})$. If we consider a probability distribution on functions $p(f)$ as the Bayesian prior for regression, then Bayesian inference can be used to make predictions for given data

$$\overbrace{p\left(f|\mathcal{D}\right)}^{\text{posterior}} = \frac{\overbrace{p\left(\mathcal{D}|f\right)}^{\text{likelihood}}\overbrace{p\left(f\right)}^{\text{prior}}}{\underbrace{p\left(\mathcal{D}\right)}_{\text{marginal likelihood}}} \tag{3.1}$$

where $p\left(f\right)$ is a Gaussian process prior, $p\left(\mathcal{D}|f\right)$ likelihood, $p\left(\mathcal{D}\right)$ is the marginal likelihood and $p\left(f|\mathcal{D}\right)$ is a posterior process over functions. Chapter 2 Section 2.3 showed how marginalization and conditioning is related to Bayesian analysis.

3.3 Gaussian Process definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution (Rasmussen and Williams (2006)). It is a continuous stochastic process and defines probability distributions for functions. It can be also viewed as a collection of random variables indexed by a continuous variable. Any finite set of values from the collection can be written as a vector. Let's consider $\mathbf{f} = \{f_1, f_2, f_3, ..., f_N\}$ corresponds with indexed inputs $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\}$. In Gaussian processes, variables from these random functions are jointly normally distributed and as a whole can be represented as a multivariate Gaussian distribution

$$p(\mathbf{f}|\mathbf{X}) \sim \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \mathbf{K}), \tag{3.2}$$

where $\boldsymbol{\mu}$ is the mean and $\mathbf{K}$ is the covariance matrix. Both potentially depends on $\mathbf{X}$. The Gaussian distribution is over vectors but the Gaussian process is over functions.

We need to define the mean function and covariance function for a Gaussian process prior. If $f(\mathbf{x})$ is a real valued process, a Gaussian process is completely defined by its mean function and covariance function given in Equation 3.3 and Equation 3.4 respectively. Usually the mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ are defined as

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \tag{3.3}$$

and

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \tag{3.4}$$

where $\mathbb{E}$ represents the expected value. We denote the Gaussian process as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{3.5}$$

The covariance matrix $\mathbf{K}$ is constructed from the covariance function $k(\mathbf{x}, \mathbf{x}')$ and $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, that is

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}. \tag{3.6}$$

Loosely speaking, a Gaussian process is multivariate Gaussian distribution defined over an infinite number of dimensions. A sample from a Gaussian process is a random function. While a $n-$dimensional Gaussian distribution is fully specified by mean $\boldsymbol{\mu}$, a $n \times 1$ vector of expectations and covariance matrix $\mathbf{K}$, the $n \times n$ matrix of covariances between all pair of points.

It is a common practice to consider a Gaussian process with zero mean when no prior information is available. This is not excessively restrictive as a variety of functions can be generated by a zero mean process. A second order stationary process has a constant mean and the covariance function solely depends on the distance between the inputs. Zero-mean process is a simplification just by centring the data as $\mathbf{t} = \mathbf{t} - \bar{\mathbf{t}}$, where $\bar{\mathbf{t}}$ is the data sample mean. An extra constant term with the covariance function can reflect

the variation from the mean of the process (MacKay (2003)). So, a constant-mean or a zero-mean assumption is not overly restrictive in practice.

3.4 GP: Covariance Functions

The covariance function (also called kernel, kernel function or covariance kernel) characterises the properties or nature of the samples drawn from a Gaussian process. The covariance function encodes the modelling assumptions we wish to incorporate in our application. The mandatory requirement of a covariance matrix is to be symmetric positive semi-definite[1]. So, as long as the covariance function generates symmetric positive semidefinite matrix, we can use that function for a Gaussian process. Smoothness, periodicity, amplitude, lengthscale etc. are basic properties that can be incorporated while designing Gaussian process covariance function. Once the decision to model with a Gaussian process has been made the choice of the covariance function is a central step in modelling. The main goal of this thesis is to develop covariance functions suitable for transcription factor activity analysis and clustering gene expressions. In this chapter, we will discuss some of the very well known and widely used covariance functions. A wide choice of valid covariance functions and their detail description can be found in Rasmussen and Williams (2006).

Any form of covariance function is acceptable, provided it satisfies the following equation

$$\sum_{i,j} a_i a_j k\left(\mathbf{x}_i, \mathbf{x}_j\right) \geq 0 \tag{3.7}$$

where, $a_i, a_j \ldots a_n$ are arbitrary real coefficients and $\mathbf{x}_i, \mathbf{x}_j \ldots \mathbf{x}_n$ are finite set of data points. A covariance function is termed 'stationary' when it follows

$$\mathrm{Cov}\left[f\left(\mathbf{x}_i\right), f\left(\mathbf{x}_j\right)\right] = k\left(\|\mathbf{x}_i - \mathbf{x}_j\|\right) \tag{3.8}$$

for all $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$. In practice, a stationary covariance function gives a function that is invariant to translation and does not depend on the absolute location of the corresponding inputs; rather it depends on the distance separating them.

If the covariance does not only depend on the distance between the data points in the input space, rather the model needs to adapt to functions where smoothness varies with the inputs, a non-stationary covariance function will be required. There are many interesting non-stationary covariance functions. Depending on nature or trend a careful

[1] A matrix $\mathbf{C}$ is called positive semi-definite if $\mathbf{z}^\top \mathbf{C}\mathbf{z} \geq 0$ for all $\mathbf{z}$. Where $\mathbf{z}$ is a non zero column vector of length n, $\mathbf{C}$ is a $n \times n$ symmetric real matrix and $\mathbf{z}^\top$ is the transpose of $\mathbf{z}$.

selection of appropriate covariance function is essential. One of the simplest examples of non-stationary covariance function which have a linear trend can be expressed by

$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \sum_{d=1}^{D} a_d x_i^d x_j^d \tag{3.9}$$

where x_i^d is the d^{th} component of $\mathbf{x}_i \in \mathbb{R}^D$.

In this thesis, as a prior, we used some stationary covariance functions, and in the following section, we briefly describe some of them. Non-stationary covariance functions are beyond our scope, and a detailed description is available in Rasmussen and Williams (2006).

3.4.1 Exponentiated Quadratic Covariance Function

The exponentiated quadratic covariance is the most widely used covariance function for Gaussian process. This is also known as squared exponential (SE) covariance or radial basis function (RBF). The exponentiated quadratic has become the de-facto default kernel for Gaussian process and has the following form

$$K_{EQ}(r) = a^2 \exp\left(-\frac{r^2}{2l^2}\right), \tag{3.10}$$

where $r = \|\mathbf{x} - \mathbf{x}'\|$. Here $\|\mathbf{x} - \mathbf{x}'\|$ is invariant to translation and rotation. So, the exponentiated quadratic covariance is stationary, as well as isotropic. Here the parameters for output variance a and the lengthscale parameter l govern the property of sample functions and are commonly known as hyperparameters. The parameter a determines the typical amplitude, i.e. average distance of the function away from the mean. l controls the lengthscale, i.e. the length of the wiggles of the function. Figure 3.1(a) represents the kernel and Figure 3.1(b) shows random sample functions drawn from the Gaussian process using exponentiated quadratic covariance with different lengthscales and amplitude hyperparameters. The random functions were generated for a given input range by drawing samples from multivariate Gaussian using Equation 3.2 with zero mean. The smoothness of the sample function depends on the Equation 3.10. Function variables located closer in the input space are highly correlated, whereas function variables located at a distance are loosely correlated or even uncorrelated. Exponentiated quadratic covariance might be too smooth to perform any realistic regression task. Depending on the basic nature of the function, other covariance functions could also be interesting.

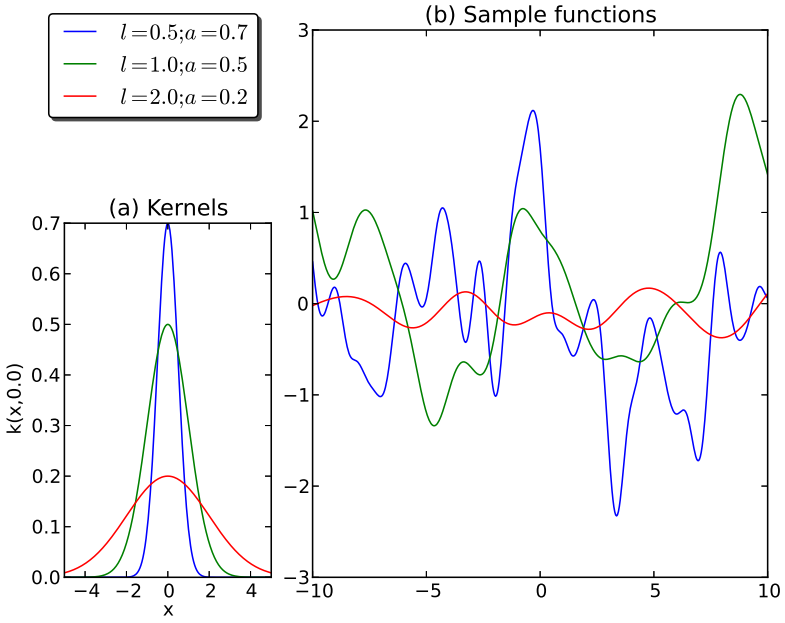


Fig. 3.1 Exponentiated quadratic kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left.

3.4.2 Rational Quadratic Covariance Function

The rational quadratic covariance function is equivalent to the sum of multiple exponentiated quadratic covariance functions with different lengthscales. Gaussian process prior with this kernel function expects a smooth function with many lengthscales. In Equation 3.11 the parameter α can control the relative weights for lengthscale variations. Exponentiated quadratic covariance function can be viewed as a special case of rational quadratic covariance function. If $\alpha \to \infty$, both rational quadratic and exponentiated quadratic functions become identical[2].

$$K_{RQ}(r) = a^2 \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha} \tag{3.11}$$

[2]The limit of a rational quadratic is exponentiated quadratic

$$\lim_{\alpha \to \infty} \left(1 + \frac{x^2}{2\alpha}\right)^{-\alpha} = \exp\left(\frac{x^2}{2}\right).$$

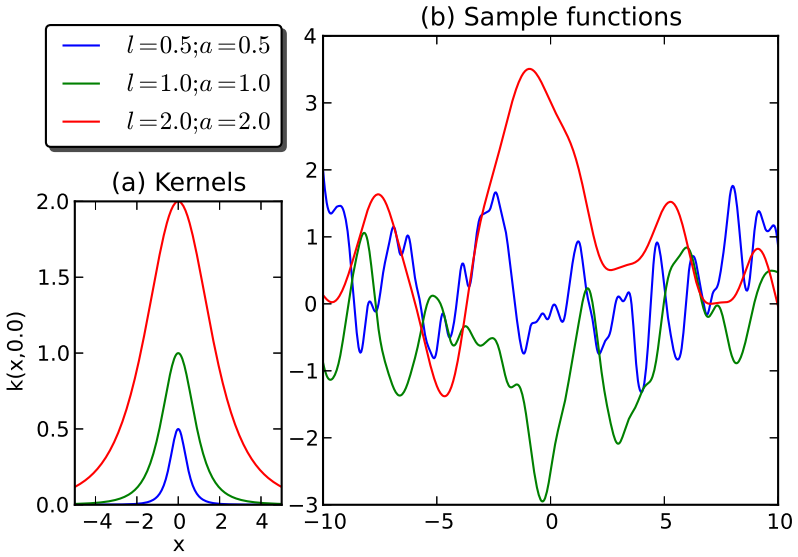


Fig. 3.2 Rational quadratic kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left.

where $r = \|\mathbf{x} - \mathbf{x}'\|$. Figure 3.2 (a) shows the kernels and (b) shows three different random sample functions drawn with different settings of hyperparameters a and l.

3.4.3 The Matérn Covariance Function

The Matérn class of covariance functions are given by Equation 3.12

$$K_{Mat}(r) = a^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu} r}{l} \right) \tag{3.12}$$

where a, l, ν are positive hyperparameters, K_{ν} is a modified Bessel function and $\Gamma(.)$ is the Gamma function. Hyperparameter ν controls the roughness of the function, and like Exponentiated quadratic covariance function the parameters a and l controls the amplitude and lengthscale respectively. Though for $\nu \to \infty$ we can obtain the exponentiated quadratic kernel. For finite value of ν, the sample functions are significantly rough.

The simpler form of Matérn covariance function is obtained when ν is half-integer: $\nu = p + 1/2$, where p is a non-negative integer. The covariance function can be expressed as a product of an exponential and a polynomial of order p. Abramowitz and Stegun

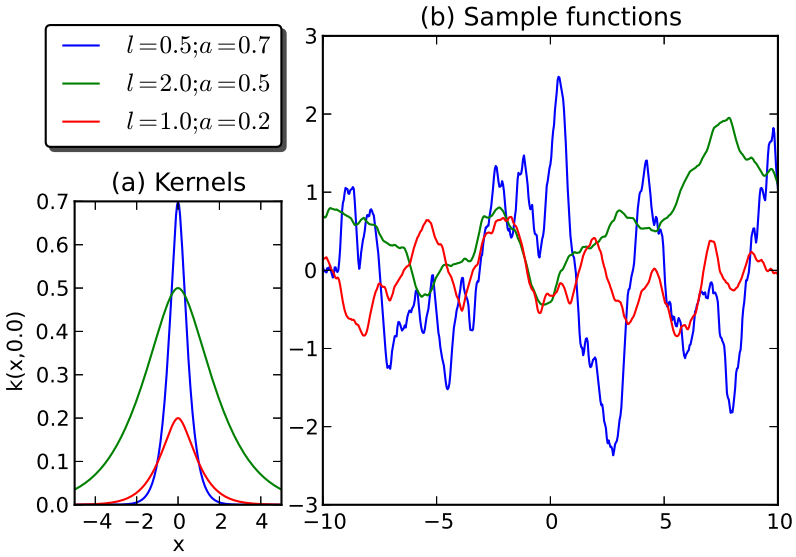


Fig. 3.3 The Matérn32 kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left.

(1965) derived the general expression as follows

$$K_{\nu=p+1/2}(r) = a^2 \exp\left(-\frac{\sqrt{2\nu}r}{l}\right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!\,(p-i)!} \left(\frac{\sqrt{8\nu}r}{l}\right)^{p-i}. \tag{3.13}$$

The most interesting cases for machine learning are $\nu = 3/2$ and $\nu = 5/2$, for which we get the following equations respectively

$$K_{\nu=3/2}(r) = a^2 \left(1 + \frac{\sqrt{3}r}{l}\right) \exp\left(-\frac{\sqrt{3}r}{l}\right) \tag{3.14}$$

and

$$K_{\nu=5/2}(r) = a^2 \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}r}{l}\right). \tag{3.15}$$

3.4.4 The Ornstein-Uhlenbeck Process

The Ornstein-Uhlenbeck process (Uhlenbeck and Ornstein (1930)) is a special case of Matérn class covariance functions. The Ornstein-Uhlenbeck (OU) process was developed as a mathematical model of the velocity of a particle moving with Brownian motion.

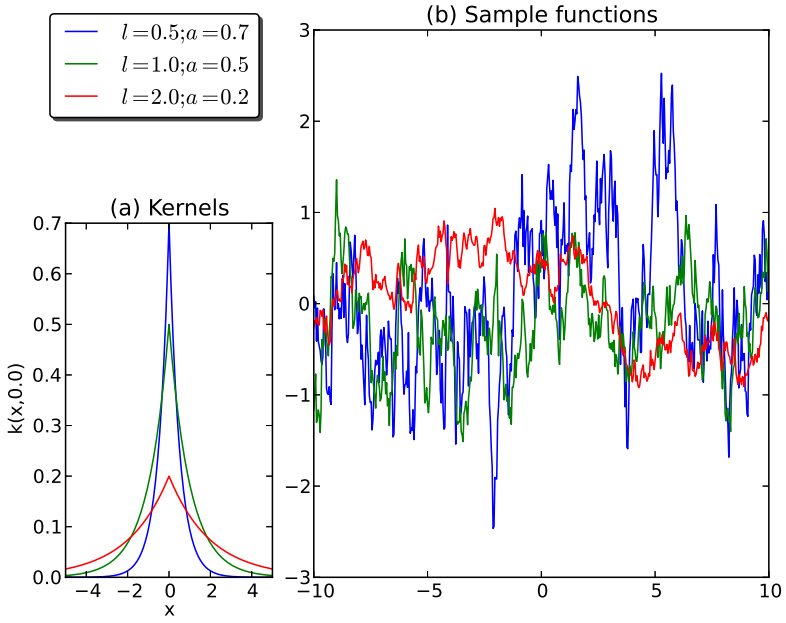


Fig. 3.4 The OU kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left.

The Ornstein-Uhlenbeck process can be found setting up $\nu = 1/2$ and expressed as Equation 3.16. Figure 3.4(a) shows the kernel and Figure 3.4(b) shows the sample functions form the OU process with different hyperparameter settings as shown to the left of the figure.

$$K_{\nu=1/2}(r) = a^2 \exp\left(-\frac{r}{l}\right) \tag{3.16}$$

3.4.5 Cosine Kernel

Perhaps the cosine random processes on $\mathbb{R}$ is one of the most basic and widely used smooth stochastic processes. This periodic stationary process is defined as

$$f(x) \triangleq \xi \cos\lambda x + \xi' \sin\lambda x \tag{3.17}$$

where λ is a positive constant, ξ and ξ' are equidistributed and uncorrelated random variables. Using the basic trigonometry Equation 3.17 can be written as

$$f(x) = R\cos(\lambda(x - \psi)) \tag{3.18}$$

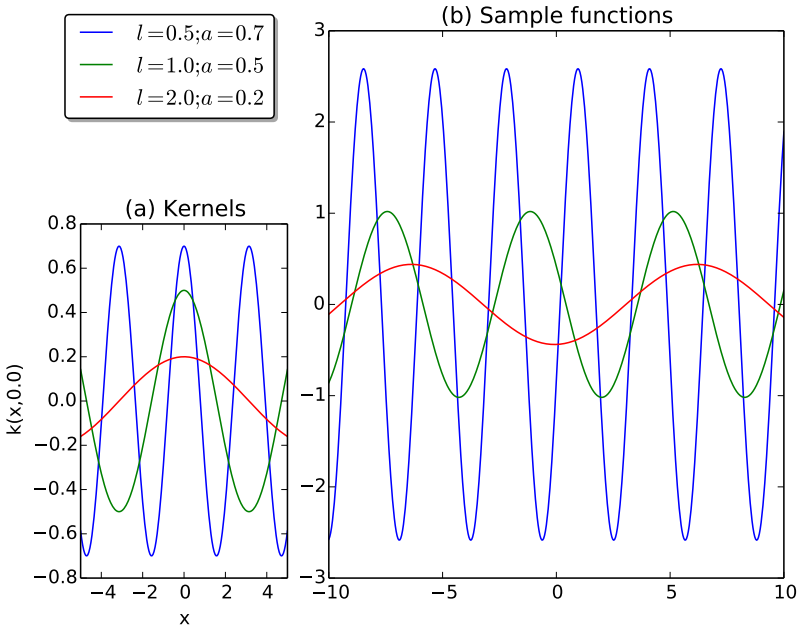


Fig. 3.5 The Cosine kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left.

where $R^2 = \xi^2 + \left(\xi^{'}\right)^2 \geqslant 0$ and $\psi = \arctan\left(\frac{\xi}{\xi^{'}}\right) \in (-\pi, \pi)$. Let's consider $\mathbb{E}[\xi] = 0$. Then, the covariance function is given by

$$\begin{aligned} K_{\text{Cos}}(x, x') &= \mathbb{E}[f(x) f(x')] \\ &= \mathbb{E}[f(\xi \text{cos}\lambda x + \xi' \text{sin}\lambda x)(f(\xi \text{cos}\lambda x' + \xi' \text{sin}\lambda x'))] \\ &= \mathbb{E}\left[(\xi')^2\right] \cos(\lambda(x - x')) \end{aligned}$$

where we considered ξ and ξ' are equidistributed and uncorrelated. The cosine kernel is a stationary kernel regardless of the distribution of ξ. Figure 3.5 shows the representation of kernels and sample functions with different hyperparameter settings.

3.5 Constructing Kernels

Modelling kernel is the central step in Gaussian process modelling. A number of 'built-in' kernels (both stationary and non-stationary) are available for the Gaussian process, yet we may need to model a complicated structure which is not expressed very well by any known kernel. To model such a structure, we may build our own

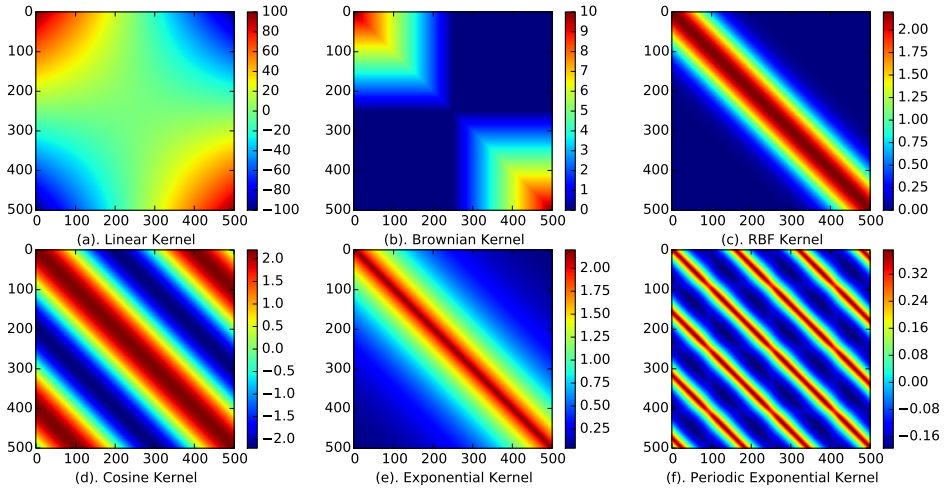


Fig. 3.6 Representation of some basic kernels using the same lengthscale and variance (a) Linear kernel (b) Brownian kernel (c) Exponentiated quadratic kernel (d) Cosine kernel (e) Ornstein-Uhlenbeck kernel and (f) Periodic exponential kernel.

'customised kernel' with the required structure or, desired properties. An addition of two Gaussian variables is a Gaussian. Scaling a Gaussian also leads to a Gaussian. These two basic mathematical properties help to develop a range of kernels from a very simple one to complex a one. Appendix A Section A.1.3, A.1.5 shows the addition and multiplication properties of Gaussian processes respectively.

Figure 3.6 shows the representations of some basic kernels using the same lengthscale and variance (a) Linear kernel (b) Brownian kernel (c) Exponentiated quadratic kernel (or RBF) (d) Cosine kernel (e) Exponential kernel (f) Periodic exponential kernel. These kernels are the realisation of different covariance functions[3]. These kernels (including others) facilitate constructing new kernels or customising 'on demand'of the structure with the desired properties.

Assume an univariate data is globally periodic and local structures governed by some random motions (Brownian motion). There are multiple choices dealing with the global structure and one of the possible solutions could be a Cosine kernel for the global structure and a Brownian kernel for local structures in an additive form. The addition of two positive semi-definite kernels together always results in another positive semi-definite kernel. Figure 3.7 shows the sample functions and representation

[3]We have not described the Linear, Brownian and Periodic exponential kernels here in this thesis. Detail description is available at Rasmussen and Williams (2006) and their *Python*-based implementation is available at The GPy Authors (2014).

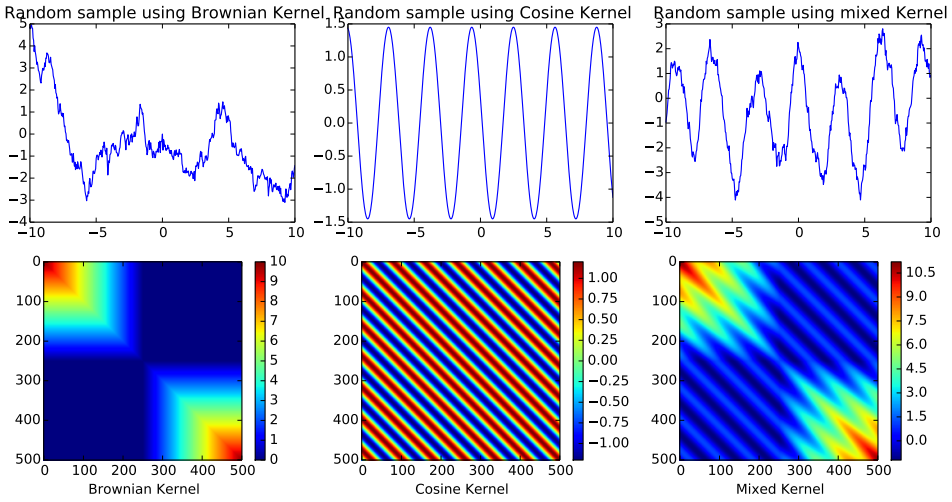


Fig. 3.7 Construction of 'made by order' kernel adding two basic kernels: an example of a univariate data which is globally periodic and locally governed by some random motions. (top-left) a sample is taken using the Brownian kernel, (top-middle) a sample is taken using Cosine kernel, (top-right) a sample taken using a newly constructed kernel by adding two kernels, (bottom-left) Brownian kernel, (bottom-middle) Cosine kernel, (bottom-right) newly constructed kernel.

of the newly constructed kernel. Figure 3.8 shows another example where we used a combination of Cosine kernel and Matérn kernel.

3.6 Gaussian Process Regression

Gaussian process regression can be done using the marginal and conditional properties of the multivariate Gaussian distribution. Let's consider that we have the observation $\mathbf{f}$ of a function at the observation point $\mathbf{x}$. Now we wish to predict the values of that function at the observation points $\mathbf{x}_\star$, which are represented by $\mathbf{f}_\star$. Then the joint probability of $\mathbf{f}$ and $\mathbf{f}_\star$ can be obtained as

$$p\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_\star\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_\star\end{bmatrix}\middle|\, \mathbf{0}, \begin{bmatrix}\mathbf{K}_{\mathbf{x},\mathbf{x}} & \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}\\ \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}} & \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star}\end{bmatrix}\right) \tag{3.19}$$

where the covariance matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ has elements derived from the covariance function $k(x, x')$, such that the $(i,j)^{th}$ element of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ is given by $k(\mathbf{x}[i], \mathbf{x}[i])$. The conditional

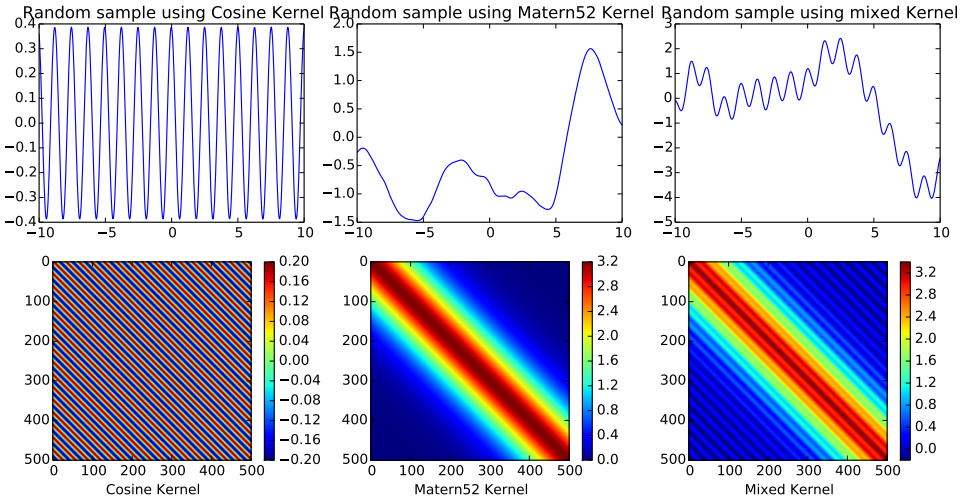


Fig. 3.8 Construction of a new kernel using Cosine kernel and Matérn kernel. (top-left) a sample taken using Cosine kernel, (top-middle) a sample taken using Matérn kernel, (top-right) a sample taken using newly constructed kernel by adding two kernels, (bottom-left) Cosine kernel, (bottom-middle) Matérn kernel, (bottom-right) newly constructed kernel.

property of a multivariate Gaussian is used to perform regression. The conditional property can be represented by

$$p\left(\mathbf{f}_\star|\mathbf{f}\right) = \mathcal{N}\left(\mathbf{f}_\star\middle|\mathbf{K}_{\mathbf{x}_\star,\mathbf{x}}\mathbf{K}_{\mathbf{x},\mathbf{x}}^{-\mathbf{1}}\mathbf{f}, \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star} - \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}}\mathbf{K}_{\mathbf{x},\mathbf{x}}^{-\mathbf{1}}\mathbf{K}_{\mathbf{x},\mathbf{x}_\star}\right). \tag{3.20}$$

In an ideal case, the observation $\mathbf{f}$ is noise-free, but, in practice, it is always corrupted with some noise. Let's consider $\mathbf{y}$ as a corrupted version of $\mathbf{f}$. If we consider this noise as Gaussian noise, we can write $p\left(\mathbf{y}|\mathbf{f}\right) = \mathcal{N}\left(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}\right)$, where σ^2 is the variance of the noise and $\mathbf{I}$ is the identity matrix with the appropriate size and marginalise the observation $\mathbf{f}$. Then, the joint probability of $\mathbf{y}$ and $\mathbf{f}_\star$ can be represented by

$$p\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_\star\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_\star\end{bmatrix}\middle|\mathbf{0}, \begin{bmatrix}\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I} & \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}\\ \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}} & \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star}\end{bmatrix}\right). \tag{3.21}$$

Regression with Gaussian process is a Bayesian method. From the knowledge of a *prior* over a function and data, we infer a *posterior* and this happens in a closed form of Equation 3.20.

Figure 3.9 shows the overall covariance structure between some training and test data. For this example, we choose 18 training points and 82 test points. We observe

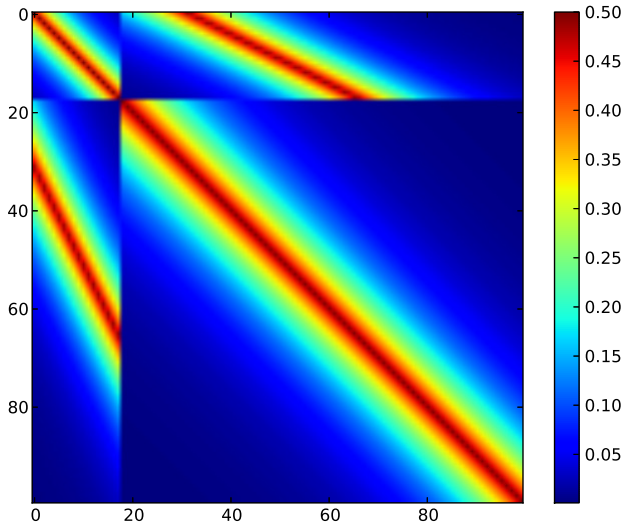


Fig. 3.9 Overall representation of covariances between training and test data.

the shaded structure because some of the training data are closer to some of the test data. Observing this structure, we can also figure out the closeness between training and test data.

3.6.1 Making Predictions

The probability density is represented by functions. Due to consistency this density is known as a process. Also by this property, any future values of $\mathbf{f}_\star$ which are unobserved can be predicted without affecting $\mathbf{f}$. To make predictions of the test data, we use the conditional distribution. In an ideal case, the conditional distribution is $p(\mathbf{f}_\star|\mathbf{f})$ and if we consider the noise, the conditional distribution will be $p(\mathbf{f}_\star|\mathbf{y})$. Both of the distributions are also Gaussian

$$\mathbf{f}_\star \sim \mathcal{N}\left(\boldsymbol{\mu}_f, \mathbf{C}_f\right). \tag{3.22}$$

The mean of the conditional distribution in Equation 3.22 is

$$\boldsymbol{\mu}_f = \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}^\top \left[\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{y} \tag{3.23}$$

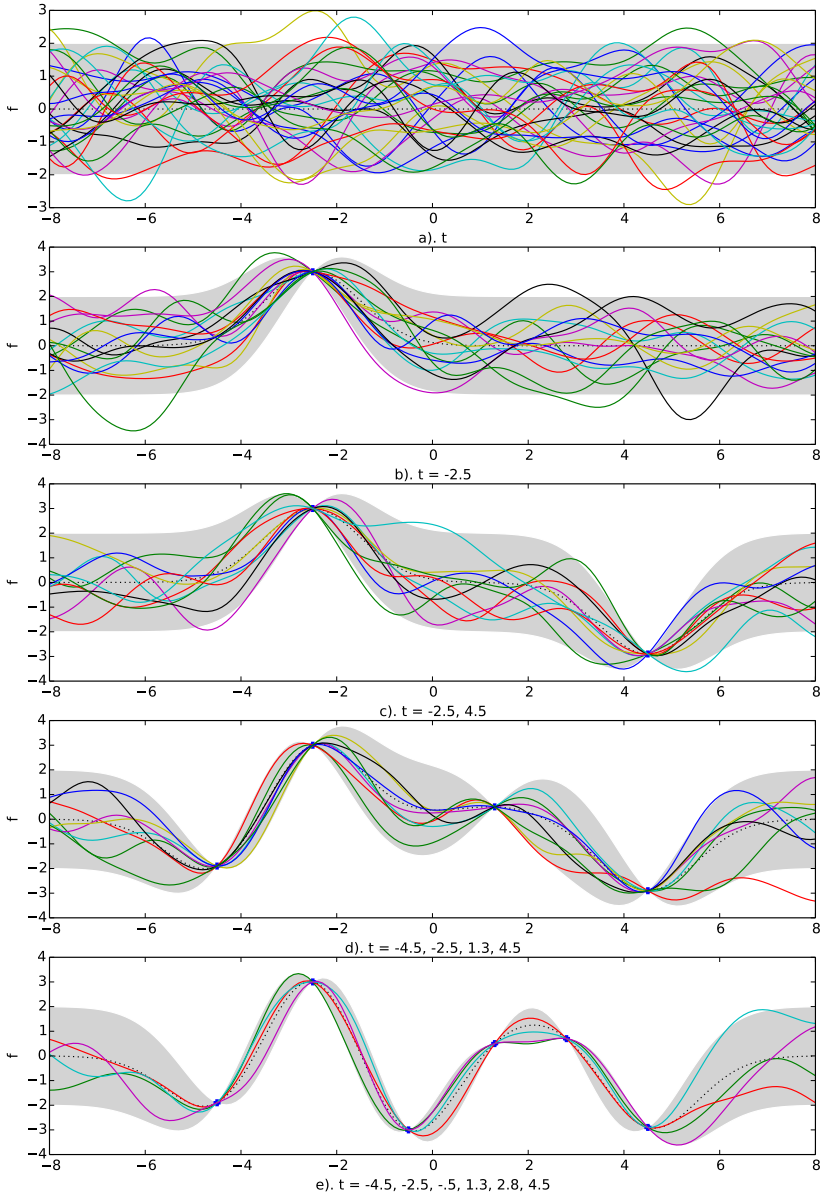


Fig. 3.10 A representation of Gaussian process regression: Modelling one-dimensional function using Gaussian process. Coloured solid lines represent different samples from the process and the dotted line is the mean function. The shaded area is the 95% confidence interval. (a) A Gaussian process not conditioned on any data points. Without any observations, the prior uncertainty about the underlying function is constant everywhere. (b-e) The posterior after conditioning on different amount of data.

and its covariance is given by

$$\mathbf{C}_f = \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star} - \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}^\top \left[\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right]^{-1} \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}. \tag{3.24}$$

These results can be calculated using the block matrix inverse rules. The derivation can be found in Appendix A Section A.2. Figure 3.10 shows a visual representation of Gaussian process regression for a one-dimensional function. Coloured solid lines represent different samples from the process and the dotted line is the mean function. The shaded area is the 95% confidence interval. Figure 3.10(a) represents a Gaussian process not conditioned on any data points. Without any observations, the prior uncertainty about the underlying function is constant everywhere. Figure 3.10(b-e) show some posterior samples after conditioning on different amount of training data as shown in the figure.

3.6.2 Hyperparameter Learning

To construct the covariance function, still, we need to consider the hyperparameters and optimize them. These adjustable parameters alter the distribution of the function output values obtained from a Gaussian process. The most efficient and commonly used optimization technique for hyperparameters is the maximum likelihood. If we consider all the hyperparameters α (controls the amplitude), σ^2 (variance of the noise) and l (length-scale) in a vector $\boldsymbol{\theta}$, we can use gradient methods to optimize $p\left(\mathbf{y}|\boldsymbol{\theta}\right)$ with respect to $\boldsymbol{\theta}$. The Log likelihood is given by

$$\log p\left(\mathbf{y}|\boldsymbol{\theta}\right) = -\frac{D}{2}\log 2\pi - \frac{1}{2} \times \log\left|\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right| - \frac{1}{2}\mathbf{y}^\top \left[\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{y}. \tag{3.25}$$

We can have the log maximum likelihood by

$$\boldsymbol{\theta}_{max} = \arg\max_{\theta}\left(p\left(\mathbf{y}|\boldsymbol{\theta}\right)\right). \tag{3.26}$$

3.7 Mean reverting Ornstein-Uhlenbeck Process

In 1827 the botanist Robert Brown noticed that discharged pollen particle from the male part of the flower moves in a random fashion in water. Though he narrated the problem but did not explain underlying mechanism that may create this motion Brown (1828). Later in 1905, Einstein argued that the motion of the particle created

by the momentum and energy exchange due to collisions of the particles surrounding it. Einstein also gave a mathematical explanation of the transition probability Einstein (1905). In 1920 Norbert Wiener presented the sound mathematical foundation for Brownian motion as a stochastic process Wiener (1923).

The Ornstein-Uhlenbeck process is a diffusion process, which was introduced to describe the model of velocity for a moving particle undergoing Brownian motion. When the position of a particle is described using Brownian motion, the time derivative does not exist. The Ornstein-Uhlenbeck process is an attempt to overcome this difficulty by modelling the velocity directly. The Ornstein-Uhlenbeck process is a stationary Gauss-Markov process (i.e. the process follow both Gaussian process and Markovian process). Over time, the Ornstein-Uhlenbeck process tends to drift towards its long-term mean value. For this reason, this process is called as mean reverting Ornstein-Uhlenbeck Process Uhlenbeck and Ornstein (1930).

The Ornstein-Uhlenbeck process is widely used for modelling a mean reverting process. An Ornstein–Uhlenbeck process, u, satisfies the following stochastic differential equation

$$du = -\lambda\left(u - \mu\right)dt + \sigma dB, \tag{3.27}$$

where

- λ controls the speed of mean reversion
- μ is the long term mean , to which the process tends to revert.
- dB is a Brownian motion, where $dB \sim \mathcal{N}(0, dt^{1/2})$
- σ is a measures of the process volatility.

In order to model the Ornstein-Uhlenbeck process, it is usual to discretize time and calculate samples at discrete time steps of width Δt. A naïve approach to derive the Equation 3.36 is

$$u_t - u_{t-1} = -\lambda\left(u_{t-1} - \mu\right)\Delta t + \sigma dB \tag{3.28}$$

$$u_t = u_{t-1} - \lambda\left(u_{t-1} - \mu\right)\Delta t + \sigma dB \tag{3.29}$$

For any any *in silico* experiment Gillespie (1996) pointed that Equation 3.29 is only valid when Δt is sufficiently small. The exact formula for any size of Δt is given by

$$u_t = e^{-\lambda\Delta t}u_{t-1} + \left(1 - e^{-\lambda\Delta t}\right)\mu + \sigma\sqrt{\frac{\left(1 - e^{-2\lambda\Delta t}\right)}{2\lambda}}dB \tag{3.30}$$

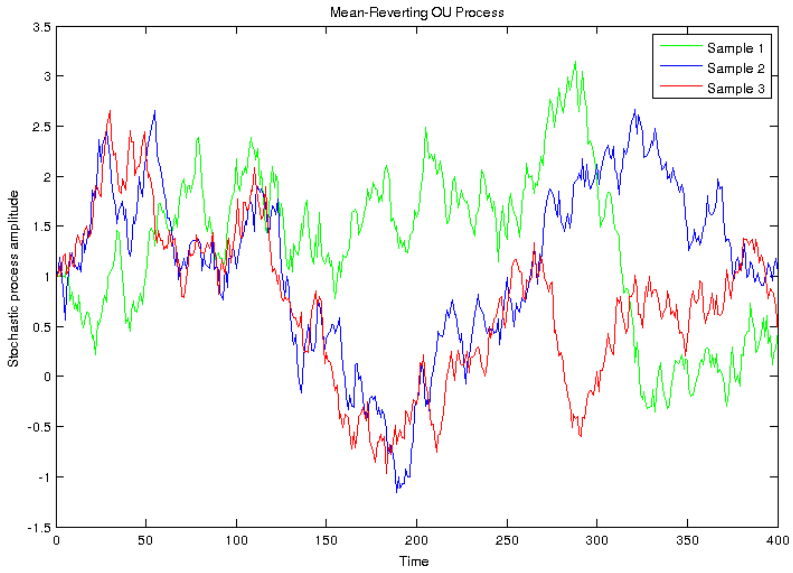


Fig. 3.11 Sample paths of various mean reverting Ornstein Uhlenbeck processes with parameters $\mu = 7, \sigma = 1.5, u_0 = 1,$ and $\Delta t = 0.01$. Where x-axis represents the time and y-axis represents the amplitude of the stochastic process.

Figure 3.11 shows few examples of sample paths of mean reverting Ornstein-Uhlenbeck processes with different parameters settings.

3.8 Toward the GP model of TFA

We are interested in developing a non-parametric model of transcription factor activity using Gaussian process. Here, we want to prove that there is an analogical pathway[4] to construct a kernel function for Gaussian process model from Markovian assumption (Appendix A, Section A.4) based probabilistic approach of Sanguinetti et al. (2006). From Chapter 2 Section 2.6 we have the probabilistic gene specific TFAs as

$$\mathbf{b}_{n(t+1)} \sim \mathcal{N}(\gamma \mathbf{b}_{nt} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}). \tag{3.31}$$

For a discrete time variable k the above equation can be rewritten as

$$\mathbf{b}_{n(k+1)} \sim \mathcal{N}\left(\gamma \mathbf{b}_{nk} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}\right), \tag{3.32}$$

[4]We would like to acknowledge Simo Särkkä, Academy Research Fellow, Aalto University, Finland for his valuable suggestions and guidelines.

and

$$\mathbf{b}_{n_1} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{3.33}$$

Let's now form a continuous model which has the same finite-dimensional distribution. First we construct a one-dimensional process with the property

$$u_{k+1} \sim \mathcal{N}\left(\gamma u_k + (1-\gamma)\,\mu, (1-\gamma^2)s\right), \tag{3.34}$$

where μ and s are scalar.

We can now assume that u_k's are actually values u_{t_k} from a continuous process $u(t)$ and let's assume that

$$t_k = kDt. \tag{3.35}$$

A good candidate for this kind of model is the mean-reverting $Ornstein-Uhlenbeck$ model (Uhlenbeck and Ornstein (1930))

$$du = -\lambda\,(u-\mu)\,dt + q^{1/2}dB, \tag{3.36}$$

where B is a standard Brownian motion (i.e., Wiener process). This equation can now be solved on the time instants t_k and the result is a recursion

$$u(t_k) = au(t_{k-1}) + b\mu + w_{k-1}, \tag{3.37}$$

where $w_{k-1} \sim \mathcal{N}(0, c)$ with

$$a = \exp(-\lambda Dt)$$

$$\begin{aligned} b &= \int_0^D t\exp(-\lambda(Dt-s))ds \\ &= 1-\exp(-\lambda Dt) \end{aligned}$$

$$\begin{aligned} c &= \int_0^D t\exp(-\lambda(Dt-s))q\exp(-\lambda(Dt-s))ds \\ &= q\int_0^D t\exp(-2\lambda(Dt-s))ds \\ &= [q/(2\lambda)][1-\exp(-2\lambda Dt)]. \end{aligned}$$

That is,

$$u_{k+1} \sim \mathcal{N}\left(au_k + b\mu, c\right). \tag{3.38}$$

We can now match the coefficients

$$a = \exp(-\lambda Dt) = \gamma \tag{3.39}$$

$$b = 1 - \exp(-\lambda Dt) = 1 - \gamma \tag{3.40}$$

$$c = (1 - \gamma^2)s = [q/(2\lambda)][1 - \exp(-2\lambda Dt)] \tag{3.41}$$

Equation 3.39 has a nice solution $\gamma = \exp(-\lambda Dt)$ and from Equation 3.41 we have another solution $s = q/(2\lambda)$, which can be inverted to give $\lambda = -[1/Dt]\log\gamma$ and $q = -[2s/Dt]\log\gamma$.

If we fix $Dt = 1$, we get $\lambda = -\log\gamma$ and $q = -2s\log\gamma$.

We can now recall the (stationary) covariance function of the Ornstein-Uhlenbeck process

$$\begin{aligned} k_u(t,t') &= [q/(2\lambda)]\exp(-\lambda|t-t'|) \\ &= s\exp((\log\gamma)|t-t'|) \\ &= s\exp(|t-t'|(\log\gamma)) \\ &= s\exp(\log\gamma^{|t-t'|}) \\ &= s\gamma^{|t-t'|}. \end{aligned}$$

When we start from variance $s = q/[2\lambda]$, the process indeed is stationary from the beginning. Returning to the original vector valued $\mathbf{b}$, because the system is separable, we can conclude that the implied covariance function is just obtained by formally replacing s with $\mathbf{\Sigma}$ everywhere

$$\mathbf{K}_b(t,t') = \mathbf{\Sigma}\boldsymbol{\gamma}^{|t-t'|} \tag{3.42}$$

This is equivalent to considering the vector process of mean-reverting $Ornstein - Uhlenbeck$ model

$$\mathbf{db} = -\lambda(\mathbf{b} - \boldsymbol{\mu})\mathbf{dt} + Q^{1/2}\mathbf{dB}. \tag{3.43}$$

3.9 Gaussian Process: Pros and Cons

The most appealing feature of the Gaussian process is expressibility. It is possible to express a very wide range of modelling assumptions through a proper choice of covariance function. Given a covariance function and some observations, the posterior distribution can be predicted analytically. Modelling with the Gaussian process is nonparametric and this is a rare property. The marginal likelihood of the data given a model is calculated by integrating over all hypothesis. The Gaussian process compares different models and improves the model selection. Integration over a wide range of hypothesis lessens over-fitting than in comparable model class. The predictive distribution of Gaussian process is a multivariate Gaussian distribution and can be easily combined with other models.

There are several issues which may make Gaussian processes sometimes difficult to use. In the generic inference and learning algorithm, we need to inverse the covariance matrix which has $\mathcal{O}(N^3)$ runtime complexity. Given the computational resource available at present, the exact inference is prohibitively slow for more than a few thousands of data-points. An exact inference for typical 'Big data' could be very costly from the perspective of runtime complexity. However, this problem can be addressed by variational inference, even for models containing millions of data points (Hensman et al. (2013a)). Non-Gaussian predictive likelihoods could be challenging while working with the Gaussian process. However, the Gaussian process framework GPy (The GPy Authors (2014)) can automatically deal with the last two issues.

3.10 Discussion

In this chapter, we briefly described the Gaussian process, regression problem and regression with the Gaussian process. The choice of the covariance function is a central step in modelling with a Gaussian process. Our main goal of this thesis is to develop covariance functions suitable for transcription factor activity analysis and gene expressions clustering. In this chapter, we briefly described some commonly used kernels. We also mentioned hyperparameter learning. Finally, we justified the rationale behind choosing the Ornstein-Uhlenbeck kernel to model the transcription factor activity. In the next chapter (Chapter 4), we will develop a Gaussian process model to infer the transcription factor activity.

Chapter 4

GP Model of TFAs

In this chapter, we design a covariance function or kernel to reconstruct transcription factor activities given gene expression profiles, and a connectivity matrix (also termed as binding data) between genes and transcription factors our Gaussian process. Our modelling framework builds on ideas of Sanguinetti et al. (2006) who used a linear-Gaussian state-space modelling framework to infer the transcription factor activity of a group of genes. In Chapter 3 Section 3.8, we showed how the linear-Gaussian state-space modelling framework of Sanguinetti et al. (2006) can be updated using Gaussian process with Ornstein-Uhlenbeck kernel to model the transcription factor activity.

We note that the transcription factor activity model with Markov property proposed by Sanguinetti et al. (2006) is a linear Gaussian state space model which is equivalent to a Gaussian process model with a particular covariance function. We, therefore, build a model straight from the Gaussian process perspective to achieve the same effect. Here we introduced a computational mechanism, based on a judicious application of singular value decomposition (SVD), to enable us to efficiently fit the Gaussian process in a reduced 'transcription factor activity' space.

In the probabilistic inference of transcription factor activities, Spellman et al. (1998) used gene expression time series data of synchronised yeast cells from the CDC-15 experiment. Two colour spotted cDNA array data set of a series of experiments to identify which genes in Yeast are cell cycle regulated.

Our second data set is from ChiP-chip experiments performed on yeast by Lee et al. (2002). These give us the connectivity or binding information between transcription factors and genes. As we know, a transcription factor can regulate multiple genes, on another way a gene can be regulated by multiple transcription factors. So, there exists a many-to-many relationship (Multiple input motifs (e) of Figure 2.6) between

genes and transformation factors. In this thesis, we are going to combine this binding information with the gene expression information to infer transcription factor activities.

4.1 Model for Transcription Factor Activities

Given $\mathbf{Y} \in \mathbb{R}^{n \times T}$, is the matrix of *log*-ed gene expression, where n is the number of genes, T is the time points[1]. We assume a linear (additive) is model giving the relationship between the expression level of the gene and the corresponding transcription factor activity which are unobserved (what we termed as our latent variable). We represent transcription factor activity by a matrix $\mathbf{F} \in \mathbb{R}^{q \times T}$, where q is the number of transcription factors and T is the time points as mentioned earlier. Our basic assumption is as follows

1. Transcription factors' activities are in time series, so they are likely to be temporally smooth.

2. The transcription factors are potentially correlated with one another (to account for transcription factors that operate simultaneously).

Correlation Between Transcription Factors: Let's consider there are q transcription factors. The correlation between different transcription factors is encoded in a covariance matrix, $\mathbf{\Sigma}$ which have $q \times q$ dimensions.

Temporal Smoothness: we assume that the *log* of the transcription factors' activities is temporally smooth, and drawn from an underlying Gaussian process with covariance $\mathbf{K}_t$.

Intrinsic Coregionalization Model: we assume that the joint process across all q transcription factor activities and across all time points is well represented by an intrinsic model of coregionalization where the covariance function is given by the Kronecker product of these terms.

$$\mathbf{K}_f = \mathbf{K}_t \otimes \mathbf{\Sigma} \tag{4.1}$$

This is known as an intrinsic coregionalization model (Wackernagel (2003)). Alvarez et al. (2012) presented the machine learning orientated review of these methods. The matrix $\mathbf{\Sigma}$ is known as the coregionalization matrix. We describe the methodology of designing kernel using *coregionalization* at Chapter 5 in Section 5.2.2. Figure 4.1 is the realization of intrinsic model of coregionalization.

[1]Data samples taken since the beginning of the experiment.

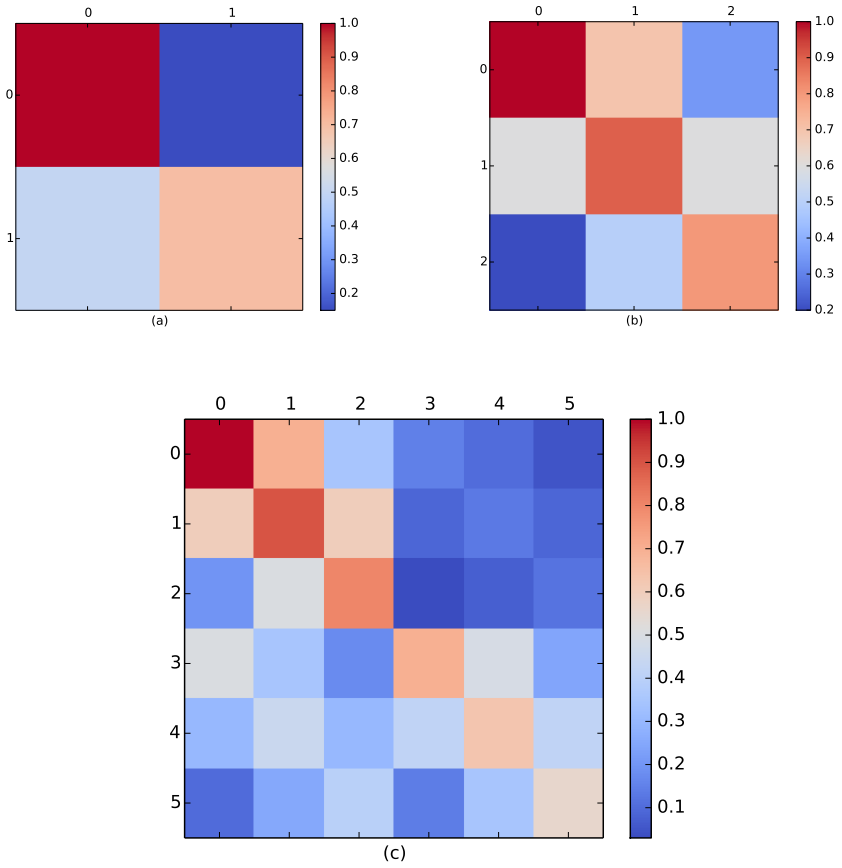


Fig. 4.1 Demonstration of Kronecker product by tiling. Assume (a) represents $\mathbf{K}_t$ and (b) represents $\mathbf{\Sigma}$ of Equation 4.1, then the representation of $\mathbf{K}_f$ will be as like (c).

4.2 Relation to Gene Expressions

Let's consider the j^{th} gene's expression is given by the product of the transcription factors that bind to that gene. Because we are working in *log*-space, that implies a $log - linear$ relationship. At the i^{th} time point, the *log* of the j^{th} gene's expression, $\mathbf{y}_{i,j}$ is linearly related to the log of the transcription factor activities at the corresponding time point, $\mathbf{f}_{i,:}$. This relationship is given by the binding information from $\mathbf{S}$. We then assume that there is some corrupting Gaussian noise to give us the final observation.

$$\mathbf{y}_{i,j} = \mathbf{S}\mathbf{f}_{:,i} + \boldsymbol{\epsilon}_i \tag{4.2}$$

where the Gaussian noise is sampled from

$$\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \tag{4.3}$$

4.3 Gaussian Process Model of Gene Expression

We consider a vector operator which takes all the separate time series in $\mathbf{Y}$ and stacks the time series to form a new vector $n \times T$ length vector $\mathbf{y}$. A similar operation is applied to form a $q \times T$ length vector $\mathbf{f}$. Using Kronecker products we can now represent the relationship between $\mathbf{y}$ and $\mathbf{f}$ as follows: Standard properties of multivariate Gaussian distributions (Appendix A, Section A.1) tell us that

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}), \tag{4.4}$$

where

$$\mathbf{K} = \mathbf{K}_t \otimes \mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top + \sigma^2\mathbf{I}. \tag{4.5}$$

This results in a covariance function that is of size n by T where n is the number of genes and T is the number of time points. However, we can get a drastic reduction in the size of the covariance function by considering the singular value decomposition (Appendix A, Section A.3) of $\mathbf{S}$. The matrix $\mathbf{S}$ is n by q matrix, where q is the number of transcription factors. It contains a '1' if a given transcription factor binds to a given gene, and '0' otherwise. The likelihood of a multivariate Gaussian is

$$L = -\frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}\mathbf{y}^\top\mathbf{K}^{-1}\mathbf{y} \tag{4.6}$$

In the worst case, because the vector $\mathbf{y}$ contains $T \times n$ points (T time points for each of n genes) we are faced with $\mathcal{O}(T^3n^3)$ computational complexity. We are going to use a rotation trick to get the likelihood.

4.4 Method of Computation

4.4.1 Rotating the Basis of a Multivariate Gaussian

For any multivariate Gaussian you can rotate the data set and compute a new rotated covariance which is valid for the rotated data set. Mathematically this works by first

inserting $\mathbf{R}\mathbf{R}^\top$ into the likelihood at three points as follows:

$$L = -\frac{1}{2}\log|\mathbf{K}\mathbf{R}^\top\mathbf{R}| - \frac{1}{2}\mathbf{y}^\top\mathbf{R}^\top\mathbf{R}\mathbf{K}^{-1}\mathbf{R}^\top\mathbf{R}\mathbf{y} + \text{const} \tag{4.7}$$

The rules of determinants and a transformation of the data allows us to rewrite the likelihood as

$$L = -\frac{1}{2}\log|\mathbf{R}^\top\mathbf{K}\mathbf{R}| - \frac{1}{2}\hat{\mathbf{y}}^\top\left[\mathbf{R}^\top\mathbf{K}\mathbf{R}\right]^{-1}\hat{\mathbf{y}} + \text{const} \tag{4.8}$$

where we have introduced the rotated data: $\hat{\mathbf{y}} = \mathbf{R}\mathbf{y}$. Geometrically what this says is that if we want to maintain the same likelihood, then when we rotate our data set by $\mathbf{R}$ we need to rotate either side of the covariance matrix by $\mathbf{R}$, which makes perfect sense when we recall the properties of the multivariate Gaussian.

4.4.2 A Kronecker Rotation

In this thesis, we are using a particular structure of covariance which involves a Kronecker product. The rotation we consider will be a Kronecker rotation (Stegle et al. (2011)). We are going to try and take advantage of the fact that the matrix $\mathbf{S}$ is square meaning that $\mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top$ is not full rank (it has rank of most q, but is size $n \times n$, and we expect number of transcription factors q to be less than number of genes n).

When ranks are involved, it is always a good idea to look at singular value decompositions (SVDs). The SVD of $\mathbf{S}$ is given by:

$$\mathbf{S} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{V}^\top \tag{4.9}$$

where $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$, $\boldsymbol{\Lambda}$ is a diagonal matrix of positive values, $\mathbf{Q}$ is a matrix of size $n \times q$: it matches the dimensionality of $\mathbf{S}$, and we have $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$. Note that because it is not square, $\mathbf{Q}$ is not in itself a rotation matrix. However it could be seen as the first q columns of an n dimensional rotation matrix (assuming n is larger than q, i.e. there are more genes than transcription factors).

If we call the $n - q$ missing columns of this rotation matrix $\mathbf{U}$ then we have a valid rotation matrix $\mathbf{R} = \begin{bmatrix}\mathbf{Q} & \mathbf{U}\end{bmatrix}$. Although this rotation matrix is only rotating across the n dimensions of the genes, not the additional dimensions across time. In other words we are choosing $\mathbf{K}_t$ to be unrotated. To represent this properly for our covariance we need to set $\mathbf{R} = \mathbf{I} \otimes \begin{bmatrix}\mathbf{Q} & \mathbf{U}\end{bmatrix}$. This gives us a structure that when applied to a covariance of the form $\mathbf{K}_t \otimes \mathbf{K}_n$ it will rotate $\mathbf{K}_n$ whilst leaving $\mathbf{K}_t$ untouched.

When we apply this rotation matrix to $\mathbf{K}$ we have to consider two terms, the rotation of $\mathbf{K}_t \otimes \mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top$, and the rotation of $\sigma^2\mathbf{I}$.

Rotating the latter is easy, because it is just the identity matrix multiplied by a scalar so it remains unchanged

$$\mathbf{R}^\top \mathbf{I}\sigma^2 \mathbf{R} = \mathbf{I}\sigma^2 \tag{4.10}$$

The former is slightly more involved, for that term we have

$$\left[\mathbf{I} \otimes \begin{bmatrix}\mathbf{Q} & \mathbf{U}\end{bmatrix}^\top\right] \mathbf{K}_t \otimes \mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top \left[\mathbf{I} \otimes \begin{bmatrix}\mathbf{Q} & \mathbf{U}\end{bmatrix}\right] = \mathbf{K}_t \otimes \begin{bmatrix}\mathbf{Q} & \mathbf{U}\end{bmatrix}^\top \mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top \begin{bmatrix}\mathbf{Q} & \mathbf{U}\end{bmatrix}. \tag{4.11}$$

Since $\mathbf{S} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{V}^\top$ then we have

$$\begin{bmatrix}\mathbf{Q} & \mathbf{U}\end{bmatrix}^\top \mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top \begin{bmatrix}\mathbf{Q} & \mathbf{U}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Lambda}\mathbf{V}^\top\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}. \tag{4.12}$$

This prompts us to split our vector $\hat{\mathbf{y}}$ into a q dimensional vector $\hat{\mathbf{y}}_u = \mathbf{U}^\top\mathbf{y}$ and an $n-q$ dimensional vector $\hat{\mathbf{y}}_q = \mathbf{Q}^\top\mathbf{y}$. The Gaussian likelihood can be written as

$$L = L_u + L_q + \text{const} \tag{4.13}$$

where

$$L_q = -\frac{1}{2}\log|\mathbf{K}_t \otimes \boldsymbol{\Lambda}\mathbf{V}^\top\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda} + \sigma^2\mathbf{I}| - \frac{1}{2}\hat{\mathbf{y}}_q^\top \left[\mathbf{K}_t \otimes \boldsymbol{\Lambda}\mathbf{V}^\top\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda} + \sigma^2\mathbf{I}\right]^{-1} \hat{\mathbf{y}}_q \tag{4.14}$$

and

$$L_u = -\frac{T(n-q)}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\hat{\mathbf{y}}_u^\top\hat{\mathbf{y}}_u \tag{4.15}$$

Strictly speaking, we should fit these models jointly, but for the purposes of illustration, we will firstly use a simple procedure. Firstly, we fit the noise variance σ^2 on $\hat{\mathbf{y}}_u$ alone using L_u. Once this is done, fix the value of σ^2 in L_q and optimise with respect to the other parameters.

With the current design, the model is switching off the temporal correlation. The next step in the analysis will be to reimplement the same model as described by Sanguinetti et al. (2006) and recover their results. That will involve using an Ornstein-Uhlenbeck covariance (we proved the rationale behind the choice of the covariance function in Chapter 3 Sec 3.8) and joint maximisation of the likelihood of L_u and L_q.

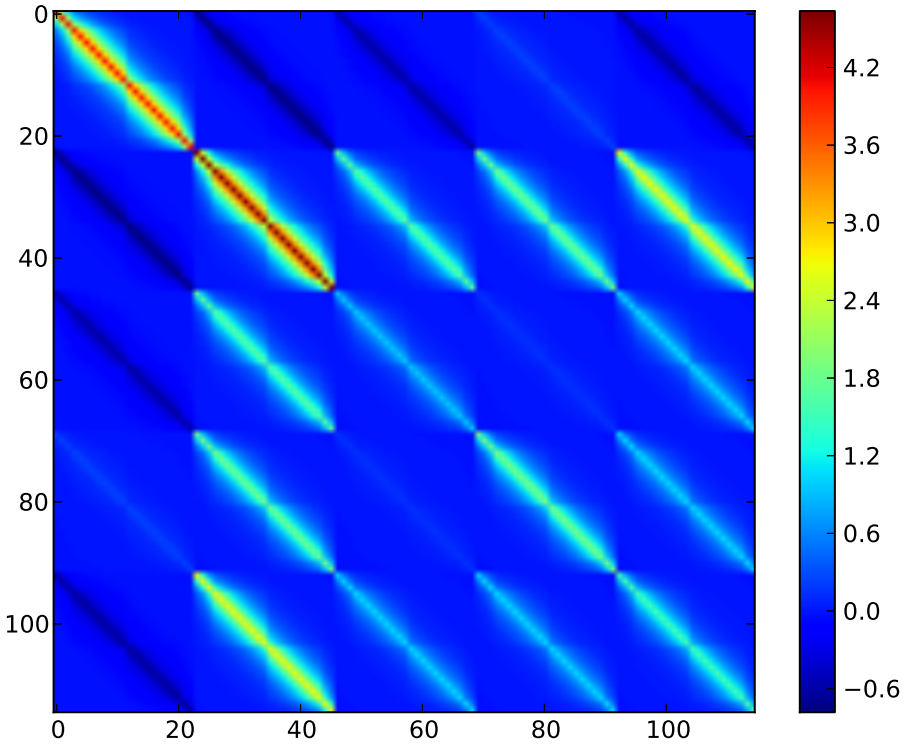


Fig. 4.2 Kernel of Intrinsic Coregionalization model $\mathbf{K}_f$ considering 5 transcription factors where covariance matrix $\mathbf{\Sigma}$ of Equation 4.5 was constructed using Ornstein-Uhlenbeck kernel and White kernel in additive form.

4.5 Making Prediction

Using Kronecker product we can rewrite the Equation 4.4 as

$$\mathbf{y_q} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}_{t,t} \otimes \mathbf{\Lambda V}^T \mathbf{\Sigma V \Lambda} + \sigma^2 \mathbf{I}\right) \tag{4.16}$$

Standard properties of multivariate Gaussian distributions tells us can split equation 4.16 into

$$\mathbf{y_q} = \mathbf{g} + \boldsymbol{\epsilon} \tag{4.17}$$

where $\mathbf{g}$ and $\boldsymbol{\epsilon}$ are also Gaussian distributions and can be represented by

$$\mathbf{g} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}_{t,t} \otimes \mathbf{\Lambda V}^T \mathbf{\Sigma V \Lambda}\right) \tag{4.18}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right) \tag{4.19}$$

Now we can represent the matrix $\mathbf{F}$ of transcription factor activity as

$$\mathbf{F} = \mathbf{I} \otimes \mathbf{V}\Lambda^{-1}\mathbf{g} \tag{4.20}$$

$$\boldsymbol{\Sigma} = \mathbf{W}\mathbf{W}^T + \operatorname{diag}\left(\boldsymbol{\kappa}\right) \tag{4.21}$$

where $\boldsymbol{\kappa}$ is the kappa value from coregionalization matrix.

$$\mathbf{F} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}_{t,t} \otimes \boldsymbol{\Sigma}\right) \tag{4.22}$$

Now we can find the conditional distribution of $\mathbf{g}$ for given $\mathbf{y}_q$ by

$$p\left(\mathbf{g}|\mathbf{y}_q\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_g, \mathbf{C}_g\right) \tag{4.23}$$

with a mean given by

$$\boldsymbol{\mu}_g = \left[\mathbf{K}_{t_\star,t} \otimes \boldsymbol{\Lambda}\mathbf{V}^T\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda}\right]\left[\mathbf{K}_{t,t} \otimes \boldsymbol{\Lambda}\mathbf{V}^T\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{y}_q \tag{4.24}$$

and the covariance given by

$$\begin{aligned}\boldsymbol{C}_g = &\left[\mathbf{K}_{t_\star,t_\star} \otimes \boldsymbol{\Lambda}\mathbf{V}^T\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda}\right] \\ &- \left[\mathbf{K}_{t_\star,t} \otimes \boldsymbol{\Lambda}\mathbf{V}^T\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda}\left[\mathbf{K}_{t,t} \otimes \boldsymbol{\Lambda}\mathbf{V}^T\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{K}_{t_\star,t} \otimes \boldsymbol{\Lambda}\mathbf{V}^T\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda}\right]\end{aligned} \tag{4.25}$$

Finally the posterior mean of the conditional distribution is is

$$\boldsymbol{\mu}_F = \mathbf{K}_{t_\star,t} \otimes \boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda}\left[\mathbf{K}_{t,t} \otimes \boldsymbol{\Lambda}\mathbf{V}^T\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{y}_q \tag{4.26}$$

and the covariance of the conditional distribution is

$$\begin{aligned}\boldsymbol{C}_F = &\mathbf{K}_{t_\star,t_\star} \otimes \boldsymbol{\Sigma} \\ &- \mathbf{K}_{t_\star,t} \otimes \boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda}\left[\mathbf{K}_{t,t} \otimes \boldsymbol{\Lambda}\mathbf{V}^T\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda} + \sigma^2\mathbf{I}\right]^{-1}\left[\mathbf{K}_{t_\star,t} \otimes \boldsymbol{\Lambda}\mathbf{V}^T\boldsymbol{\Sigma}\right]\end{aligned} \tag{4.27}$$

Figure 4.2 shows the pictorial representation of intrinsic coregionalization kernel (Equation 4.5) $\mathbf{K}_f$ considering 20 transcription factors where covariance matrix $\boldsymbol{\Sigma}$ of was constructed using Ornstein-Uhlenbeck kernel and white kernel in additive form.

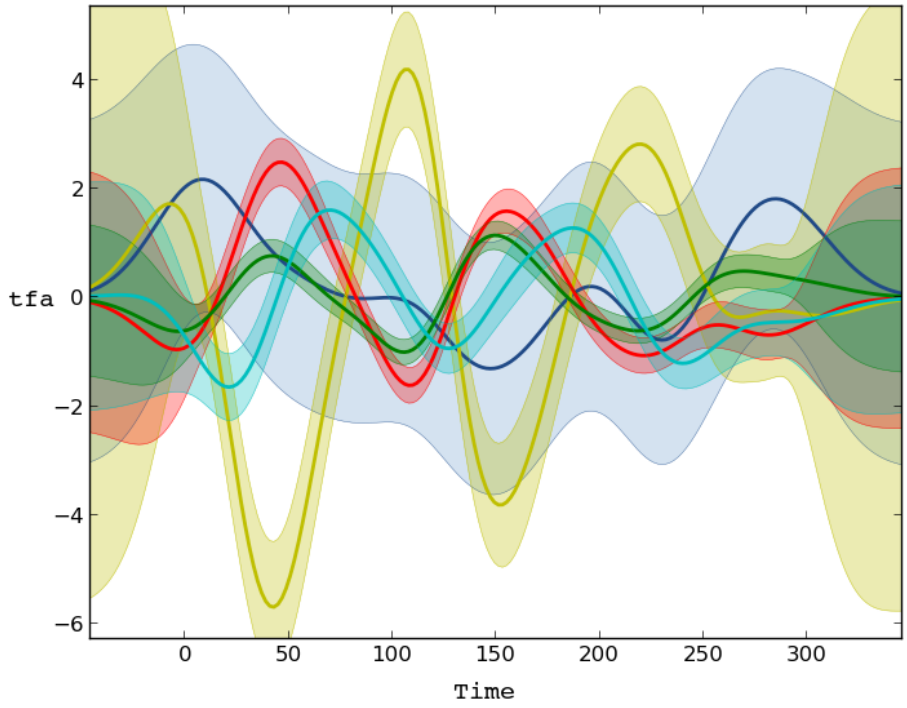


Fig. 4.3 Variation of activities of different transcription factors with the exponentiated quadratic kernel and white kernel in additive form. A solid line represents a posterior mean, and shaded area represents the 95% confidence interval for a specific transcription factor. Different colour shows the transcription factor activity of randomly picked transcription factors. Here we have noticed that for every transcription factors the activities are extremely smooth. In a practical case this behaviour is highly unlikely. So, a combination of exponentiated quadratic kernel and white kernel in additive form is not a very good choice to infer the transcription factor activity.

Rahman and Lawrence (2016) describes this methodology and a Jupyter Notebook demo using programming language *python* is available at Rahman and Lawrence (2014).

4.6 Dataset and Result analysis

Here in this experiment, we used the classic Spellman et al. (1998) yeast cell cycle dataset. The *cdc15* time series data has 23 time points.

The exponentiated quadratic kernel is very smooth kernel compared to Ornstein-Uhlenbeck kernel and perhaps is not a very good choice for the determination of actual transcription factors activities. Still, it can figure out the basic nature of the activities

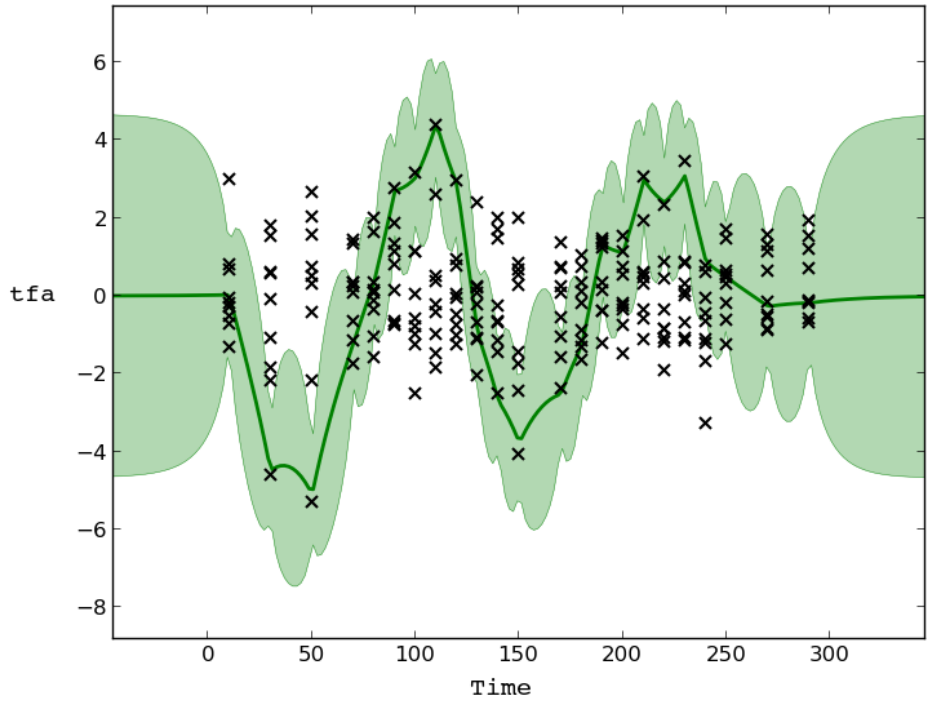


Fig. 4.4 Transcription factor activity of ACE2. The solid line represents a posterior mean of transcription factor activity, and shaded area represents 95% confidence interval. x-axis represents the time, and y-axis represents the transcription factor activity. Here a 'x' represents a gene expression level. At any time point, multiple gene expression levels ('x') represent the level of gene expression obtained from different genes which were regulated by transcription factor ACE2. We inferred the transcription factor activity considering all these expression levels.

with over smoothness. Figure 4.3 shows activities of different transcription factors while the model was developed considering exponentiated quadratic kernel with the White kernel in additive form.

Figure 4.4 shows transcription factor activity of ACE2. While developing the model we chose Ornstein-Uhlenbeck kernel and White kernel in additive form. In Figure 4.4, we noticed at any time point, multiple gene expression levels (represented by black 'x') were present, which shows the level of gene expression at that time point from different genes. A transcription factor can regulate multiple genes. To infer the activity of a transcription factor, we have to consider all these expressions. Here, we inferred the transcription factor activity of ACE2 considering all these expression levels. Earlier, we described (Chapter 3 Sec 3.8) how *Ornstein-Uhlenbeck* kernel can be used to model

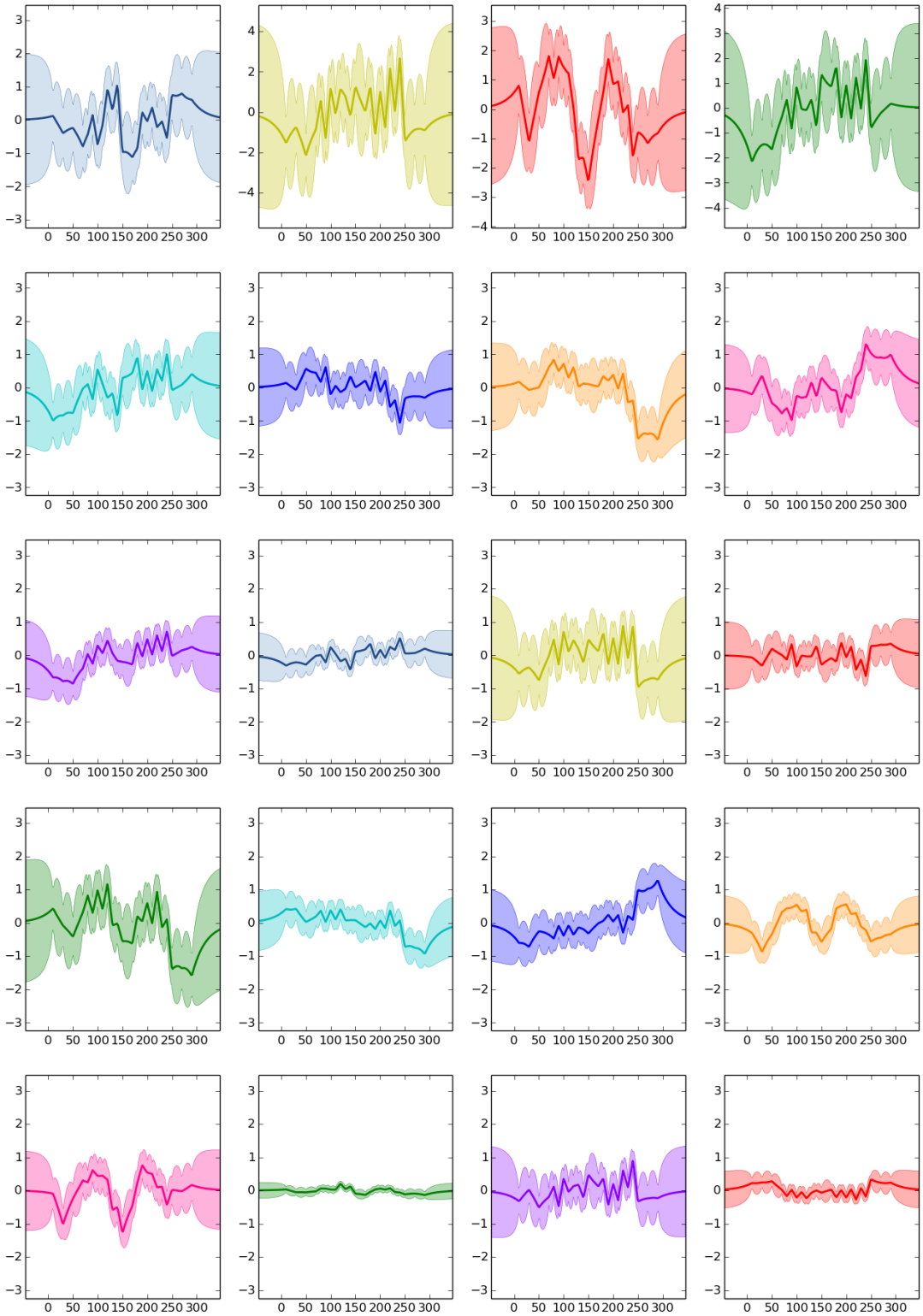


Fig. 4.5 Transcription factor activity of different transcription factors: individual plots shows the activity of the transcription factor. Here, the solid line represents a posterior mean and shaded area represents the 95% confidence interval. x-axis represents the time and y-axis represent the TFA.

the transcription factor activities. So, here we believe, the Ornstein-Uhlenbeck kernel will consider the basic nature of the transcription factors activity while White kernel will deal the noise associated the collected gene expression data.

Figure 4.5 transcription factor activity of different (arbitrarily chosen) transcription factors where we used our newly developed model. Here, as a prior, we used *Ornstein-Uhlenbeck* kernel and White kernel. In the plot, a solid line represented a posterior mean of transcription factors activity and shaded area represents the 95% confidence interval.

4.7 Discussion

At Chapter 3 Section 3.8 we noted that the linear Gaussian model is equivalent to a Gaussian process with a particular covariance function. In this chapter, we, therefore, build a model directly from the Gaussian process perspective. Here we designed a covariance function for reconstructing transcription factor activities given gene expression profiles and a connectivity information between genes and transcription factors. In the beginning section (Sec 4.1) of the chapter, we assumed that the joint process across all transcription factor activities and across all time points might have some correlation, hence we introduced an intrinsic model of coregionalization for the joint process. We also introduced a computational trick, based on a judicious application of singular value decomposition, to enable us to efficiently fit the Gaussian process in a reduced 'TF activity' space.

Chapter 5

Clustering Gene Expression Data

The dynamic behaviour or analysis of time series data in particular clusters is important for exploring and understanding gene networks. In many conventional time series models, one of the key requirements is data with regular intervals. Gene expression experiments data with regular intervals might be less informative or may not be optimal from a statistical perspective or even may not be cost effective for various reasons. A model designed to obtain data with regular intervals may not elicit as much information as a method designed to collect pertinent special temporal features. Again, in many cases, multiple biological replicates are available when the same experiments are repeated multiple times. For these cases simply considering only one experiment or taking the mean values from different replicates may not be the best solution. Interesting information might be discarded while dealing only with one data set or with their mean values.

The aim of this chapter is to specify the significantly different genes that may affect the speed of ALS progression by building a new model. We used the Gaussian processes, and here we introduced *coregionalization* principle while developing the kernel of the Bayesian hierarchical Gaussian process model. We believe, there might be some degree of temporal continuity between different replicates, conditions and/or genetic backgrounds. So, the kernel designed considering *coregionalization* model will consider the shared information between those replicates and conditions of genetic background. We used *python* programming language based tool *GPy* (The GPy Authors (2014)), to develop our model. Later we optimised these models and compared them based on likelihood scores and selected the best.

Amyotrophic lateral sclerosis (ALS) is a diverse neurodegenerative disorder with around 10% of familial cases and the remaining sporadic. The disease is currently irreversible from the onset and heterogeneous with variable severity in terms of speed

of progression of the disease course. Injury and cell death of motor neurones in the brainstem, spinal cord and motor cortex are the main reasons of this relentlessly progressive disorder (Brockington et al. (2013); Ferraiuolo et al. (2011); Haverkamp et al. (1995); Peviani et al. (2010)). Among the familial ALS [fALS] 20% is caused by mutation in the $Cu/Zn\ Superoxide\ Dismutase1$ ($SOD1$) gene. The median survival of this lethal disorder is less than 5 years, only 20% of patients live longer than 5 years, and less than 10% of patients survive more than 10 years from the symptom onset (Beghi et al. (2011); Saccon et al. (2013)). The speed of disease progression is not clear from the biological basis. Even in fALS, affected members clearly show the clinical heterogeneity in terms of site of onset, age and progression rate of the disease. In a study, Camu et al. (1999) reported the presence of potential gene modifiers and pathways that particularly affect the disease phenotype. Mutation in the $SOD1$ gene notably characterised the distinctive nature by intrafamilial and interfamilial variabilities in the phenotype. Many of the clinical and pathological features of human ALS can be replicated very well by transgenic mice. These murine models also mimic the human disease and show the heterogeneity in the disease progression for the clinical phenotype. These variabilities may be related with expression levels of mutant $SOD1$ protein or specific $SOD1$ mutations (Turner and Talbot (2008)).

In a study Pizzasegola et al. (2009) reported that disease progression is much faster in $129Sv$ mice with the survival time of 129 ± 5 days, while the $C57$ mouse strain can survive 180 ± 16 days. Both the $129Sv$ and $C57$ carry the same copy numbers of human mutant $SOD1$ and express the same amount of mutant $SOD1^{G93A}$ messenger RNA in the spinal cord. Marino et al. (2015) reported about the differences in protein quality control of these mice models in terms of speed of progression of the disease course.

Here in this work, we built a mathematical model to cluster gene expression time series data using hierarchical Gaussian process. Then as a part of validation, we performed an investigation of these clusters. We have calculated the enrichment scores (Huang et al. (2009a, 2007)) for every cluster using $DAVID$[1] (Database for Annotation, Visualization and Integrated Discovery) (Huang et al. (2009b)) and identified clusters which have very high enrichment scores. We perform a gene ontology overrepresentation analysis, where we showed our clusters are less likely to be grouped by chance. We carried out further analysis of clusters with high enrichment score and which also demonstrated some interesting characteristics in their dynamic behaviour at the four-time stages (pre-symptom, onset, symptom and end-stage) of disease course.

[1]http://david.abcc.ncifcrf.gov/tools.jsp

Our functional annotation clustering and pathway analysis reveal some interesting information for a group of genes which might have some functionality for the speed of propagation of ALS particularly with reference to this specific type of mouse model.

5.1 Related Work

Gene expression time series data has been used extensively over the last few decades and implemented for *in-silico* experiments to investigate various fundamental biological processes. Among the many processes examined, some of the notable examples are cell cycle Spellman et al. (1998), cell signalling Barenco et al. (2006), regulatory activity Sanguinetti et al. (2006), and developmental process Tomancak et al. (2002). Gaussian processes have been applied to gene expression time series widely with several aims and analyses, such as transcription factor target identification (Honkela et al. (2010)), the inference of RNA polymerase transcription dynamics (Maina et al. (2014)), and ranking differentially expressed time series (Kalaitzis and Lawrence (2011)).

Hierarchical models can significantly improve the inference in the Bayesian statistical problems (Gelman et al. (2004)) while dealing with multiple related groups of data allowing an exchange of information. Inference on the whole structure of data is always preferable to a partial independent structure. Estimating replicate time shifts were proposed by Liu et al. (2010), where they used Gaussian process regression with uncertain measurement of mRNA expression. This method requires a large number of variables optimisation. Previously, Ng et al. (2006) also Medvedovic et al. (2004) used a clustering method to model replicates using a hierarchical structure. Both of the models compute the replicate variance as multivariate Gaussian around some gene-specific mean.

In a clustering application, Gaussian process regression could be useful for parsimonious temporal inference. Temporal covariance of genes within a cluster can be designed by adding a hierarchical layer, again covariance between multiple biological replicates can be constructed considering one more hierarchical layer (Hensman et al. (2013b); Menzefricke (2000)). While Gaussian process also overcomes the requirement of evenly spaced time points for time expression data. As a part of the motivation, first, we clustered the gene expression time series data of *C.elegans* that has been used earlier to analyse transcription factor activity. We used the method proposed by Hensman et al. (2013b). Figure 5.1 shows the cluster analysis result. Apparently, this is a flexible model for irregularly sampled replicated time series data.

Here we constructed a hierarchical Gaussian process (Hensman et al. (2013b)) based model to analyse the gene expression time series data collected from four mouse models with different genetic backgrounds ($129Sv$ and $C57$ with transgenic and non-transgenic). We also considered their replicates (four in our case) and built a covariance matrix based on their shared information and the time points were pre-symptom, onset, symptom and end stage of the disease course.

5.2 Methodology

5.2.1 Hierarchical Gaussian Process

Our gene expression time series came from four different genetic backgrounds or strains and there are four biological replicates. With the idea of Hensman et al. (2013b), every individual gene we can incorporate these in an hierarchical fashion. Let's consider the vector of measurements $\mathbf{y}_{nr}^{i}$ denote gene expression of n^{th} gene in the r^{th} biological replicate and i^{th} biological strain or condition. Measurements were made at different times and collected into the vector $\mathbf{x}_{nr}^{i}$, where n, r and i represent the same as before. The data for the n^{th} gene and i^{th} strain is denoted by $\mathbf{Y}_{n}^{i} = \{\mathbf{y}_{nr}^{i}\}_{r=1}^{R}$ and $\mathbf{X}_{n}^{i} = \{\mathbf{x}_{nr}^{i}\}_{r=1}^{R}$, where R is the total number of replicates. The data for the n^{th} gene is represented by $\mathbf{Y}_{n} = \{\mathbf{Y}_{n}^{i}\}_{i=1}^{S}$ and $\mathbf{X}_{n} = \{\mathbf{X}_{n}^{i}\}_{i=1}^{S}$, where S is the total number of strains or conditions.

Let's consider some underlying function $g_n(x)$ model gene expression activity of the n^{th} gene, we have other functions $e_{nr}(x)$ which consider r^{th} replicates and finally we have some other functions $f_{inr}(x)$ for the i^{th} condition of the genetic background. The Gaussian process models are given by

$$g_n(x) \sim \mathcal{GP}(\mathbf{0}, k_g(x, x')) \tag{5.1}$$

$$e_{nr}(x) \sim \mathcal{GP}(g_n(t), k_e(x, x')) \tag{5.2}$$

$$f_{inr}(x) \sim \mathcal{GP}(e_{nr}, k_f(x, x')) \tag{5.3}$$

For the input dataset $\mathbf{X}_n$ and hyperparameters $\boldsymbol{\theta}$ we can calculate the likelihood by

$$p(\mathbf{Y}_n|\mathbf{X}_n, \boldsymbol{\theta}) = \mathcal{GP}\left(\hat{\mathbf{Y}}_n|\mathbf{0}, \Sigma_n\right), \tag{5.4}$$

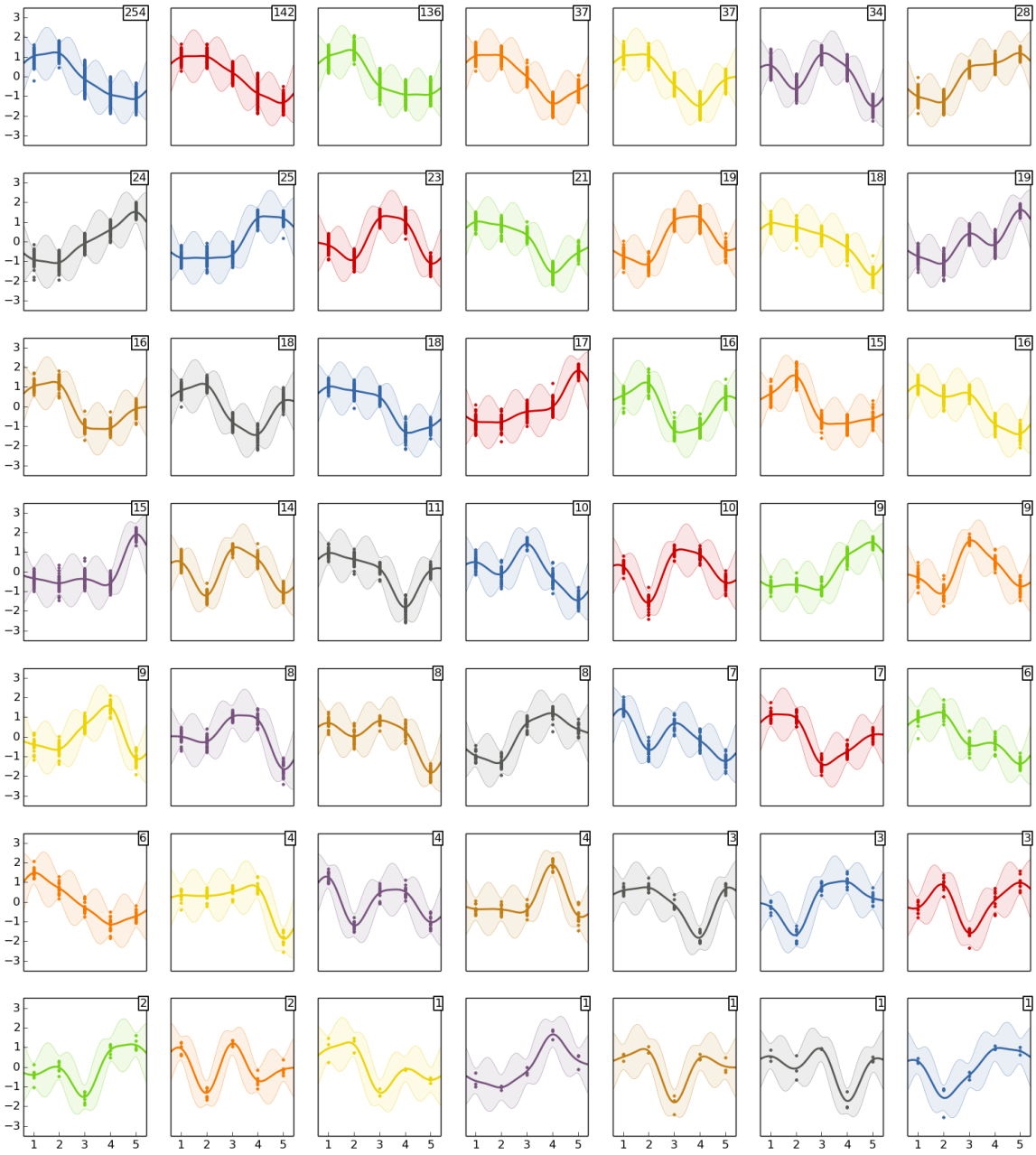


Fig. 5.1 Clustering gene expression data of *C.elegans* using the method proposed by Hensman et al. (2013b). Along x-axis the five time points are as described in Section 2.7.1 (shown in Figure 2.3). Number at the corner indicates number of genes belong to this cluster. Solid line represents a posterior mean and shaded area represents the 95% confidence interval. We used top most 10,000 differentially expressed genes from the total 15,000+ genes.

where- $\hat{\mathbf{Y}}_n = \left[Y_{n,1}^\top, Y_{n,2}^\top, ...Y_{n,S}^\top\right]^\top$ and $\boldsymbol{\theta}$ represents the hyperparameters for the covariance function k_g, k_e and k_f. The structure of the covariance matrix Σ_n for two genes n and n' are given by

$$\Sigma\left[n, n'\right] = \begin{cases} \Sigma_n + \mathbf{k}_h\left(x_n, x_{n'}\right), & \text{if } n = n'. \\ \mathbf{k}_h\left(x_n, x_{n'}\right), & \text{otherwise.} \end{cases} \tag{5.5}$$

While designing different kernels $\mathbf{k}$ we have used *coregionalization* model.

5.2.2 Kernel Design with Coregionalization

Gaussian process models have been already used to capture structure in the data arising from temporal correlation. Our innovation is to realise that there is additional correlation structure relating to the genetic background of the organism (in our case, the mice strains) and the status as control/experiment (in our case the presence or absence of the SOD1 mutation). By acknowledging such structure in the covariance matrix, we can increase the power of our method. Standard approaches force each of these conditions to be fully independent. Our model allows the correlation structure to be learned.

Our formalism for introducing correlations across conditions and strains is the *coregionalization* principle (Alvarez and Lawrence (2011)) that originates in geostatistics (Wackernagel (2003)). *Coregionalization* matrices allow us to share the information between genetic background and replicates. In machine learning language this approach is sometimes known as 'multi-task learning'(Bonilla et al. (2007)) where each condition and strain is assumed to be a different task. However, in statistical terms, it is simply a multivariate regression or a multiple output model.

An appropriate general model that can capture the dependencies between all the data points and conditions is known as the linear model of coregionalization (LMC) is a model where output is a linear combination of independent random functions. (A detail explanation of the *coregionalization* model is available at Alvarez and Lawrence (2011); Alvarez et al. (2012)). If we can consider our problem with a set of D output functions for $\mathbf{x}\epsilon\mathbb{R}^p$ input domain, then output function $\{f_d(\mathbf{x})\}_{d=1}^D$ of LMC can be expressed as

$$f_d(\mathbf{x}) = \sum_{q=1}^{Q} a_{d,q} u_q(\mathbf{x}) \tag{5.6}$$

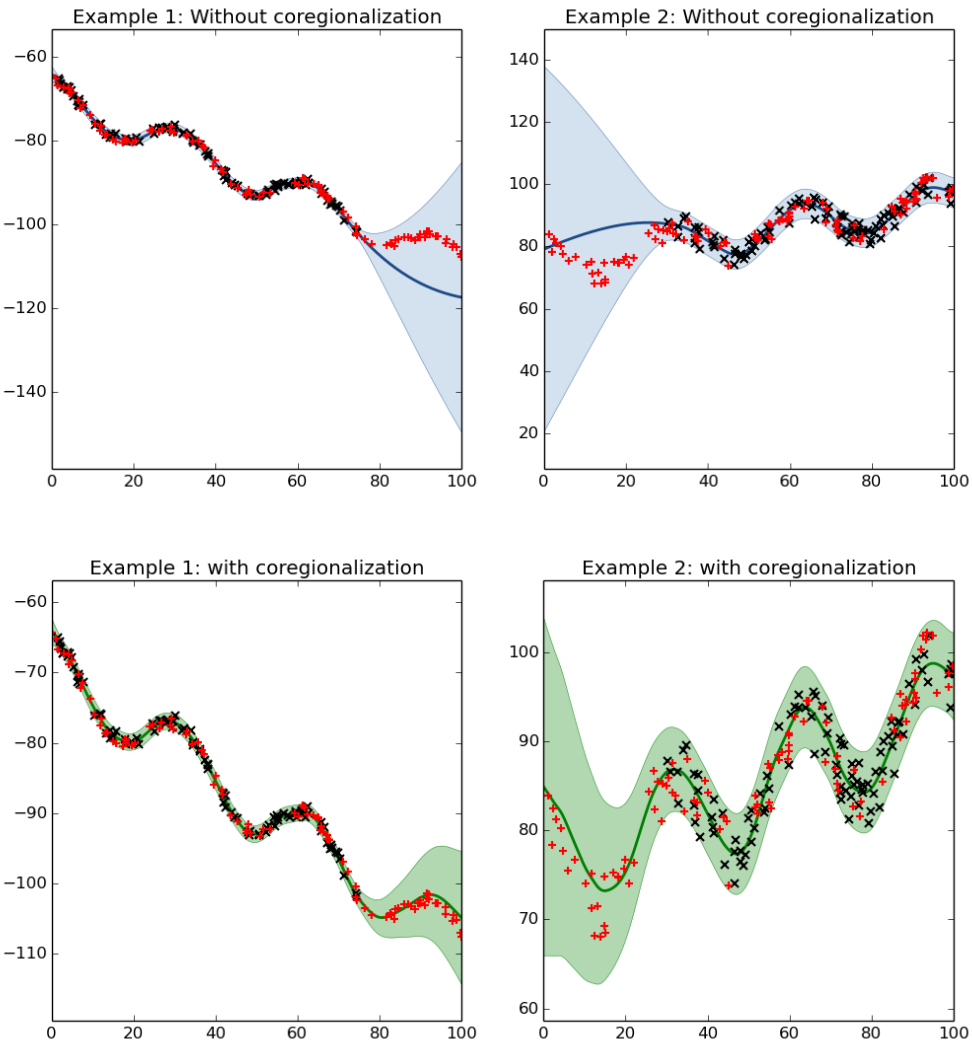


Fig. 5.2 A simple demonstration of regression using *coregionalization* model with the Gaussian process. Here, training data marked with black, while red represents test data. The solid line represents a posterior mean function, and shaded area represents the 95% confidence interval. The independent models (top-left and top-right) do not share information across outputs. The independent models tend to return to the prior assumptions in the regions where there is no training data specific to an output. While the coregionalized model (bottom-left and bottom-right) shares information across outputs. Here both outputs have associated patterns, where there is no training data the fit is better with the coregionalized model. A Jupyter Notebook demo of coregionalization using Gaussian processes is available at The GPy Authors (2014).

Here the interpretation is that $\{u_q^i(\mathbf{x})\}_{i=1}^{R_q}, i = 1, ..., R_q$ are a set of functions that each share the same covariance function (one can think of them as some form of underlying *latent* processes that determine system behaviour). The parameters $a_{d,q}$ represent the relationship between a given latent function, q and an observed condition and or strain. If we consider there can be several different covariance functions associated with separate latent sets then, equation 5.6 is expressed as

$$f_d(\mathbf{x}) = \sum_{q=1}^{Q} \sum_{i=1}^{R_q} a_{d,q}^i u_q^i(\mathbf{x}) \tag{5.7}$$

and the cross covariance function between $f_d(\mathbf{x})$ and $f_{d'}(\mathbf{x})$ in terms of the function $u_q^i(\mathbf{x})$ is given by

$$\begin{aligned} &\operatorname{cov}\left[f_d(\mathbf{x}), f_{d'}(\mathbf{x}')\right] = \\ &\sum_{q=1}^{Q} \sum_{q'=1}^{Q} \sum_{i=1}^{R_q} \sum_{i'=1}^{R_q} a_{d,q}^i a_{d',q'}^{i'} \operatorname{cov}\left[u_q^i(\mathbf{x}), u_{q'}^{i'}(\mathbf{x}')\right]. \end{aligned} \tag{5.8}$$

For the so-called homotopic case (Alvarez and Lawrence (2011); Wackernagel (2003)) the covariance matrix for the joint process $\mathbf{f}$ can be rewritten as a sum of Kronecker products, finally we can write the covariance as

$$\mathbf{K}_{f,f} = \sum_{q=1}^{Q} \mathbf{A}_q \mathbf{A}_q^\top \otimes \mathbf{K}_q = \sum_{q=1}^{Q} \mathbf{B}_q \otimes \mathbf{K}_q \tag{5.9}$$

where $\otimes$ represents Kronecker product, $\mathbf{A}_q \in \mathbb{R}^{D \times R_q}$ and $\mathbf{B}_q$ is the *coregionalization* matrix[2]. The positive semi-definite covariance functions of the latent processes, $k_q(\mathbf{x}, \mathbf{x}')$ can be chosen from a wide range of covariance functions.

Figure 5.2 shows a simple demonstration of regression using coregionalization model with Gaussian process. Here, the independent models do not share information across outputs and tend to return to the prior assumptions in the regions where there is no training data specific to an output. While the coregionalized model shares information across outputs. Here both outputs have associated patterns, where there is no training data the fit is better with the coregionalized model. We introduced coregionalization

[2]In Chapter 4 Section 4.1 we developed a model for transcription factor activity and used intrinsic coregionalization model. There the matrix $\mathbf{\Sigma}$ was termed as the coregionalization matrix. Coregionalization matrix $\mathbf{\Sigma}$ of 4.1 and coregionalization matrix $\mathbf{B}_q$ of Equation 5.9 have the similar realization.

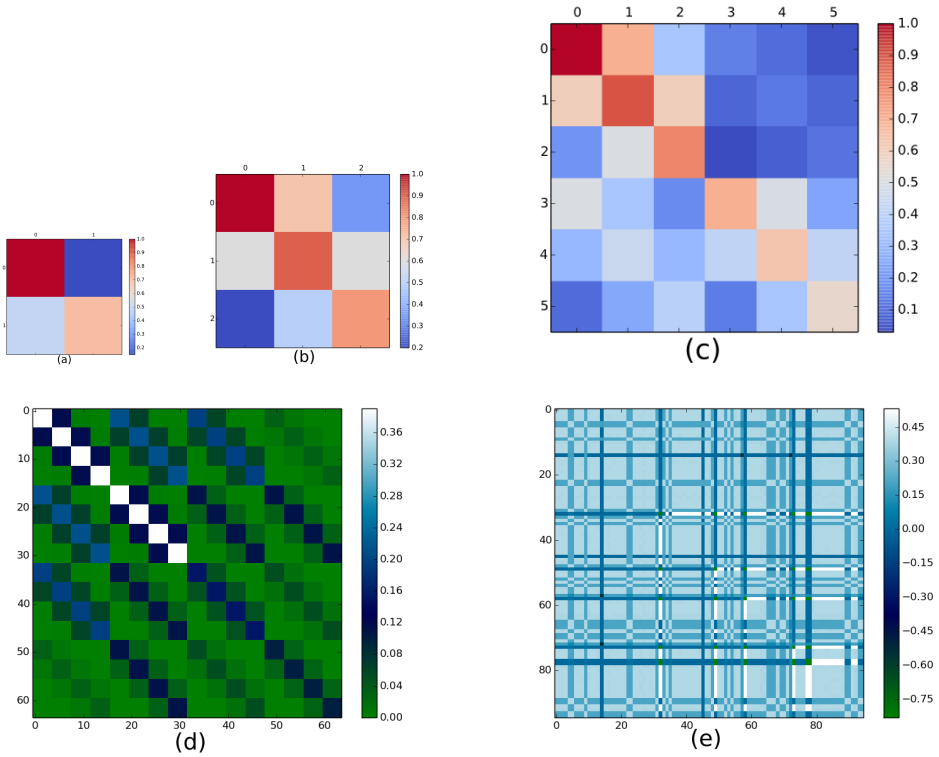


Fig. 5.3 A simple representation of *Coregionalization*: (a-c) demonstration of Kronecker product by tiling. (d) *Coregionalization* kernel in the input space with 64×64 dimensions; (4 strains $(129Sv - SOD1, 129Sv - Ntg, C57 - Ntg$ and $C57 - SOD1) \times$ 4 replicates $\times$ 4 time points or stages of the disease). With a closer look, we can find four primary segments, where every quarter can be treated as a strain; Each quarter has four more segments which indicate four replicates; Each replicate has a further four segments which represent four different disease stages or time points (e) kernel after optimization considering top most 100 (an arbitrary suitable number for visualization) differentially expressed genes. This is a realisation of covariance between genes, where every pixel is computed using the *coregionalization* kernel.

while developing the kernels in the hierarchical Gaussian process clustering. So, the information between the conditions, replicates and disease stages will be shared.

In our clustering problem, we used a combination of the exponentiated quadratic kernel (also known as squared exponential or RBF kernel) to describe the properties of the function which underlay each cluster. We used a white noise kernel in additive form to deal with the noise of the process. The experimental conditions of acquisition of gene expression measurements cannot be ideally controlled so that the measurements could be corrupted by noise, incorporated either at the biological origin or introduced in the measurement process. Figure 5.3 (a-c) demonstrate the Kronecker product by tiling, (d) shows the representation of the coregionalized kernel in the input space and (e) the representation of an optimised kernel where we considered only 100 (top most differentially expressed; an arbitrary number suitable for visualisation) genes.

5.2.3 Clustering

We aimed to discover groups of genes that were exhibiting the same functional behaviour across times and conditions. Our coregionalization approach allows us to cluster these similar functional behavioural genes through a mixture of Gaussian process models: each component is a function over time, genetic background and condition.

Partitioning genes into clusters are done by inference. Using Dirichlet process prior for mixing coefficients and partitioning Dunson (2010) proposed a method where the Gaussian process was used to model the function within a cluster. This mechanism leads to Gibbs sampling. In this process, at every Gibbs step, a gene is removed from the cluster and then reallocated it stochastically. The whole process can be slow in practice. A potentially improved model was proposed by Hensman et al. (2013b), where they consider the structure of covariance across the gene and separately across replicates. They used a variational lower bound for model inference. Each gene is placed in an individual cluster and later merged with a greedy selection process to maximise the log marginal likelihood of time series data. Hyperparameters are optimised when no merges are possible to improve the overall marginal likelihood. Then an expectation maximization algorithm is used with the new covariance function[3] (Hensman et al. (2013b)).

[3]The idea is implemented in a tool named named *GPClust*, available at https://github.com/jameshensman/GPclust

5.3 Dataset and Results

Microarray Data Analysis: We used the Affymetrix data from Nardo et al. (2013). In this experiment, spinal cord tissues were obtained from $C57$ and $129Sv$ transgenic $SOD1^{G93A}$ mice and age-matched non-transgenic littermates at the presymptomatic, the early symptomatic (onset) stage, symptomatic and end stage. The transcription profiles of laser captured motor neurons isolated from the lumbar ventral spinal cords of the rapid progressor ($129Sv - SOD1^{G93A}$), slow progressor ($C57 - SOD1^{G93A}$) mice at four stages of the disease (presymptomatic, onset, symptomatic, end stage) and respective non-transgenic littermates were generated using the murine GeneChip Mouse Genome 430 2.0 Plus (Affy MOE4302). We used *Bioconductor*[4] package *Puma* (Pearson et al. (2009)) to extract the point estimates of gene expression levels from the GeneChip Affymetrix data.

Select differentially expressed genes: All the gene expression time series data extracted from Affymetrix data might not be differentially expressed and filtering out the requisite genes is obvious. Considering the temporal nature of data Kalaitzis and Lawrence (2011) analysed time series gene expressions and filtered the quiet or inactive genes from the differentially expressed ones using Gaussian process. In addition identifying genes that have a good signal-to-noise ratio (SNR) is also used to filter down the total number of genes that need further analysis. We can rank the genes by the ratio of the mean replicate-wise variance to the variance of the replicate-wise means. In our analysis, we used a combination of both approaches. First, we made the initial ranking of the gene expressions (45,037 genes for our case) using the method of Kalaitzis and Lawrence (2011) and then we use the SNR to choose 10,000 genes [5] for further analysis. The gene expression levels of each replicate were normalised to zero-mean over all the samples before the filtering.

Cluster analysis: In the previous selection stage, we chose 10,000 genes[6] from the total of 45,037 probe sets which were more dynamically differentially expressed.

[4]Bioconductor is an open-source computational framework for the analysis of high-throughput genomic data in the *R* programming language.

[5]10,000 is an arbitrary choice. We did the similar experiments choosing 2,000, 5,000 and 15,000 genes and found similarity in the results with minor variations.

[6]The number of genes 10,000 is an arbitrary choice. We performed the similar experiments choosing 2,000, 5,000 and 15,000 genes and found similarity in the results with minor variations. We assumed, 10,000 most differentially expressed genes might be an appropriate choice for further analysis reducing the computational cost.

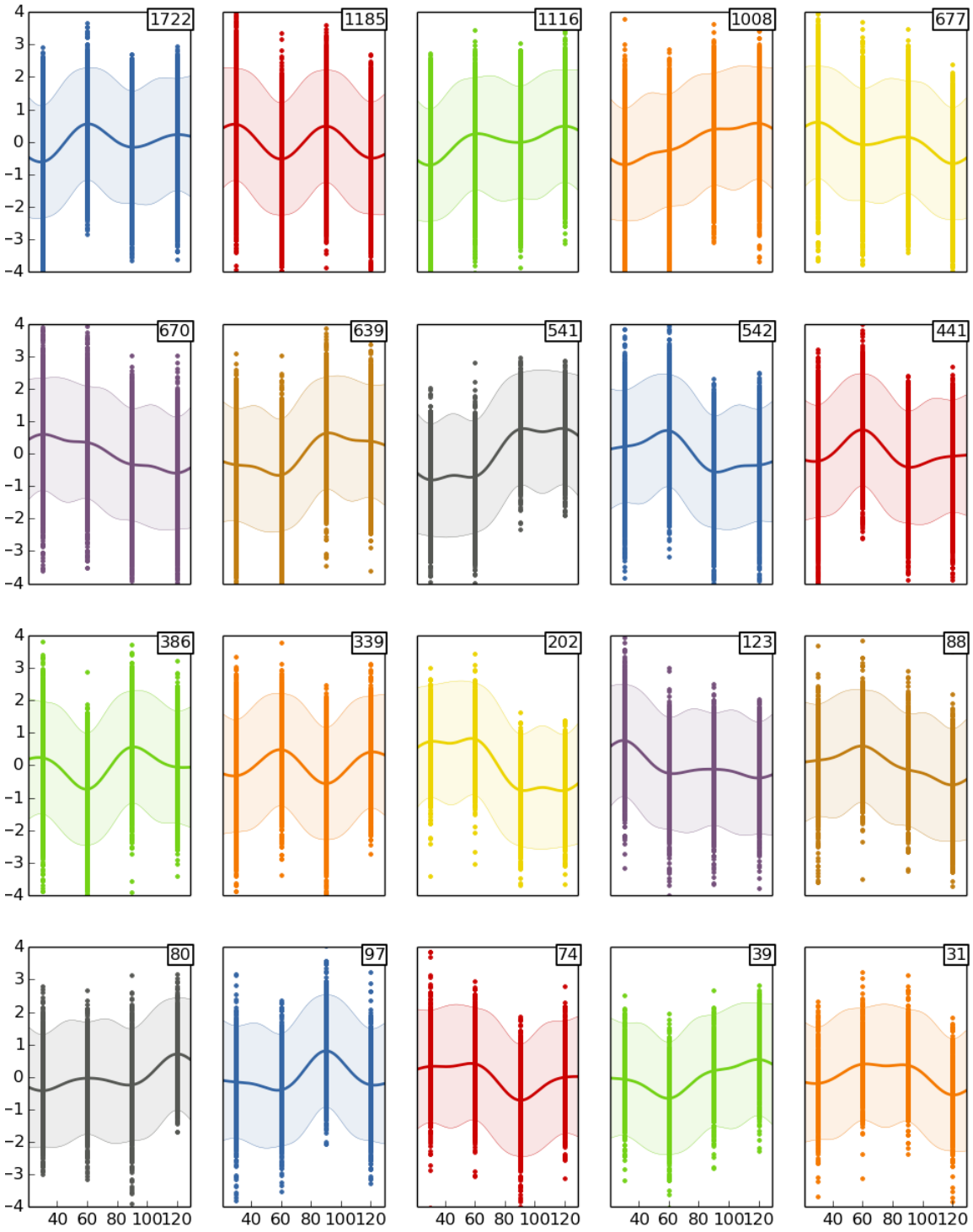


Fig. 5.4 Clustering gene expression data using the method proposed by Hensman et al. (2013b). Along x-axis the four-time stages are pre-symptom, onset, symptom and end-stage (all the data points together formed like solid vertical lines). A number at the corner indicates the number of genes belong to this cluster. The solid line represents a posterior mean and shaded area represents the 95% confidence interval. We used top most 10,000 differentially expressed genes and they were clustered among 20 clusters. The number of the clusters and the genes belongs to a specific cluster was selected by the algorithm.

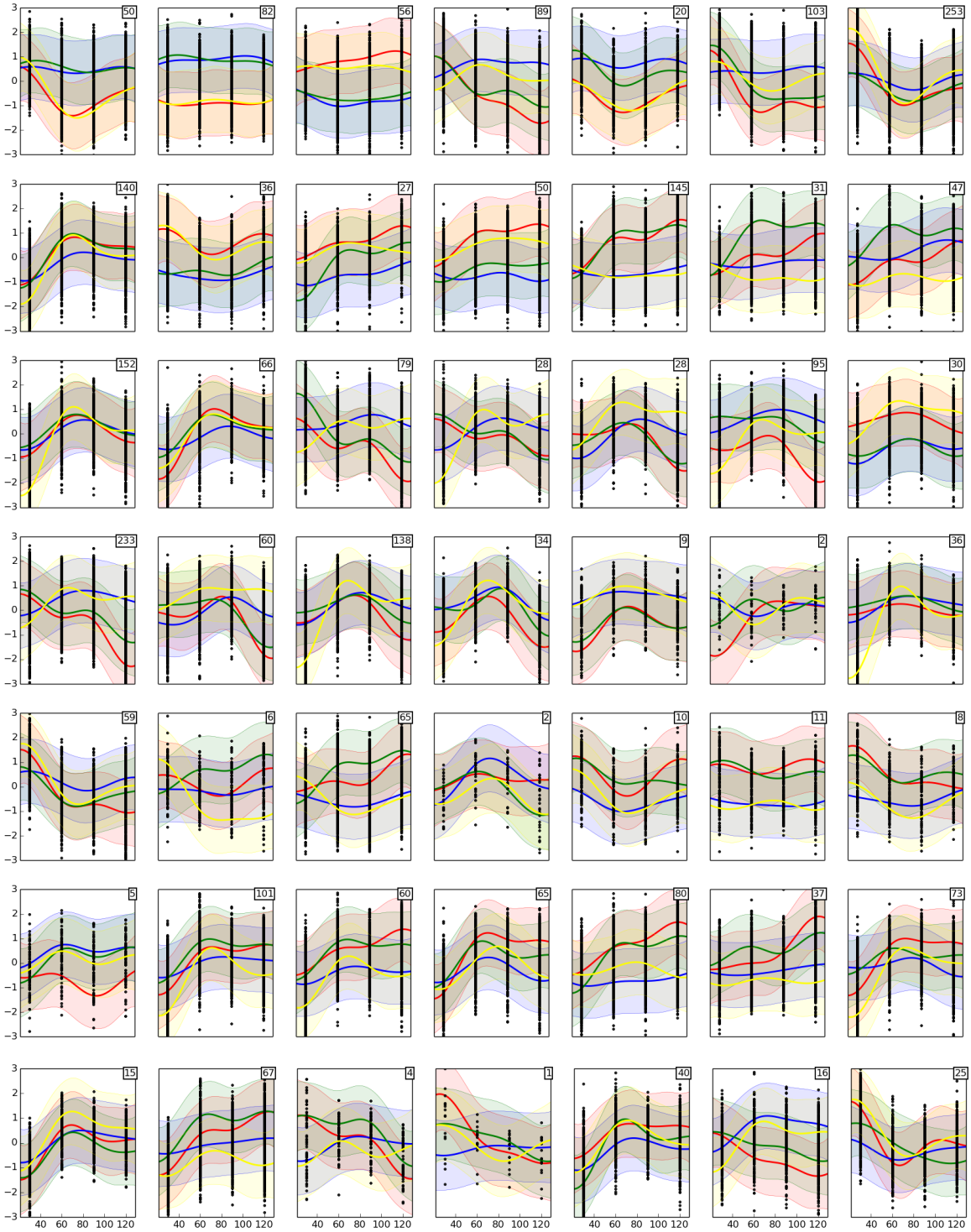


Fig. 5.5 Clustering genes expressions using a hierarchy of Gaussian processes. Some representative clusters from the 203 clusters generated (top to bottom, left to right: cluster 1 to 7, 16 to 22, 31 to 37, 46 to 52, 61 to 67, 76 to 82 and 196 to 202). Along x-axis the four-time stages are pre-symptom, onset, symptom and end-stage (all the data points together formed like solid vertical lines). Four different colours yellow, red, green and blue are representing four mouse strains $129Sv-SOD1, 129Sv-Ntg, C57-Ntg$ and $C57-SOD1$ respectively. A number at the corner indicates the number of genes belong to this cluster. Solid line represents a posterior mean function, and shaded area represents the 95% confidence interval.

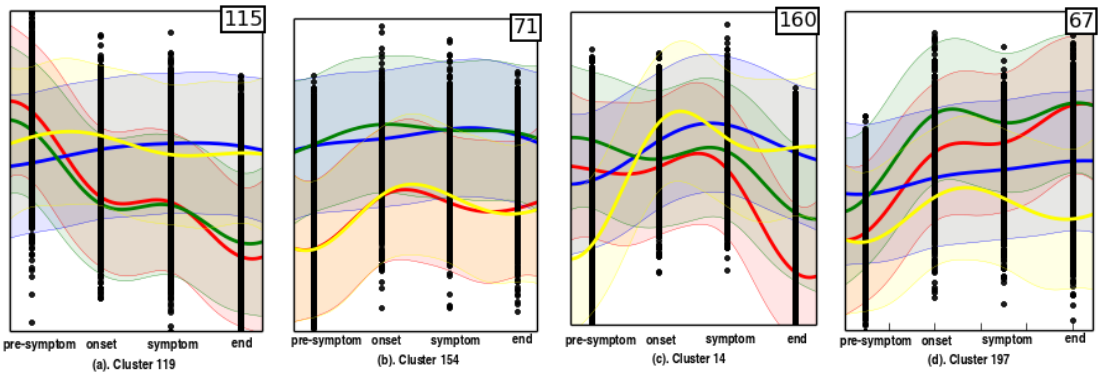


Fig. 5.6 Along x-axis of each figure four-time stages are pre-symptom, onset, symptom and end-stage. Four different colours yellow, red, green and blue are representing four mouse strains $129Sv-SOD1, 129Sv-Ntg, C57-Ntg$ and $C57-SOD1$ respectively. Examples of clusters where genes from different phenotypic background have different behaviour in time series expression. We used a simple numbering system to represent our clusters and here we are presenting (Figure left to right) cluster119, cluster154, cluster14 and cluster197. Cluster119 showing the clear separation between transgenic group ($129Sv-SOD1$ and $C57-SOD1$) with non-transgenic mouse model($129Sv-Ntg$ and $C57-Ntg$), while cluster154 separating mouse $C57$ from mouse $129sv$. Cluster14 and cluster197 showing the different characteristics of $129Sv-SOD1$ from other three models where it is increasing sharply or becoming very low respectively after the end stage.

We started with the baseline proposed by Hensman et al. (2013b) and clustered the genes systematically. We found the most differentially expressed genes were clustered into 20 clusters, where the first four cluster included 4,023 genes (1722, 1185, 1116 and 1008 respectively). A cluster with such a huge number of genes deemed insignificant for further analysis. Moreover, this method was not showing the individual dynamic behaviour for different genetic backgrounds. Figure 5.4 shows the first 10 clusters.

Later we applied our proposed hierarchical Gaussian process cluster model on the same 10,000 most differentially expressed genes and collected the results for further analysis. Figure 5.5 shows a small part of our result. For any individual graph, along with x-axis the four-time stages are pre-symptom, onset, symptom and end-stage. We have used four different colours (yellow, red, green and blue) to separate four mouse strains ($129Sv-SOD1, 129Sv-Ntg, C57-Ntg$ and $C57-SOD1$ respectively). Any individual cluster contains a number of genes which might be biologically associated or co-regulated and we mention the number of the genes belong to that cluster at the corner of the plot. In the plot, a solid line represents posterior mean function and shaded area represents 95% confidence interval. We have found a total of 203 different clusters with a variety of the number of genes. Many of the clusters indicated different dynamic behaviour of the gene set. Many of the clusters were attractive for further analysis but that is beyond the scope of this study. We included some examples in the Figure 5.6. We have limited our consideration to the clusters where the strain $129Sv-SOD1$ (yellow colour in our representation) had different characteristics and focussed our consideration for further analysis.

Enrichment score analysis: A typical biological process is regulated by a group of genes. If we apply a high-throughput screen technology then the co-functioning genes are more likely to appear together with a higher potential (or enrichment) score. These logical reasons instigate the analysis of a gene list or group of genes moving from individual gene-oriented view. The enrichment score is a quantitative measure derived from some well known statistical methods like Binomial probability, hypergeometric distribution, Chi-square, Fisher's exact test. In a previous study, Huang et al. (2009a) reported about 68 Bioinformatics tools to compute the enrichment score and grouped them into three major categories. *DAVID* Huang et al. (2009b) is a widely used tool developed based on Fisher's exact and extensively used for singular enrichment analysis (SEA) and modular enrichment analysis (MEA). We used DAVID on our clusters of genes to calculate the enrichment score for individual clusters. Figure 5.7 and Figure 5.8 shows the results of enrichment score analysis. In a functional annotation clustering

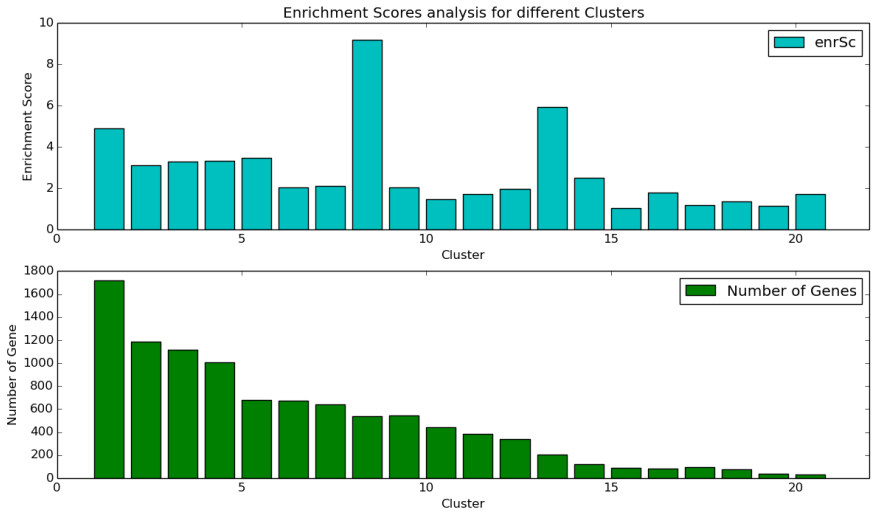


Fig. 5.7 Enrichment scores analysis without *coregionalization*. Figure at the top shows the enrichment score for different clusters, where x-axis is the cluster number and y-axis shows the enrichment score of that cluster. Figure at the bottom shows the number of genes belongs to any specific cluster.

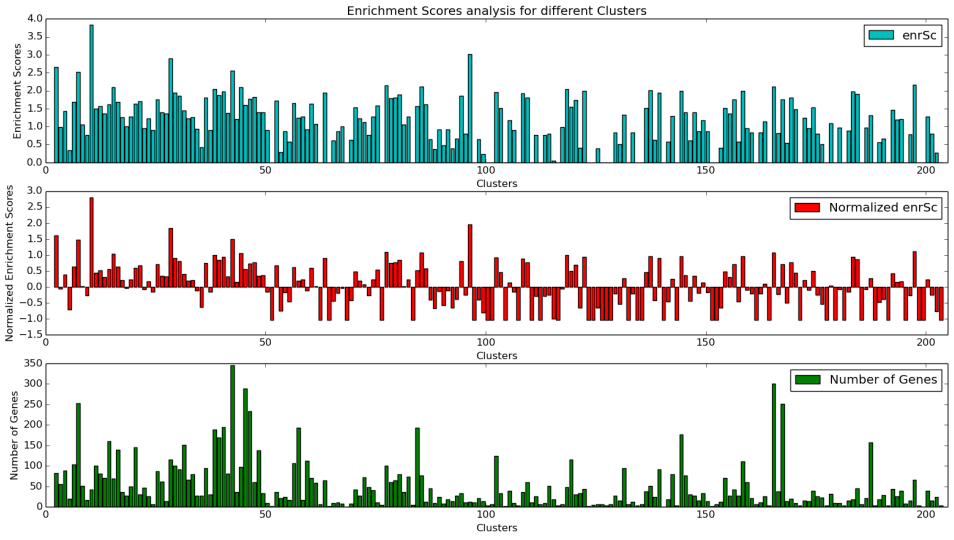


Fig. 5.8 Enrichment scores analysis. Figure at the top shows the enrichment score for different clusters where x-axis is the cluster number and y-axis shows the enrichment score of that cluster. Figure located in the middle shows the normalised score (mean enrichment score is 1.05). While the bottom figure shows the number of genes belongs to any specific cluster.

example Huang et al. (2007) analysed a group of genes and used enrichment score of 1.3 as a threshold value to decide whether a list of genes is enriched or not. Here for our 203 clusters, the mean enrichment score is 1.05 and we found at least 15 clusters have an enrichment score of ≥ 2. We choose these top 15 annotation clusters out of total 203 clusters for further analysis.

Pathway analysis: Pathway analysis allows us to gain an insight of the underlying biology of the differentially expressed genes. Pathway analysis can reduce the complexity and increase the explanatory power where high-throughput sequencing and gene profiling are used to investigate whether a gene or a list of genes have any roles for a phenotype or a given phenomena. It is also used for the analysis of gene ontology, physical interaction networks, the inference of pathways from expression and sequence data, and further comparisons. In a given condition it can identify the pathway by correlating information with a pathway knowledge base. We identified some clusters (which were deemed interesting in the cluster analysis and enrichment score analysis stages) and performed gene ontology enrichment analysis (one example is given at Table 5.2) and pathway analysis on individual clusters. We identified one of our clusters (cluster197; Figure 5.6) which were selected at the earlier stage for further analysis and had a relatively high enrichment score (2.16) is related to ALS. In previous study Brockington et al. (2013) reported about $SOD1$ related genes and ALS. One of the $SOD1$ genes, $Derlin-1$ (Prob ID: 1415693_at), can accumulate with other misfolded proteins and cause the neurone death and belongs to our chosen cluster.

The analysis carried out using the tool *DAVID* looks for shared ontological terms that bind the clustered genes in a biologically meaningful manner. It also highlights pathways or cellular mechanisms that are relevant to the paradigm being investigated. For example, in the *cluster 197* the gene $Derlin-1$ is shown to play an important role in ALS sitting at a position between the gene known to be involved in ALS, namely SOD1, and an important pathway that of the accumulation of misfolded proteins. Figure 5.9 shows the pathway analysis. Interestingly some other genes from the same cluster; $Stub1$, $Fzr1$, $Gabarapl1$, $Ap3m1$ and $Fuca1$ also play a role in some aspects of protein degradation pathways. Figure 5.10 shows the pathway analysis of Alzheimer's disease and Figure 5.11 shows the pathway analysis of Parkinson's disease. Both of these diseases are caused by cell death and the gene $Atp5j$ is related with them. Gene $Atp5j$ is present in our *cluster number 197*. Further investigation might find some further association between these genes and their protein products.

Annotation Cluster 1	**Enrichment Score: 2.16**	**Count**	**P_Value**	**Benjamini**
GOTERM_CC_FAT	organelle inner membrane	9	3.0E-5	4.5E-3
GOTERM_CC_FAT	mitochondrial inner membrane	8	1.6E-4	1.2E-2
GOTERM_CC_FAT	organelle membrane	12	2.8E-4	1.4E-2
GOTERM_CC_FAT	mitochondrial membrane	8	6.1E-4	2.3E-2
GOTERM_CC_FAT	mitochondrial envelope	8	8.7E-4	2.6E-2
GOTERM_CC_FAT	organelle envelope	9	1.2E-3	3.1E-2
GOTERM_CC_FAT	envelope	9	1.3E-3	2.7E-2
SP_PIR_KEYWORDS	mitochondrion inner membrane	5	3.8E-3	4.2E-1
GOTERM_CC_FAT	mitochondrial part	8	4.6E-3	8.3E-2
SP_PIR_KEYWORDS	mitochondrion	9	5.8E-3	3.5E-1
GOTERM_CC_FAT	mitochondrial membrane part	3	1.6E-2	1.9E-1
GOTERM_BP_FAT	transmembrane transport	6	3.1E-2	8.8E-1
GOTERM_MF_FAT	hydrogen ion transmembrane transporter activity	3	3.3E-2	9.9E-1
GOTERM_MF_FAT	monovalent inorganic cation transmembrane transporter activity	3	3.6E-2	9.4E-1
GOTERM_MF_FAT	inorganic cation transmembrane transporter activity	3	7.1E-2	8.9E-1
SP_PIR_KEYWORDS	transit peptide	5	7.4E-2	5.8E-1
GOTERM_CC_FAT	mitochondrion	10	7.6E-2	5.5E-1
UP_SEQ_FEATURE	transit peptide:Mitochondrion	5	1.0E-1	1.0E0
KEGG_PATHWAY	Oxidative phosphorylation	3	1.0E-1	8.6E-1
GOTERM_BP_FAT	generation of precursor metabolites and energy	3	2.6E-1	9.9E-1

Table 5.1 Gene ontology enrichment analysis from functional annotation clustering of cluster number 197 using DAVID.

Prob ID	Gene name	KEGG Pathway
1416143_at	Atp5j	Oxidative phosphorylation, Alzheimer's disease, Parkinson's disease, Huntington's disease
1415693_at	Derl1	Amyotrophic lateral sclerosis (ALS)
1416580_a_at	Stub1	Ubiquitin mediated proteolysis
1416375_at	Ap3m1	Lysosome
1417651_at	Cyp2c29	Arachidonic acid metabolism, Linoleic acid metabolism, Retinol metabolism, Metabolism of xenobiotics by cytochrome P450, Drug metabolism
1416565_at	Cox6b1	Oxidative phosphorylation, Cardiac muscle contraction, Alzheimer's disease, Parkinson's disease, Huntington's disease
1419353_at	Dpm1	N-Glycan biosynthesis
1417357_at	Emd	Hypertrophic cardiomyopathy (HCM), Arrhythmogenic right ventricular cardiomyopathy (ARVC), Dilated cardiomyopathy
1419451_at	Fzr1	Cell cycle, Ubiquitin mediated proteolysis, Progesterone-mediated oocyte maturation
1416047_at	Fuca2	Other glycan degradation
1416419_s_at	Gabarapl1	Regulation of autophagy
1416340_a_at	Man2b1	Other glycan degradation, Lysosome
1415917_at	Mthfd1	Glyoxylate and dicarboxylate metabolism, One carbon pool by folate
1418226_at	Orc2	Cell cycle
1416116_at	Orc3	Cell cycle
1416875_at	Parvg	Focal adhesion
1422525_at	Atp5k	Oxidative phosphorylation
1417242_at	Eif4a3	Spliceosome
1416754_at	Prkar1b	Apoptosis, Insulin signalling pathway
1416588_at	Ptprn	Type I diabetes mellitus
1416383_a_at	Pcx	Citrate cycle (TCA cycle), Pyruvate metabolism
1416625_at	Serping1	Complement and coagulation cascades
1423501_at	Max	MAPK signaling pathway, Pathways in cancer, Small cell lung cancer
1415891_at	Suclg1	Citrate cycle (TCA cycle), Propanoate metabolism
1415943_at	Sdc1	ECM-receptor interaction, Cell adhesion molecules (CAMs)

Table 5.2 Pathway analysis from functional annotation clustering of cluster number 197 using DAVID.

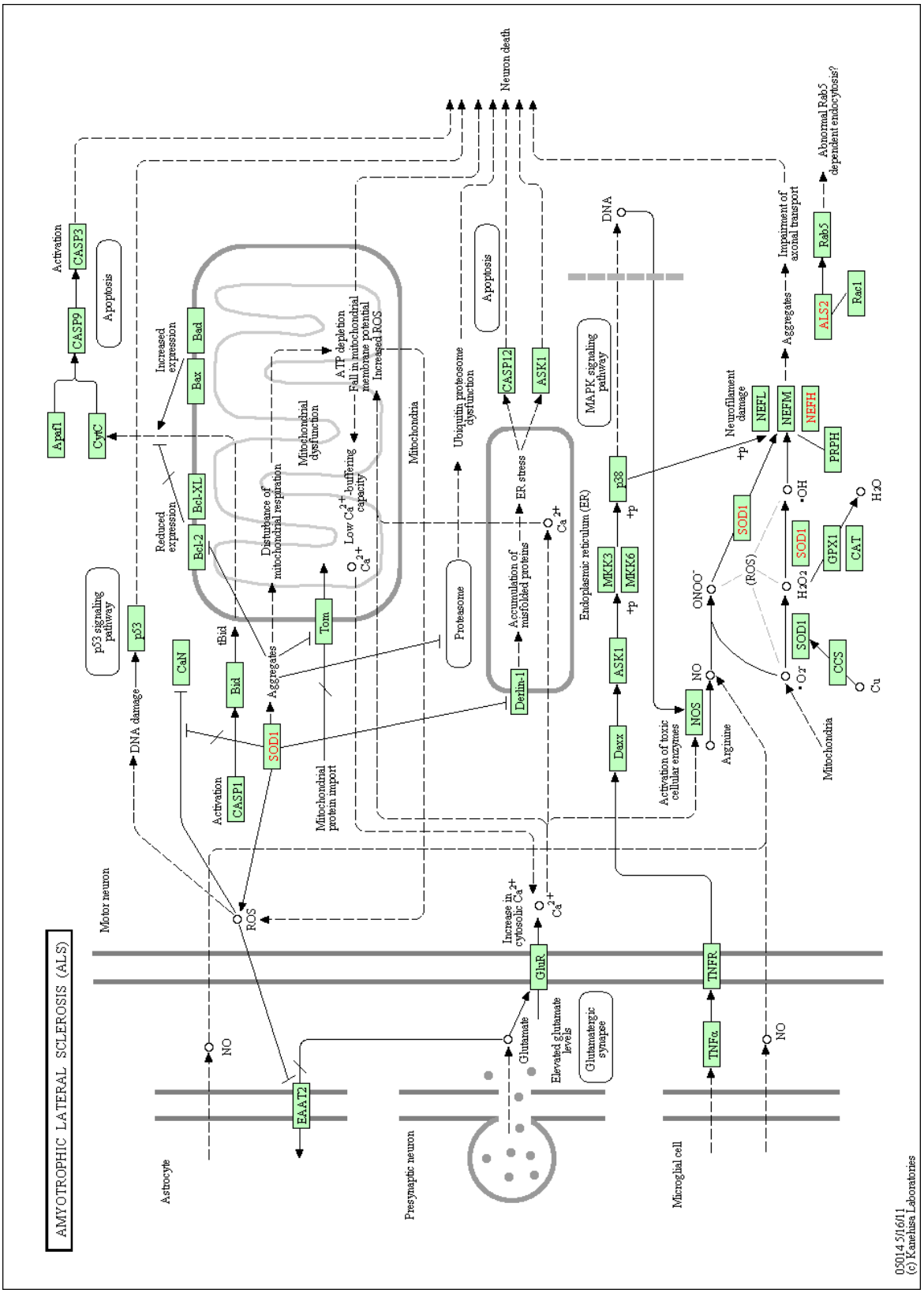


Fig. 5.9 KEGG pathway analysis of Amyotrophic lateral sclerosis (ALS).

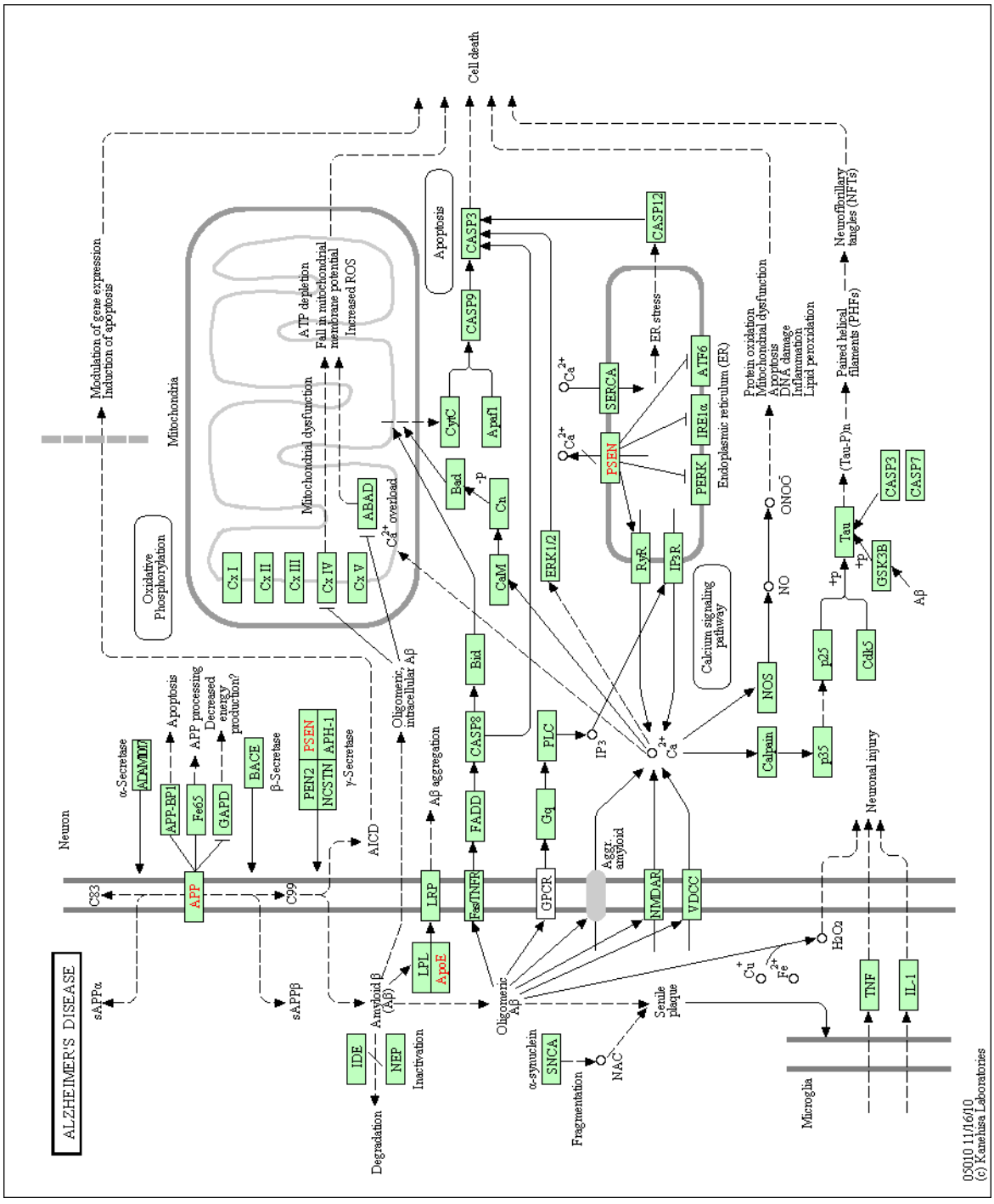


Fig. 5.10 KEGG pathway analysis of Alzheimer's disease.

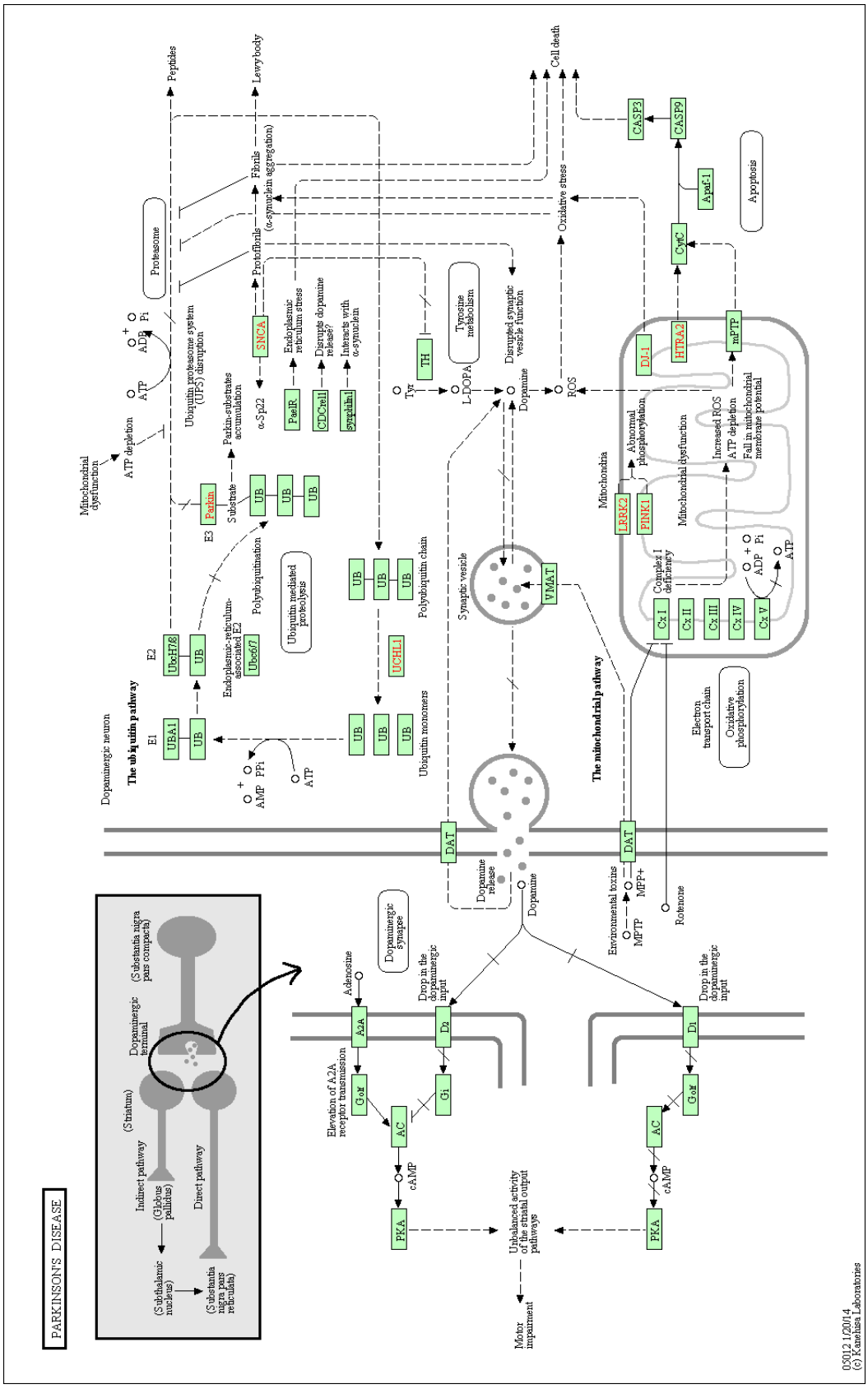


Fig. 5.11 KEGG pathway analysis of Parkinson's disease.

5.4 Gene Ontology overrepresentation analysis with Hypergeometric Test

In typical next generation high throughput experiments (e.g. gene expression microarray data analysis) clustering analysis aims to produce groups of genes based on specific criteria (e.g. differentially expressed genes). To learn more about the underlying biology and gain more mechanistic insights, overrepresentation analysis (ORA) is one of the most widely used approaches. Overrepresentation analysis uses the knowledge of the functional behaviours or characteristics of the genes and probes whether the identified gene groups or clusters functionally associated or not. Drăghici et al. (2003) first discussed the overrepresentation problem for a set, or cluster, of genes and proposed different statistical methods to overcome this issue. Overrepresentation analysis depends on the postulate whether a biological process has more identified genes than expected by chance alone, that biological process is probably linked to the experiment Grossmann et al. (2007).

Gene ontology terms provide a universal language for researchers to use that categorises each gene and its related function. A gene ontology (GO) term annotates a cluster of genes, shows the involvement in biological processes or indicates their known molecular functions or cellular component locations. Gene ontology terms represent the parent-child relationship by a directed acyclic graph, where a child represents a more precise biological classification than its parent or parents (in the case where a child has multiple parents). The annotation of a gene ontology term indicates automatic annotation to all the other ancestors of that gene ontology term based on the annotation propagation or true path rule. All genes that annotate to the gene ontology term *transcription factor activity, sequence-specific DNA* will also annotate to the gene ontology term *regulation of transcription, DNA templated* after some stages (layers of parent-child graph) which is annotated to gene ontology term *biological regulation* and finally *biological process*. Figure 5.12 shows the parent-child relationship by a directed acyclic graph for the gene ontology term *transcription factor activity, sequence-specific DNA* to the *biological process*.

There are many popular gene description databases or sites used for gene ontology overrepresentation analysis, such as the Gene Ontology Consortium Ashburner et al. (2000), the Database for Annotation, Visualization and Integrated Discovery (DAVID) Huang et al. (2007) and so on. In our analysis, we determined the most relevant genes associated with a given gene ontology term using the database of Huang et al. (2007).

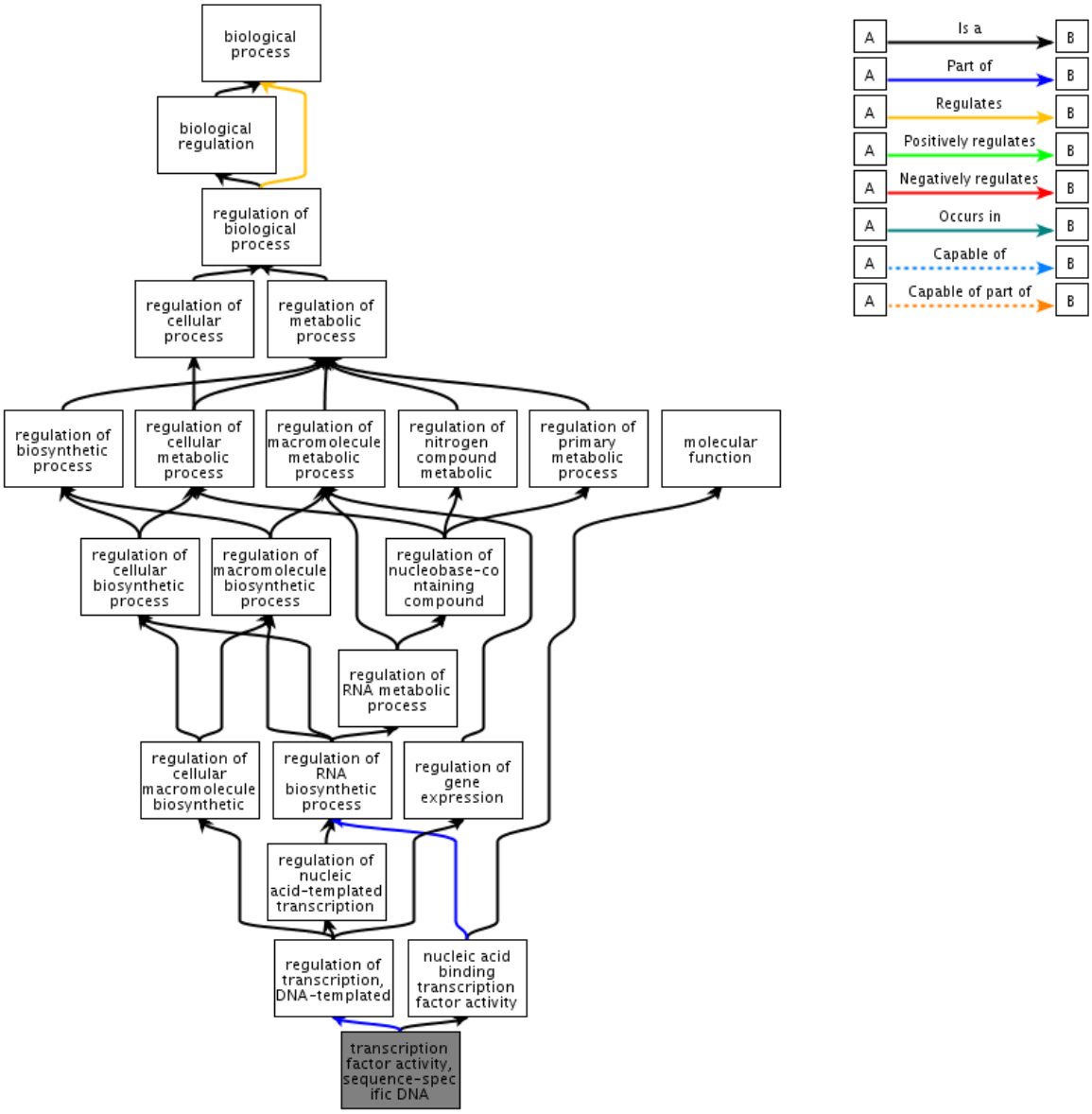


Fig. 5.12 The figure shows the parent-child relationship by a directed acyclic graph for the gene ontology term *transcription factor activity, sequence-specific DNA* to the *biological process*. All genes that annotate to the gene ontology term *transcription factor activity, sequence-specific DNA* will also annotate to the gene ontology term *regulation of transcription, DNA templated* which is annotated to *biological regulation* and finally *biological process*. Figure curtsey Huang et al. (2007) and http://www.ebi.ac.uk/QuickGO

In overrepresentation analysis, the most widely used statistical test is based on the hypergeometric distribution or its binomial approximation Beißbarth and Speed (2004); Drăghici et al. (2003); Lee et al. (2005); Martin et al. (2004). Let's consider in an experiment the full gene list is represented by G, A denotes a gene ontology term and S represents a cluster of genes obtained from a systematic clustering approach. The set of genes annotated to A is represented by g_A and set of genes belonging to the cluster S is denoted by g_S from that complete gene list G. The number of genes belonging to both A and S ($A \cap S$), denoted by n_A, indicates the representation of A in S. Under the null hypothesis that A and S are independent (i.e. the gene ontology term is not relevant to the gene cluster), n_A follows a hypergeometric distribution. The probability or p-value measuring the significance of association is the tail probability of observing n_A or more genes annotated by A in S is given by

$$p - \text{value} = \sum_{k=n_A}^{\min(g_A, g_S)} \frac{\binom{g_A}{k}\binom{G-g_A}{g_S-k}}{\binom{G}{g_S}} \tag{5.10}$$

where $\binom{p}{q} = \frac{p!}{q!(p-q)!}$ is the binomial coefficient. The closer the p-value is to zero, the more significant the particular gene ontology term associated with the group of genes is (i.e. the less likely the observed annotation of the particular gene ontology term to a group of genes occurs by chance) Ashburner et al. (2000).

In our analysis, we have considered a random number of gene ontology terms. In this thesis, we were interested about few gene ontology terms such as *transcription factor activity, sequence-specific DNA binding, cell death, immune response*. Including these gene ontology terms, we chose a total number of 20 terms for our further analysis. Earlier from the clustering approach of Hensman et al. (2013b) (we called the base model) we found a total number of 20 clusters and from our proposed model we have found a total number of 203 clusters. For the above selected 20 gene ontology terms, we performed the overrepresentation analysis using hypergeometric distribution and generated the heatmaps showed in Figure 5.13. As we know the closer the p-value is to zero (0), the more significant the particular gene ontology term associated with the group of genes is. In another way, the closer the p-value is to one (1), a particular gene ontology term to a group of genes is more likely occurs by chance. From the heatmap we can see for the cluster created by base model, at least a subset of clusters (9 out of 20; top heatmap of Figure 5.13 right portion of the dendrogram) are highly penalised. Whereas for our proposed model (bottom heatmap of Figure 5.13) there was no such penalization. Later for a fair trial, we randomly sampled 20 clusters out of our 203 clusters and constructed the heatmaps for those gene ontology terms and

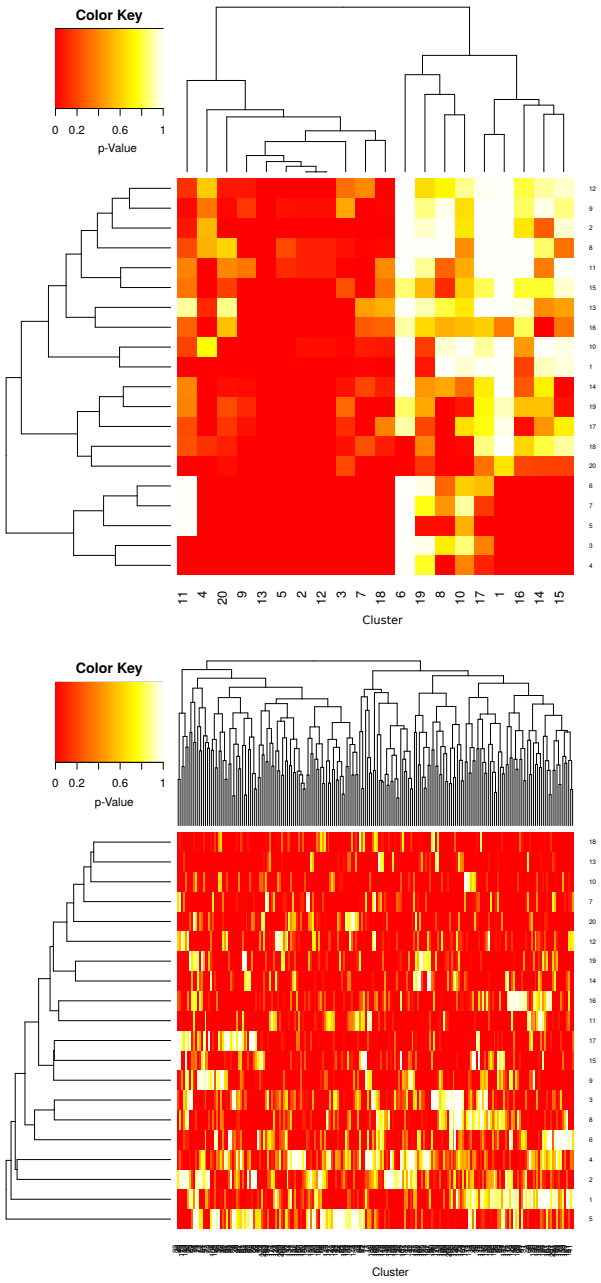


Fig. 5.13 Heatmaps generated using the hypergeometric test shows the overrepresentation pattern (top- the baseline model, bottom- our developed model using coregionalization). For an individual heatmap, a column represents a cluster and a row represents a specific gene ontology term. Here we chose 20 gene ontology terms randomly. Every pixel of the heatmap represents the p-value obtained from the hypergeometric test for a given gene ontology terms to a cluster of genes. In the heatmap *red* means relatively low expression (0) and *white* means relatively high expression (1). Colour key (located at the corner) shows the gradient.

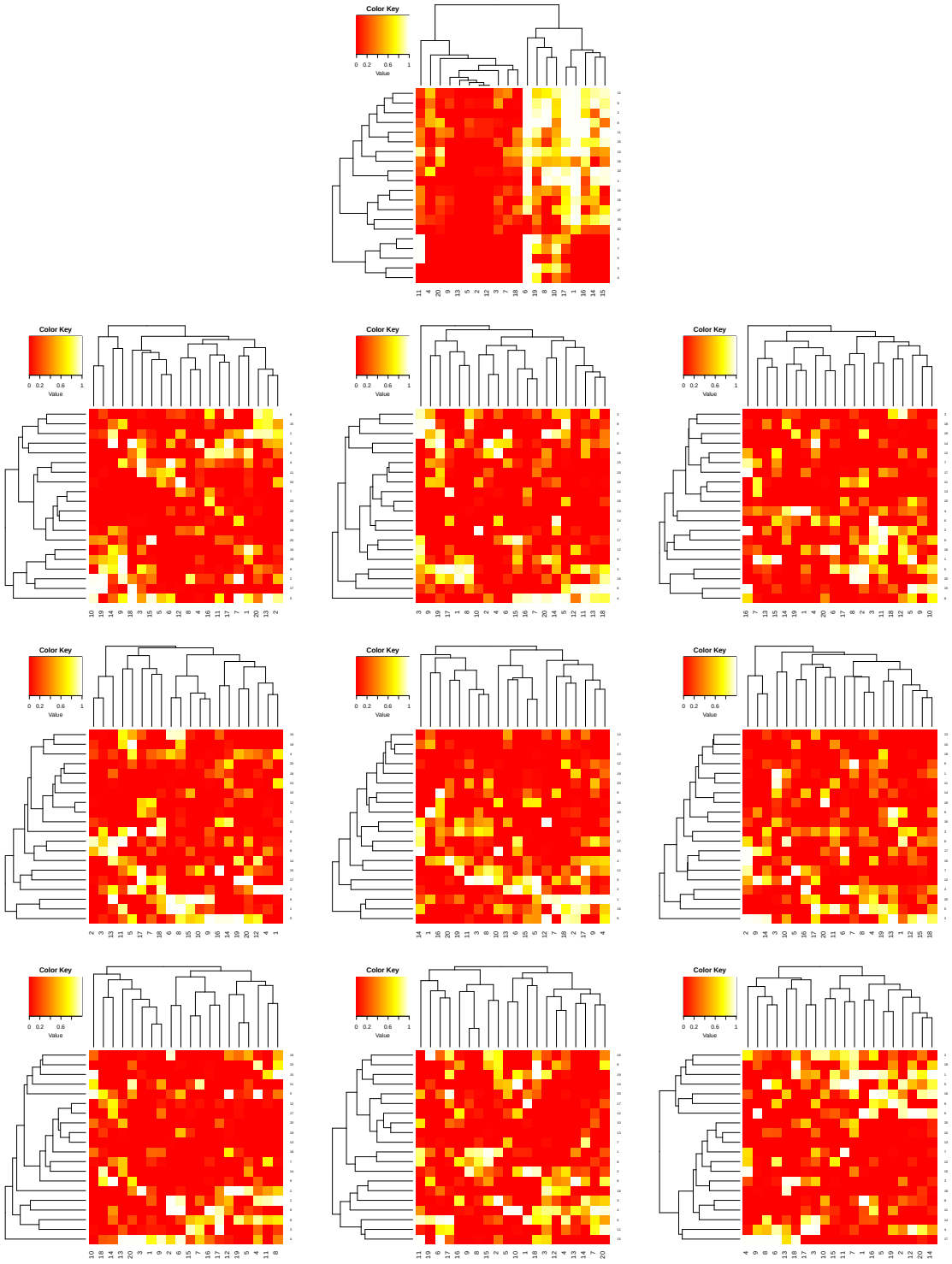


Fig. 5.14 Heatmaps generated using the hypergeometric test shows the overrepresentation pattern (top- the base model, bottom three rows- heatmaps generated from our model where for each map 20 clusters were selected randomly). For an individual heatmap, a column represents a cluster, and a row represents a specific gene ontology term. Every pixel of the heatmap represents the p-value obtained from the hypergeometric test for a given gene ontology terms to a cluster of genes. In the heatmap *red* means relatively low expression (0) and *white* means relatively high expression (1).

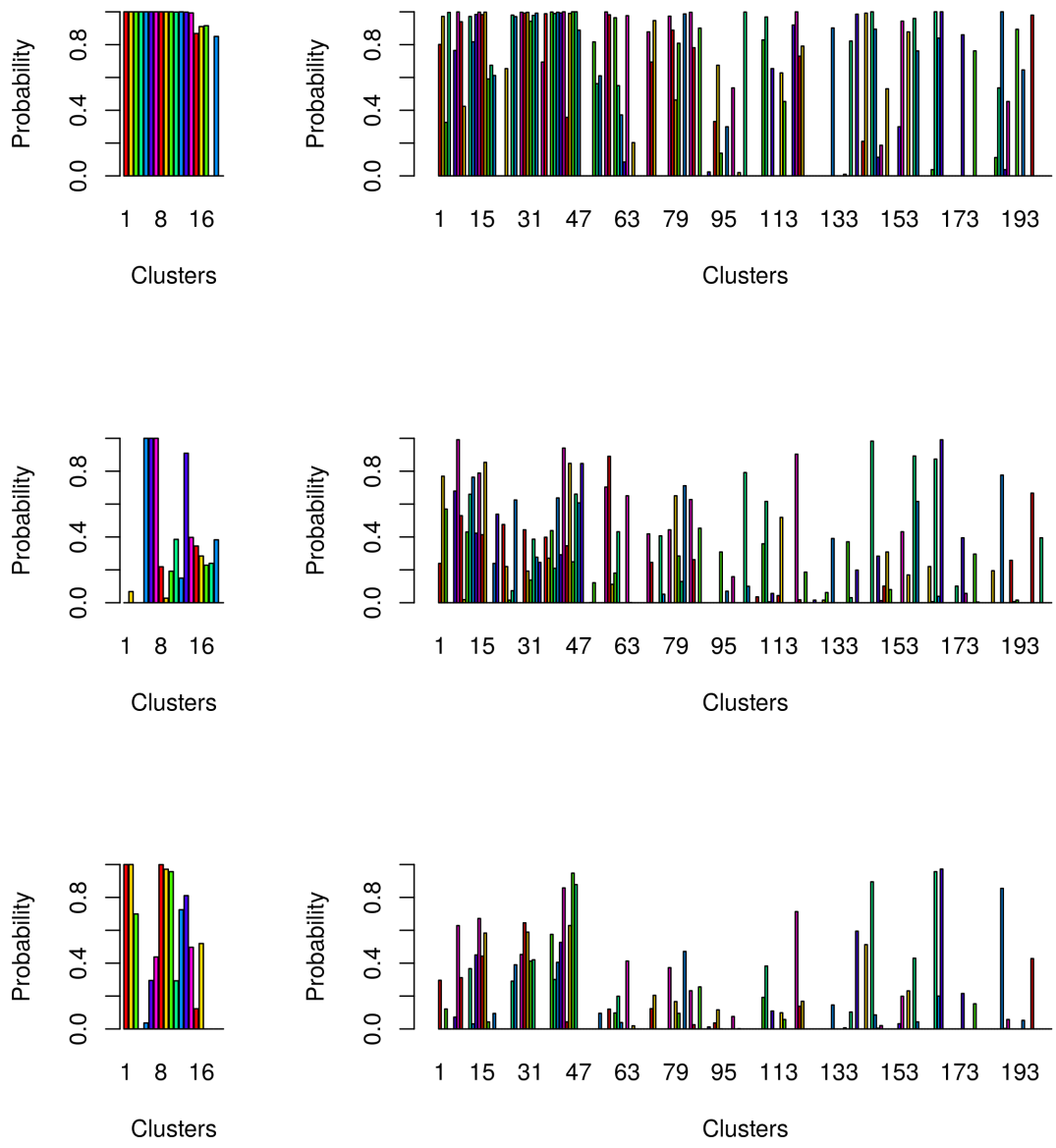


Fig. 5.15 Individual gene ontology overrepresentation analysis for gene ontology term *immune response* (top), *neurological system process* (middle) and *innate immune response* (bottom) using the hypergeometric test. Left column shows overrepresentation scores for the clusters obtained from the baseline model and right column shows the results of proposed model. For any individual plot x-axis represents the clusters and y-axis represents the probability or p-value.

showed in the Figure 5.14. Where we noticed there was no subset of clusters (like the base model) could be treated as highly penalised, or, in another way, grouped together just by chance.

Figure 5.15 shows the comparative study of penalization (by more and higher p-values) overrepresentation analysis for different gene ontology (*immune response* (top-row), *neurological system process* (middle-row) and *innate immune response* (bottom-row)) terms using the hypergeometric test. Left column shows overrepresentation scores of the clusters obtained from the baseline model and right column shows the p-values for proposed model.

GO Term	g_A	$P_{min_{Baseline}}$	$P_{min_{WithCorealization}}$	**GO Meaning**
GO:0002768	841	0.849994946	0.009751129	Immune response
GO:0008219	1450	0.362907818	0.021004198	Cell death
GO:0003700	680	0.0489814	0.002020474	Transcription factor activity
GO:0051252	1017	0.032311122	0.003872594	Regulation of RNA metabolic process
GO:0000398	233	0.028107818	0.002842114	mRNA splicing
GO:0015031	723	0.013961439	0.005218494	Protein transport
GO:0043565	660	0.009939761	0.002853619	Sequence-specific DNA binding
GO:0006333	93	0.007166567	0.000506799	Chromatin assembly or disassembly
GO:0012501	752	0.005944224	0.005645826	Programmed cell death
GO:0002758	218	0.000164461	0.000668375	Innate immune response
GO:0006928	284	3.84E-05	0.000803479	Cell motion
GO:0016477	47	1.52E-05	2.16E-05	Cell migration
GO:0001764	47	1.52E-05	2.16E-05	Neuron migration
GO:0050877	340	4.15E-06	0.001152254	Neurological system process
GO:0051674	153	3.03E-07	0.00023249	Localization of cell

Table 5.3 A comparison of the minimum p-values (obtained from the hypergeometric test for a given gene ontology term and clusters obtained from a specific clustering method) between the base model proposed by Hensman et al. (2013b) ($P_{min_{Baseline}}$) and our proposed method ($P_{min_{WithCorealization}}$). Here g_A is the number of genes related to a given gene ontology term obtained from the gene ontology database DAVID Huang et al. (2007).

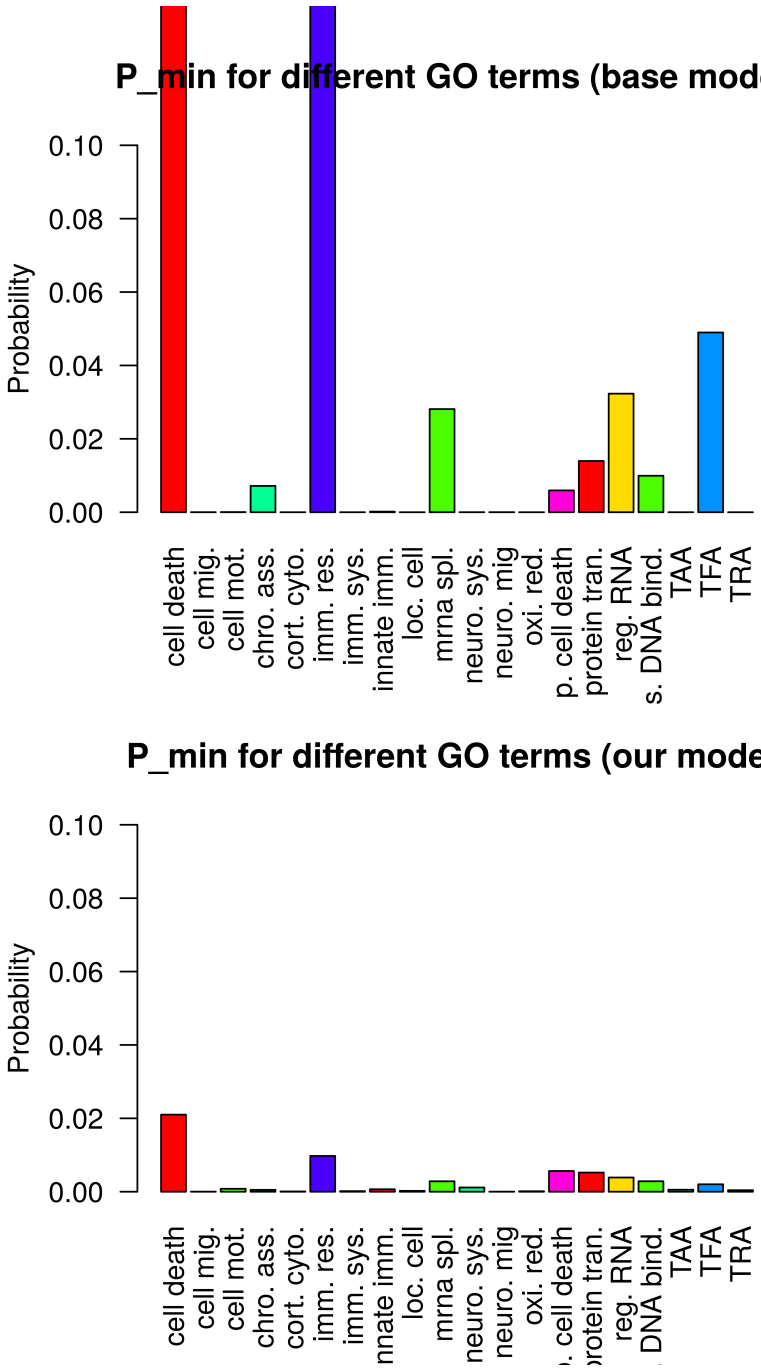


Fig. 5.16 A comparison of minimum p-values between two methods for a given gene ontology term. After determining the overrepresentation scores (p-values) from all clusters for a given gene ontology term, we have calculated the minimum scores (P_{min}) for both of the methods. For any individual plot x-axis represents different ontology terms, and y-axis represents the minimum p-values. Figure on the top shows scores for baseline model while the bottom one is for our developed method.

Table 5.3 shows the comparison of p-values obtained from the hypergeometric test for a given gene ontology term and clusters obtained from a specific clustering method. We have noticed, for most of the chosen gene ontology terms, our proposed model has a less p-value than the base model i.e. the clusters we found from our model is less likely occurred by chance.

After determining the overrepresentation scores (p-values) from all these clusters for a given gene ontology term we have calculated the minimum scores (P_{min}) for both of the methods. Figure 5.16 shows the comparison. We have noticed that for some gene ontology terms (such as *cell death, transcription factor activity, immune response* and so on) we discussed earlier in this thesis the baseline model was comparatively more penalized than our developed model i.e. genes we have clustered considering genetic backgrounds, disease states and replications using the coregionalization model are less likely to be clustered by chance.

Rather searching for statistical significance of each cluster, here in gene ontology overrepresentation analysis we compared two methods and showed that our model, in general, provides lower p-values, and therefore should result in more significant effect regardless of later corrections.

5.5 Discussion

We have performed genome-wide analysis to cluster genes systematically and analysed the rationale behind the variation in the speed of propagation for ALS. Our particular innovation was to include the condition and genetic background of the organisms within the underlying functional component of our clusters. This ensured that sub-groups, where the underlying expression behaved similarly, were more likely to cluster together. The hierarchical Gaussian process we used considers multiple replicates. For validation, we have used a widely acceptable Gene ontology and functional annotation tool to validate our clusters and their characteristics obtained from our model. We found a number of clusters are highly enriched. Gene expression time series characteristics curve and enrichment scores analyse helped us to narrow down our search and lead toward finding the lists of genes or clusters which could be involved in the speed of disease propagation. Our pathway analysis found a gene which is known to be involved in the ALS disease process. Few other genes are also related to the neural disorder. So, for further research exploring the biological characteristics of a few other clusters might be interesting. Here we started with a whole genome set and ended with a single gene directly related to disease. At the last part of this chapter, we made a comparison

of gene ontology overrepresentation analysis for a number of gene ontology terms using hypergeometric test where we found our clusters are less likely to be clustered just by chance.

This finding leads us to conclude that the model we have developed based on the Gaussian process can cluster the genes successfully and they are very much informative. These clusters can be useful for further analysis. Even the model we have developed using hierarchical Gaussian process could be useful to investigate other biological activity where clustering from shared information is required.

Chapter 6

Conclusions and Future work

Over the last few decades, Machine learning has become one of the central components of information technology, though mostly hidden part of our life. The increasing availability of very high dimensional data, with diverse characteristics and growing complexity, there is a good reason to believe that smart data analysis has become a necessary ingredient for technological progress and achieve the wisdom. Machine learning is a joint field of artificial intelligence and modern statistics, mostly focused on the design and development of models, algorithms and techniques to extract information automatically from data. Data modelling with the Gaussian process is a state of the art technique in the wider community, and in practice turned to multidisciplinary. Our main focus of this thesis was to achieve few goals by building Gaussian process models on transcriptome data and analyse their behaviour. Here, the final chapter of this thesis aims to summarise the key ideas and main contributions of the previous chapters and consider possible directions for future work.

6.1 Summary of the Specific Contributions

Chapter 2: We have reimplemented the tool *ChipDyno* using *R* programming language with the aim to make it public through an open source platform via GitHub. Earlier the dynamics of TFAs were obtained for a unicellular microorganism (yeast), our tool modelled transcription factor activities for a multicellular eukaryote (*C.elegans*). We constructed our connectivity information between genes and transcription factors from the evidence of gene to gene interaction to model the TFAs. The probabilistic dynamic model for quantitative inference of TFA we described in this chapter has been used as the basis for the Chapter 3 and Chapter 4.

Chapter 3: In this technical background chapter we briefly described Gaussian processes, the regression problem and regression with Gaussian processes. The choice of the covariance function is a central step in modelling with a Gaussian process. We justified the rationale behind choosing the Ornstein-Uhlenbeck kernel to model the transcription factor activity using Gaussian processes. This analogy leads us to develop a special covariance function suitable for transcription factor activity analysis.

Chapter 4: The probabilistic transcription factor activity model with Markov property proposed by Sanguinetti et al. (2006) is a linear Gaussian model, which is equivalent to a Gaussian process model with a particular covariance function. We, therefore, built a model directly from the Gaussian process perspective. Here we designed a covariance function for reconstructing transcription factor activities given gene expression profiles and a connectivity information between genes and transcription factors. The joint process across all transcription factor activities and across all time points might have some correlation. Here we incorporated intrinsic model of coregionalization for the joint process. We also introduced a computational trick, based on judicious application of singular value decomposition, to efficiently fit the Gaussian process in a reduced 'transcription factor activity' space.

Chapter 5: We have performed genome-wide analysis to cluster genes systematically and analysed the rationale behind the variation in the speed of propagation for ALS. Our particular innovation was to include the condition and genetic background of the organisms within the underlying functional component of our clusters. This ensured that the underlying expressions behaved similarly were more likely to be clustered together. We used a widely acceptable gene ontology and functional annotation tool to validate our clusters and their characteristics obtained from our model. We made a comparison of gene ontology overrepresentation analysis for a number of gene ontology terms where we found our clusters are less likely to be clustered just by chance. Gene expression time series characteristics curve and enrichment scores analyse helped us to narrow down our search and lead toward finding the lists of genes or clusters which could be involved in the speed of disease propagation. Our pathway analysis found evidence of genes which are known to be involved in the disease process. The special covariance function we have developed for clustering considering models condition, genetic background, replicates and disease states with coregionalization could be useful to investigate other biological activity where clustering is required.

6.2 Future Work

Here we are going to set some possible directions of future work

Bridge between TFA and clustering: In Chapter 4, we developed a model to analyse a latent variable (transcription factor) and determined their dynamic behaviour using the gene expression time series data. In Chapter 5, we developed another model to cluster gene expressions considering the genetic background, model conditions and replicates. We aim to build a model 'as a whole', which will model the dynamics of latent factor (transcription factor) considering various genetic backgrounds, conditions and replicates and hence cluster the gene expressions based on both of the latent factors and shared information.

Validation of clustered genes: Differential multi information (DMI) (Gambardella et al. (2015)) value indicates the level of differential co-expression of the gene set among the two classes. Each gene set is associated with a DMI value, computed as the absolute difference of the Rényi mutual information (RMI) (Rényi (1960)) among diseases and among controls. A high DMI value means that the same genes are coexpressed (and thus co-regulated) in a different manner between the conditions. For this reason, the gene set is addressed as 'differentially coexpressed', and considered relevant for the analysed disease. On the contrary, if a gene set has a low DMI value, it means that the co-expression of the genes in the two biological conditions is almost the same. Thus the gene set is not relevant for the analysed disease since different biological conditions do not seem to affect the co-expression of that particular gene set. One of our future plans is to validate the cluster of genes by building a model with using DMI and RMI and calculate the associativity.

Big Data: In Chapter 3 we addressed data with a higher number of features might be an issue while modelling with Gaussian process. On the other side, due to the advancement of data acquisition techniques, every day the amount of data increasing tremendously. These data are well known by a fancy term *Big Data*. Knowledge extraction and interpretation of *Big Data* is a new challenge, which also triggered the demand for special algorithms or models. The generic inference and learning algorithms in Gaussian processes where we need to inverse the covariance matrix have $\mathcal{O}(N^3)$ runtime complexity and $\mathcal{O}(N^2)$ memory complexity. However, an increasing number of machine learning research (Dai et al. (2014); Hensman et al. (2013a)) has focused to overcome these problems even with $N > 10^6$ data size. Our clustering algorithm (we described in Chapter 5) targets multiple model conditions where data size may grow geometrically with model conditions. We aim to extend our clustering model presented in this thesis (Chapter 5), which will able to handle the *Big Data*.

Classification of Single Sweep LFPs: To decipher brain activity and unprecedented information high-resolution neuronal probes has been developed. In a certain investigation, data generated from this probe contain spiking activity as well as field potential. Local field potential (LFP) is the extracellular space around neurones. LFPs are used to investigate the information processing pathway among various cortical layers. An average of recorded signal leads toward discarding valuable information during the analysis of LFPs, while shapes in single-sweep signals play an important role to decipher different neuronal network activity. From a pool of single LFPs extracting the shape of LFPs is a clustering task. A commonly used method proposed by Mahmud et al. (2010) is a parametric model which used intelligent K-means (iK-means) clustering algorithm. In this method, the pool of signals was treated as an outcome of the single experiment though in practice they were collected from separate experiments. We believe, there might have some correlation between different experiments with a temporal condition. We already showed how coregionalization model with Gaussian process could improve the capability of clustering when information is shared or, they came from a temporal condition. We believe, a Gaussian process based non-parametric model with coregionalization may reveal some interesting insight while clustering single sweep LFPs.

Deep Gaussian Process: The *deep Gaussian process* (Damianou and Lawrence (2013)) formed of a cascade of hidden layers of latent variables where the output of any node from a certain layer acts as the input of the layer below. This output-input relation from consecutive layers forms a hierarchy. In the deep Gaussian process, the mapping between layers is governed by Gaussian process where data is modelled as an output of a multivariate Gaussian process. In the domain of deep learning the *deep Gaussian process,* is a very recent advancement and gaining its popularity day by day. In our thesis (at Chapter 5) we developed a mechanism to cluster transcriptomic data using hierarchical Gaussian process. A plausible advancement of this model might be a clustering model using *deep Gaussian processes*. However, such a model may be more difficult to implement because the interactions in a deep Gaussian process are non-additive and therefore may be difficult to understand as a result.

References

Abramowitz, M. and Stegun, I. A. (1965). *Handbook of Mathematical Functions*. Dover, New York. (page 60)

Adams, J. U. (2008). Transcriptome: Connecting the genome to gene function. *Nature Education*, 1(1):195. (page 7)

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002). *Molecular Biology of the Cell*. Garland Science, New York. (page 24)

Alon, U. (2006). An introduction to systems biology: design principles of biological circuits. *CRC Press*. (pages 14 and 16)

Alter, O. and Golub, G. H. (2004). Integrative analysis of genome-scale data by using pseudoinverse projection predicts novel correlation between DNA replication and RNA transcription. *Proceedings of the National Academy of Sciences USA*, 101(47):16577–16582. (page 30)

Alvarez, M. A. and Lawrence, N. D. (2011). Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, (12):1459–1500. (pages 92 and 94)

Alvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3). (pages 76 and 92)

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25(1):25–29. (pages 109 and 111)

Barenco, M., Tomescu, D., Brewer, D., Callard, R., Stark, J., and Hubank, M. (2006). Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(3):R25. (page 89)

Beghi, E., Chio, A., Couratier, P., Esteban, J., Hardiman, O., Logroscino, G., Millul, A., Mitchell, D., Preux, P.-M., Pupillo, E., Stevic, Z., Swingler, R., Traynor, B. J., den Berg, L. H. V., Veldink, J. H., and Zoccolella, S. (2011). The epidemiology and treatment of ALS: focus on the heterogeneity of the disease and critical appraisal of therapeutic trials. *Amyotrophic Lateral Sclerosis*, 12(1):1–10. (page 88)

Beißbarth, T. and Speed, T. P. (2004). Gostat: find statistically overrepresented gene ontologies within a group of genes. *Bioinformatics*, 20(9):1464–1465. (page 111)

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press. (page 53)

Bishop, C. M. (1999). Latent variable models. In *Learning in Graphical Models*, pages 371–403. MIT Press. (page 26)

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York. (page 54)

Bonilla, E. V., Agakov, F. V., and Williams, C. K. I. (2007). Kernel multi-task learning using task-specific features. (page 92)

Boulesteix, A.-L. and Strimmer, K. (2005). Predicting transcription factor activities from combined analysis of microarray and chip data: a partial least squares approach. *Theoretical Biology and Medical Modelling*, 2(23). (page 30)

Boyadjiev, S. and Jabs, E. (2000). Online mendelian inheritance in man (OMIM) as a knowledgebase for human developmental disorders. *Clinical Genetics*, 57(4):253–266. (page 25)

Brenner, S. (1974). The genetics of Caenorhabditis elegans. *Genetics*, 77:71–94. (pages 11, 12, and 34)

Brivanlou, A. H. and Darnell, J. E. (2002). Transcription signal transduction and the control of gene expression. *Science*, 295(5556):813–818. (page 14)

Brockington, A., Ning, K., Heath, P. R., Wood, E., Kirby, J., Fusi, N., Lawrence, N., Wharton, S. B., Ince, P. G., and Shaw, P. J. (2013). Unravelling the enigma of selective vulnerability in neurodegeneration: motor neurons resistant to degeneration in ALS show distinct gene expression characteristics and decreased susceptibility to excitotoxicity. *Acta Neuropathol*, 125(1):95–109. (pages 14, 88, and 103)

Brown, R. (1828). A brief account of microscopical observations made in the months of June, July and August, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies. (page 69)

Byerly, L., Cassada, R., and Russell, R. (1976). The life cycle of the nematode Caenorhabditis elegans. i. wild-type growth and reproduction. *Developmental Biology*, 51(1):23–33. (page 12)

Camu, W., Khoris, J., Moulard, B., Salachas, F., Briolotti, V., Rouleau, G., and Meininger, V. (1999). Genetics of familial ALS and consequences for diagnosis. french ALS research group. *Journal of the Neurological Sciences*, 165:s21–s26. (page 88)

Chen, N., Qian, Z., Nabney, I. T., and Meng, X. (2014). Wind power forecasts using Gaussian processes and numerical weather prediction. *IEEE transactions on power systems*, 29(2):656–665. (page 1)

Cossins, A. R., Murray, P., Hayward, S. A. L., Govan, G. G., and Gracey, A. Y. (2007). An explicit test of the phospholipid saturation hypothesis of acquired cold tolerance in Caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 104(13):5489–5494. (pages 47 and 48)

Dai, Z., Damianou, A., Hensman, J., and Lawrence, N. D. (2014). Gaussian process models with parallelization and GPU acceleration. *arXiv:1410.4984*. (page 121)

Damianou, A. C. and Lawrence, N. D. (2013). Deep Gaussian processes. *Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics*, pages 207 – 215. (page 122)

De, S., Lopez-Bigas, N., and Teichmann, S. A. (2008). Patterns of evolutionary constraints on genes in humans. *BMC Evolutionary Biology*, 8(1):275. (page 25)

Deisenroth, M. P. (2012). *Efficient reinforcement learning using Gaussian processes*. PhD thesis, Faculty of Informatics, Karlsruhe Institute of Technology. (page 55)

Deisenroth, M. P., Fox, D., and Rasmussen, C. E. (2014). Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423. (page 1)

Drăghici, S., Khatri, P., Martins, R. P., Ostermeier, G., and Krawetz, S. A. (2003). Global functional profiling of gene expression. *Genomics*, 81(2):98 – 104. (pages 109 and 111)

Dunson, D. D. (2010). Nonparametric Bayes applications to biostatistics. *Bayesian Nonparametrics, Cambridge University press*. (page 96)

Dynlacht, B. D. (1997). Regulation of transcription by proteins that control the cell cycle. *Nature*, 389:149–152. (page 25)

Einstein, A. (1905). Uber die von der molekularkinetischen theorie der warme geforderte bewegung von in ruhenden flussigkeiten suspendierten teilchen. *Teilchen., Annalen der Physik*, 322(8):549–560. (pages 53 and 70)

Einstein, A. (1926). *The theory of the Brownian movement*. Dover Publications, Inc. (page 53)

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95(25):14863–14868. (page 47)

Ek, C. H., Torr, P. H., and Lawrence, N. D. (2008). Gaussian process latent variable models for human pose estimation. In Popescu-Belis, A., Renals, S., and Bourlard, H., editors, *Machine Learning for Multimodal Interaction (MLMI 2007)*, volume 4892 of *LNCS*, pages 132–143, Brno, Czech Republic. springer. (page 55)

Ferraiuolo, L., Kirby, J., Grierson, A. J., Sendtner, M., and Shaw, P. J. (2011). Molecular pathways of motor neuron injury in amyotrophic lateral sclerosis. *Nature Reviews Neurology*, 7(11):616–630. (page 88)

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620. (page 9)

Furney, S. J., Higgins, D. G., Ouzounis, C. A., and López-Bigas, N. (2006). Structural and functional properties of genes involved in human cancer. *BMC Genomics*, 7(1):3. (page 25)

Gambardella, G., Peluso, I., Montefusco, S., Bansal, M., Medina, D. L., Lawrence, N., and di Bernardo, D. (2015). A reverse-engineering approach to dissect post-translational modulators of transcription factor's activity from transcriptional data. *BMC Bioinformatics*, 279(279). (page 121)

Gao, F., Foat, B. C., and Bussemaker, H. J. (2004). Defining transcriptional networks through integrative modeling of mRNA expression and transcription factor binding data. *BMC Bioinformatics*, 5(31). (page 30)

Geertz, M. and Maerkl, S. J. (2010). Experimental strategies for studying transcription factor–DNA binding specificities. *Briefings in Functional Genomics*, 9(5-6):362–373. (page 25)

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman and Hall / CRC. (page 89)

Gerstein, M. B., Lu, Z. J., Nostrand, E. L. V., and Cheng, C. (2010). Integrative analysis of the Caenorhabditis elegans genome by the modencode project. *Science*, 330. (page 12)

Gertz, J., Siggia, E. D., and Cohen, B. A. (2009). Analysis of combinatorial cis-regulation in synthetic and genomic promoters. *Nature*, 457:215–218. (page 26)

Gillespie, D. T. (1996). Exact numerical simulation of the ornstein-uhlenbeck process and its integral. *Physical review E 54*, 2:2084–2091. (page 70)

Grossmann, S., Bauer, S., Robinson, P. N., and Vingron, M. (2007). Improved detection of overrepresentation of gene-ontology annotations with parent–child analysis. *Bioinformatics*, 23:3024–3031. (page 109)

Haverkamp, L. J., Appel, V., and Appel, S. H. (1995). Natural history of amyotrophic lateral sclerosis in a database population validation of a scoring system and a model for survival prediction. *Brain*, 118:707–719. (page 88)

Hensman, J., Fusi, N., and Lawrence, N. D. (2013a). Gaussian processes for big data. *arXiv:1309.6835*. (pages 74 and 121)

Hensman, J., Lawrence, N. D., and Rattray, M. (2013b). Hierarchical Bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. *BMC Bioinformatics*, 14(252). (pages 89, 90, 91, 96, 98, 101, 111, and 115)

Honkela, A., Girardot, C., Gustafson, H., Liu, Y.-H., Furlong, E. E. M., Lawrence, N. D., and Rattray, M. (2010). Model-based method for transcription factor target identification with limited data. *Proceedings of the National Academy of Sciences*, 107(17):7793–7798. (page 89)

Horak, C. E. and Snyder, M. (2002). Chip-chip: A genomic approach for identifying transcription factor binding sites. *Methods in Enzymology*, 350:469 – 483. Guide to Yeast Genetics and Molecular and Cell Biology - Part B. (page 25)

Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2009a). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research*, 37(1):1–13. (pages 88 and 101)

Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2009b). Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature Protoc.*, 4(1):44–57. (pages 88 and 101)

Huang, D. W., Sherman, B. T., Tan, Q., Collins, J. R., Alvord, W. G., Roayaei, J., Stephens, R., Baseler, M. W., Lane, H. C., and Lempicki, R. A. (2007). The DAVID gene functional classification tool: a novel biological module-centric algorithm to functionally analyze large gene lists. *Genome Biology*, 8(9):R183. (pages 88, 103, 109, 110, and 115)

Hughes, T. R. and de Boer, C. G. (2013). Mapping yeast transcriptional networks. *Genetics*, 195(1):9–36. (page 26)

Ingalls, B. P. (2012). *Mathematical Modelling in Systems Biology: An Introduction*. MIT Press. (page 4)

Inmaculada, B. M., Philippe, V., Fabien, C., Laurent, J., and Albertha, W. (2007). EDGEdb: a transcription factor-DNA interaction database for the analysis of C. elegans differential gene expression. *BMC Genomics*, 8(1):21. (pages 38 and 42)

Johnson, D. S., Mortazavi, A., Myers, R. M., and Wold, B. (2007). Genome-wide mapping of in vivo protein-DNA interactions. *Science*, 316(5830):1497–1502. (page 25)

Kadonaga, J. T. (2004). Regulation of RNA polymerase II transcription by sequence-specific DNA binding factors. *Cell*, 116(2):247 – 257. (page 24)

Kalaitzis, A. A. and Lawrence, N. D. (2011). A simple approach to ranking differentially expressed gene expression time courses through Gaussian process regression. *BMC Bioinformatics*, 12(180). (pages 46, 48, 51, 89, and 97)

Karin, M. (1990). Too many transcription factors: positive and negative interactions. *The New Biologist*, 2(2):126–131. (page 14)

Kasowski, M., Grubert, F., Heffelfinger, C., Hariharan, M., Asabere, A., Waszak, S. M., Habegger, L., Rozowsky, J., Shi, M., Urban, A. E., Hong, M.-Y., Karczewski, K. J., Huber, W., Weissman, S. M., Gerstein, M. B., Korbel, J. O., and Snyder, M. (2010). Variation in transcription factor binding among humans. *Science*, 328(5975):232–235. (page 26)

Keller, A. D. (1994). Specifying epigenetic states with autoregulatory transcription factors. *Journal of Theoretical Biology*, 170(2):175–181. (page 24)

Kitano, H. (2000). Perspectives on systems biology. *New Generation Computing*, 18(3):199–216. (page 2)

Kitano, H. (2002). *Systems Biology: Toward system-level understanding of Biological Systems*. MIT Press. (pages 2 and 5)

Kolmogorov, A. N. (1941). Interpolation und extrapolation von stationären zufälligen folgen. *Bull. Acad. Sci. (Nauk) U.R.S.S. Ser. Math.*, 5:3–14. (page 53)

Latchman, D. S. (1997). Transcription factors: an overview. *The International Journal of Biochemistry & Cell Biology*, 29(12):1305–1312. (page 14)

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816. (page 28)

Lee, H. K., Braynen, W., Keshav, K., and Pavlidis, P. (2005). Erminej: Tool for functional analysis of gene expression data sets. *BMC Bioinformatics*, 6(1):269. (page 111)

Lee, I., Lehner, B., Crombie, C., Wang, W., Fraser, A. G., and Marcotte, E. M. (2007). A single network comprising the majority of genes accurately predicts the phenotypic effects of gene perturbation in C. elegans. *Nature Genetics*, 40:181–188. (page 39)

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J.-B., Volkert, T. L., Fraenkel, E., Gifford, D. K., and Young, R. A. (2002). Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298(5594):799–804. (pages 11 and 75)

Lee, T. I. and Young, R. A. (2000). Transcription of eukaryotic protein-coding genes. *Annual Review of Genetics*, 34:77–137. (pages 14 and 24)

Levine, M. and Tjian, R. (2003). Transcription regulation and animal diversity. *Nature*, (424):147–151. (page 25)

Liao, J. C., Boscolo, R., Yang, Y.-L., Tran, L. M., Sabatti, C., and Roychowdhury, V. P. (2003). Network component analysis: Reconstruction of regulatory signals in biological systems. *Proceedings of the National Academy of Sciences*, 100(26). (pages 30, 31, and 38)

Liu, Q., Lin, K. K., Andersen, B., Smyth, P., and Ihler, A. (2010). Estimating replicate time shifts using Gaussian process regression. *Bioinformatics*, 26(6):770–776. (page 89)

Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., Mittmann, M., Wang, C., Kobayashi, M., Norton, H., and Brown, E. L. (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680. (page 8)

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. (pages 54 and 57)

Mahmud, M., Travalin, D., Bertoldo, A., Girardi, S., Maschietto, M., and Vassanelli, S. (2010). An automated classification method for single sweep local field potentials recorded from rat barrel cortex under mechanical whisker stimulation. *Journal of Medical and Biological Engineering*, 32(6):397–404. (page 122)

Maina, C. W., Honkela, A., Matarese, F., Grote, K., Stunnenberg, H. G., Reid, G., Lawrence, N. D., and Rattray, M. (2014). Inference of RNA polymerase II transcription dynamics from chromatin immunoprecipitation time course data. *PLOS Computational Biology*, 10(5). (page 89)

Marino, M., Papa, S., Crippa, V., Nardo, G., Peviani, M., Cheroni, C., , Trolese, M. C., Lauranzano, E., Bonetto, V., Poletti, A., DeBiasi, S., Ferraiuolo, L., Shaw, P. J., and Bendotti, C. (2015). Differences in protein quality control correlate with phenotype variability in 2 mouse models of familial amyotrophic lateral sclerosis. *Neurobiology of Aging*, 36:492–504. (page 88)

Martin, D., Brun, C., Remy, E., Mouren, P., Thieffry, D., and Jacq, B. (2004). Gotoolbox: functional analysis of gene datasets based on gene ontology. *Genome Biology*, 5(12):R101. (page 111)

Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in Applied Probability*, 5:439–468. (page 53)

Medvedovic, M., Yeung, K., and Bumgarner, R. (2004). Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8). (page 89)

Menke, D. B., Guenther, C., and Kingsley, D. M. (2008). Dual hindlimb control elements in the tbx4 gene and region-specific control of bone size in vertebrate limbs. *Development*, 135(15):2543–2553. (page 26)

Menzefricke, U. (2000). Hierarchical modeling with Gaussian processes. *Communications in Statistics - Simulation and Computation*, 29(4):1089–1108. (page 89)

Mitchell, P. J. and Tjian, R. (1989). Transcriptional regulation in mammalian cells by sequence-specific DNA binding proteins. *Science*, 245(4916):371–378. (page 14)

Nachman, I., Regev, A., and Friedman, N. (2004). Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, 20(1). (page 30)

Nardo, G., Iennaco, R., Fusi, N., Heath, P. R., Marino, M., Trolese, M. C., Ferraiuolo, L., Lawrence, N., Shaw, P. J., and Bendotti, C. (2013). Transcriptomic indices of fast and slow disease progression in two mouse models of amyotrophic lateral sclerosis. *Brain A journal of Neurology*, 136(11). (page 97)

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics No. 118, New York: Springer-Verlag. (page 53)

Ng, S. K., McLachlan, G. J., Wang, K., Jones, L. B.-T., and Ng, S. W. (2006). A mixture model with random-effects components for clustering correlated gene-expression profiles. *Bioinformatics*, 22(14):1745–1752. (page 89)

Nickisch, H. and Rasmussen, C. E. (2008). Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9(1). (page 55)

Nikolov, D. B. and Burley, S. K. (1997). Rna polymerase II transcription initiation: A structural view. *Proceedings of the National Academy of Sciences*, 94(1):15–22. (page 14)

O'Hagan, A. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society B*, 40(1):1–42. (page 53)

Ong, I. M., Glasner, J. D., and Page, D. (2002). Modelling regulatory pathways in e. coli from time series expression profiles. *Oxford University Press*, 18(1):S241–S248. (page 9)

Palikaras, K. and Tavernarakis, N. (2013). *Caenorhabditis elegans* (Nematode). (1):404–408. (page 12)

Papoulis, A. (1991). *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Companies, 3rd edition. (page 53)

Pearson, R. D., Liu, X., Sanguinetti, G., Milo, M., Lawrence, N. D., and Rattray, M. (2009). *puma*: a bioconductor package for propagating uncertainty in microarray analysis. *BMC Bioinformatics*, 10(211). (pages 35 and 97)

Pe'er, D. (2003). *From gene expression to molecular pathways*. PhD thesis, The Hebrew University. (page 47)

Peviani, M., Caron, I., Pizzasegola, C., Gensano, F., Tortarolo, M., and Bendotti, C. (2010). Unraveling the complexity of amyotrophic lateral sclerosis: recent advances from the transgenic mutant SOD1 mice. *CNS & Neurological Disorders - Drug Targets*, 9(4):491–503. (pages 14 and 88)

Phillips, T. and Hoopes, L. (2008). Transcription factors and transcriptional control in eukaryotic cells. *Nature Education*, 1(1):119. (page 24)

Pizzasegola, C., Caron, I., Daleno, C., Ronchi, A., Minoia, C., Carrı, M. T., and Bendotti, C. (2009). Treatment with lithium carbonate does not improve disease progression in two different strains of SOD1 mutant mice. *Amyotrophic Lateral Sclerosis*, 10(4):221–228. (pages 14 and 88)

Ptashne, M. and Gann, A. (1997). Transcriptional activation by recruitment. *Nature*, 386:569–577. (page 14)

Rahman, M. A. and Lawrence, N. D. (2014). Inferring transcription factor activities by combining binding information with gene expression profiles. http://nbviewer.jupyter.org/github/SheffieldML/notebook/blob/master/compbio/transcriptionFactorActivities.ipynb. (page 83)

Rahman, M. A. and Lawrence, N. D. (2016). A gaussian process model for inferring the dynamic transcription factor activity. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '16, pages 495–496, New York, NY, USA. ACM. (page 82)

Rajpaul, V., Aigrain, S., Osborne, M. A., Reece, S., and Roberts, S. J. (2015). A Gaussian process framework for modelling stellar activity signals in radial velocity data. *arXiv:1506.07304*. (page 1)

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. (pages 53, 55, 57, 58, and 64)

Ren, B., Robert, F., Wyrick, J. J., Aparicio, O., Jennings, E. G., Simon, I., Zeitlinger, J., Schreiber, J., Hannett, N., Kanin, E., Volkert, T. L., Wilson, C. J., Bell, S. P., and Young, R. A. (2000). Genome-wide location and function of DNA binding proteins. *Science*, 290:2306–2309. (pages 7 and 25)

Rényi, A. (1960). On measures of information and entropy. *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, page 547–561. (page 121)

Roeder, R. G. (1996). The role of general initiation factors in transcription by RNA polymerase II. *Trends Biochem. Sci.*, 21(9):327–335. (page 14)

Rogers, S. and Girolami, M. (2011). *A First Course in Machine Learning*. Chapman & Hall/CRC, 1st edition. (page 54)

Roweis, S. (1998). EM algorithms for PCA and SPCA. In *in Advances in Neural Information Processing Systems*, pages 626–632. MIT Press. (page 27)

Rowley, J. (2007). The wisdom hierarchy: Representations of the DIKW hierarchy. *Journal of Information Science*, 33(2):163–180. (page 23)

Saccon, R. A., Bunton-Stasyshyn, R. K. A., Fisher, E. M., and Fratta, P. (2013). Is SOD1 loss of function involved in amyotrophic lateral sclerosis? *Brain A journal of Neurology*, 136(pt 8):2342–58. (page 88)

Sanguinetti, G., Rattray, M., and Lawrence, N. D. (2006). A probabilistic dynamical model for quantitative inference of the regulatory mechanism of transcription. *Bioinformatics, Oxford University Press*, 22(14):1753–1759. (pages 10, 18, 31, 32, 34, 49, 71, 75, 80, 89, and 120)

Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–70. (page 24)

Shaye, D. D. and Greenwald, I. (2011). Ortholist: A compendium of C. elegans genes with human orthologs. *PLoS ONE*, 9(1). (pages 12 and 49)

Silva, R., Glymour, C., and Spirtes, P. (2005). Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7:2006. (page 27)

Simon, I., Barnett, J., Hannett, N., Harbison, C. T., Rinaldi, N. J., Volkert, T. L., Wyrick, J. J., Zeitlinger, J., Gifford, D. K., Jaakkola, T. S., and Young, R. A. (2001). Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, 106(6):697 – 708. (page 25)

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297. (pages 34, 75, 83, and 89)

Stegle, O., Lippert, C., Mooij, J. M., Lawrence, N. D., and Borgwardt, K. M. (2011). Efficient inference in matrix-variate Gaussian models with iid observation noise. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 24*, pages 630–638. Curran Associates, Inc. (page 79)

Sulston, J. E. and Horvitz, H. (1977). Post-embryonic cell lineage of the nematode, Caenorhabditis elegans. *Developmental Biology*, 56(1):110–156. (page 12)

Sulston, J. E., Schierenberg, E., j. G. White, and Thomson, J. N. (1980). The embryonic cell lineage of the nematode Caenorhabditis elegans. *Developmental Biology*, 100(1):64–119. (page 12)

Takahashi, K. and Yamanaka, S. (2006). Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. *Cell*, 126(4):663 – 676. (page 24)

The GPy Authors (2012–2014). GPy: A Gaussian process framework in python. http://github.com/SheffieldML/GPy. (pages 64, 74, 87, and 93)

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: series B*, 61(3):611–622. Published on behalf of the Royal Statistical Society. (page 27)

Tomancak, P., Beaton, A., Weiszmann, R., abd ShengQiang Shu, E. K., Lewis, S. E., Richards, S., Ashburner, M., Hartenstein, V., Celniker, S. E., and Rubin, G. M. (2002). Systematic determination of patterns of gene expression during drosophila embryogenesis. *Genome Biology*, 3(12):research 0088.1 – 0088.14. (page 89)

Topa, H., Jónás, Á., Kofler, R., Kosiol, C., and Honkela, A. (2015). Gaussian process test for high-throughput sequencing time series: application to experimental evolution. *Bioinformatics*. (page 1)

Turcotte, B. and Guarente, L. (1992). Hap1 positive control mutants specific for one of two binding sites. *Gene and Development*, 6:2001–2009. (page 26)

Turner, B. J. and Talbot, K. (2008). Transgenics, toxicity and therapeutics in rodent models of mutant SOD1-mediated familial ALS. *Progress in Neurobiology*, 85(1):94–134. (page 88)

Tzamarias, D. and Struhl, K. (1994). Functional dissection of the yeast cyc8-tup1 transcriptional co-repressor complex. *Nature*, 369(6483). (page 25)

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Phys. Rev.*, 36:823–841. (pages 61, 70, and 72)

Wackernagel, H. (2003). *Multivariate Geostatistics An Introduction with Applications*. Springer Science and Business Media. (pages 76, 92, and 94)

Watson, J. D. and Crick, F. H. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, (171):737–738. (page 2)

Wiener, N. (1923). Differential space. *Journal of Mathematical Physics*, pages 131–174. (page 70)

Wiener, N. (1949). *Extrapolation, Interpolation and Smoothing of Stationary Time Series.* (page 53)

Wienert, S., Heim, D., Saeger, K., Stenzinger, A., Beil, M., Hufnagl, P., Dietel, M., Denkert, C., and Klauschen, F. (2012). Detection and segmentation of cell nuclei in virtual microscopy images: A minimum-model approach. *Scientific Reports*, (503). (page 24)

Williams, C. K. I. and Barber, D. (1998). Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1342–1351. (page 55)

Williams, C. K. I. and Rasmussen, C. E. (1996). Gaussian processes for regression. *Advances in Neural Information Processing Systems 8, MIT Press*, pages 1–7. (page 53)

Wood, W. B. (1988). The nematode C. elegans. *Cold Spring Harbor Laboratory Press, New York*, pages 1–16. (page 12)

WormNet (2015). Wormnet. http://www.functionalnet.org/wormnet/about.html [Online; accessed 01-Sept-2016]. (pages 38, 40, and 42)

Xie, X., Lu, J., Kulbokas, E. J., Golub, T. R., Mootha, V., Lindblad-Toh, K., Lander, E. S., and Kellis, M. (2005). Systematic discovery of regulatory motifs in human promoters and 3' utrs by comparison of several mammals. *Nature*, (434):338–345. (page 38)

Appendix A

Mathematical Background

A.1 Gaussian Identities

This appendix aims to make the thesis self contained with a very short reference to the basic mathematical identities we used in this document.

A.1.1 Gaussian Density

Perhaps Gaussian density is the most common probability density, given by

$$p\left(x|\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left(x-\mu\right)^2}{2\sigma^2}\right) \triangleq \mathcal{N}\left(x|\mu,\sigma^2\right) \tag{A.1}$$

where μ denotes the mean and σ^2 is the variance of the density.

A.1.2 Multivariate Gaussian

Let's $\mathbf{x}$ is an d-dimensional multivariate normal random variable, then the probability function is given by

$$p\left(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}\right) \triangleq \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}\right) = \left(2\pi\right)^{-d/2}\left|\boldsymbol{\Sigma}\right|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}\right)^{\top}\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)\right) \tag{A.2}$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ denotes the mean and $\boldsymbol{\Sigma}$ is a symmetric, positive covariance matrix with $[d \times d]$ dimension.

A.1.3 Sum of two Gaussians

Sum of Gaussian variables is also Gaussian. Let's

$$x_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right) \tag{A.3}$$

then the sum is distributed as

$$\sum_{i=1}^{n} x_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right) \tag{A.4}$$

According to central limit theorem, as sum increases, sum of non-Gaussian with finite variance variables is also Gaussian.

A.1.4 Scaling a Gaussians

Scaling a Gaussian variable is also Gaussian. Let's

$$x \sim \mathcal{N}\left(\mu, \sigma^2\right) \tag{A.5}$$

then the scaled density is distributed as

$$wx \sim \mathcal{N}\left(w\mu, w^2\sigma^2\right) \tag{A.6}$$

which leads to product of Gaussians. The product of two Gaussian is also Gaussian.

A.1.5 Product of two Multivariate Gaussians

Let's, $\mathbf{x}$, $\boldsymbol{\mu}_a$ and $\boldsymbol{\mu}_b$ be the size of $[d \times 1]$ and $\boldsymbol{\Sigma}_a$ and $\boldsymbol{\Sigma}_b$ be $[d \times d]$ covariance matrices. The product of two multivariate Gaussian distributions is proportional to another multivariate Gaussian distribution given by

$$\mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a\right)\mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\right) = Z\mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right) \tag{A.7}$$

where the covariance is

$$\boldsymbol{\Sigma}_c = \left(\boldsymbol{\Sigma}_a^{-1} + \boldsymbol{\Sigma}_b^{-1}\right)^{-1} \tag{A.8}$$

and mean is

$$\boldsymbol{\mu}_c = \boldsymbol{\Sigma}_c\left(\boldsymbol{\Sigma}_a^{-1}\boldsymbol{\mu}_a + \boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right). \tag{A.9}$$

The normalising constant Z is Gaussian in either $\boldsymbol{\mu}_a$ or $\boldsymbol{\mu}_b$

$$z_c = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_a \boldsymbol{\Sigma}_b \boldsymbol{\Sigma}_c^{-1}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{\mu}_a^\top \boldsymbol{\Sigma}_a^{-1} \boldsymbol{\mu}_a + \boldsymbol{\mu}_b^\top \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\mu}_b - \boldsymbol{\mu}_c^\top \boldsymbol{\Sigma}_c^{-1} \boldsymbol{\mu}_c\right)\right). \quad \text{(A.10)}$$

Let's $\mathbf{y}$ is a $[d' \times 1]$ Gaussian random variable whose mean depends linearly depends on $\mathbf{x}$ where $\boldsymbol{\Sigma}_d$ has the dimension $[d' \times d]$, and $\boldsymbol{\Sigma}_b$ has the dimension $[d' \times d']$. The product of two Gaussian is given by

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a)\mathcal{N}(\mathbf{y}|\boldsymbol{\Sigma}_d \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b) \propto \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \quad \text{(A.11)}$$

The product is proportional to a multivariate normal density with mean

$$\boldsymbol{\mu}_c = \boldsymbol{\Sigma}_c\left(\boldsymbol{\Sigma}_a^{-1}\boldsymbol{\mu}_a + \boldsymbol{\Sigma}_d^\top \boldsymbol{\Sigma}_b^{-1}\mathbf{y}\right). \quad \text{(A.12)}$$

and the covariance is

$$\boldsymbol{\Sigma}_c = \left(\boldsymbol{\Sigma}_a^{-1} + \boldsymbol{\Sigma}_d^\top \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\Sigma}_d\right)^{-1}. \quad \text{(A.13)}$$

A.1.6 Conditional and Marginal Distributions

Let's $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian, partitioned into $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]^\top$ such that

$$p\left(\begin{bmatrix}\mathbf{x}_1 \\ \mathbf{x}_2\end{bmatrix}\right) \propto \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_1 \\ \mathbf{x}_2\end{bmatrix} \middle| \begin{bmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}\right) \quad \text{(A.14)}$$

The marginal distributions are

$$p(\mathbf{x}_2) \sim \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \quad \text{(A.15)}$$

and

$$p(\mathbf{x}_2) \sim \mathcal{N}(\mathbf{x}_2|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \quad \text{(A.16)}$$

The conditional distributions are

$$p(\mathbf{x}_1|\mathbf{x}_2) \sim \mathcal{N}\left(\mathbf{x}_1|\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\right) \quad \text{(A.17)}$$

and

$$p(\mathbf{x}_2|\mathbf{x}_1) \sim \mathcal{N}\left(\mathbf{x}_2|\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1), \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\right). \quad \text{(A.18)}$$

A.1.7 Linear Forms

Let's $p(\mathbf{x}) \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{c}$ then $p(\mathbf{y}) \sim \mathcal{N}\left(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{c}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^{\top}\right)$.

A.1.8 Gaussian Integrals

The probability density function integrates to one (by definition), given by

$$\int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, \mathrm{d}\mathbf{x} = 1 \tag{A.19}$$

A.2 Matrix Analysis

Let's $\mathbf{P}$ and $\mathbf{Q}$ be non-singular matrices with $[d \times d]$ dimensions. The inverse of the product of two matrices can be written in terms of the individual inverses

$$(\mathbf{P}\mathbf{Q})^{-1} = \mathbf{Q}^{-1}\mathbf{P}^{-1}. \tag{A.20}$$

and the product with a scaler c is

$$(c\mathbf{P})^{-1} = c^{-1}\mathbf{P}^{-1}. \tag{A.21}$$

The sum of two matrices with inverses are valid for following identity

$$\mathbf{P}^{-1} + \mathbf{Q}^{-1} = \mathbf{P}^{-1}(\mathbf{P} + \mathbf{Q})\mathbf{Q}^{-1}. \tag{A.22}$$

and also for

$$\left(\mathbf{P}^{-1} + \mathbf{Q}^{-1}\right)^{-1} = \mathbf{P}(\mathbf{P} + \mathbf{Q})^{-1}\mathbf{Q} = \mathbf{Q}(\mathbf{P} + \mathbf{Q})^{-1}\mathbf{P}. \tag{A.23}$$

The Woodbury formula, or the matrix inversion lemma or the Sherman–Morrison–Woodbury formula is given by

$$\left(\mathbf{P} + \mathbf{Q}\mathbf{R}^{-1}\mathbf{S}\right)^{-1} = \mathbf{P}^{-1} - \mathbf{P}^{-1}\mathbf{R}\left(\mathbf{Q} + \mathbf{S}\mathbf{P}^{-1}\mathbf{R}\right)^{-1}\mathbf{S}\mathbf{P}^{-1}. \tag{A.24}$$

A.3 Singular Value Decomposition

The singular value decomposition is a matrix factorization technique of a real or complex matrix. Let's the matrix $\mathbf{S}$ is an $|m \times n|$ real matrix with $m > n$, then by

singular value decomposition $\mathbf{S}$ can be written as

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top \tag{A.25}$$

where $\mathbf{U}$ is an orthonormal matrix (i.e. $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$) with $|m \times n|$ dimensions, $\mathbf{\Lambda}$ is the diagonal matrix containing the eigenvalues of $\mathbf{S}$ with $|n \times n|$ dimensions and $\mathbf{V}$ is another orthonormal matrix (i.e. $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$) with $|n \times n|$ dimensions. When applied to a positive semi-definite matrix, the singular value decomposition is equivalent to the *eigendecomposition*.

A.4 Markov Property

Let's $\{X(t), t \geq 0\}$ be a stochastic time continuous process with non-negative integer values. This process is termed as a discrete Markov process if for every $n \geq 0$, time points $0 \leq t_0 < t_1 < \cdots < t_n < t_{n+1}$ and states $i_0, i_1, ..., i_{n+1}$ it holds that

$$\begin{aligned} &p(X(t_{n+1}) = i_{n+1} \mid X(t_n) = i_n, X(t_{n-1}) = i_{n-1}, ..., X(t_0) = i_0 \\ &= p(X(t_{n+1}) = i_{n+1} \mid X(t_n = i_n)). \end{aligned} \tag{A.26}$$

This definition states that any information of the future behaviour of the process solely depends on the present state. Adding the history of the process does not increase or update any new information.

A.5 Cholesky Decompositions

Inversion of a symmetric positive definite matrix is a very common requirement while working with Gaussian processes and their approximations. Let's $\mathbf{\Sigma}$ the symmetric positive definite covariance matrix. In practice we rarely need the $\mathbf{\Sigma}^{-1}$ itself. For a given vector $\mathbf{y}$ common forms we require are $|\mathbf{\Sigma}|$, $\mathbf{y}^\top\mathbf{\Sigma}^{-1}\mathbf{y}$, and $\mathbf{\Sigma}^{-1}\mathbf{y}$. The most efficient and computationally stable way to obtain these forms is via Cholesky decomposition (also known as the matrix square root)

$$\mathbf{\Sigma} = \mathbf{L}\mathbf{L}^\top \tag{A.27}$$

where $\mathbf{L}$ is a triangular matrix known as Cholesky factor. Though Cholesky factor has the same order cost as matrix inversion $\mathcal{O}(N^3)$, it is cheaper in terms of constant

factors.

$$\begin{aligned}\mathbf{y}^\top \boldsymbol{\Sigma}^{-1}\mathbf{y} &= \mathbf{y}^\top \left(\mathbf{L}\mathbf{L}^\top\right)^{-1}\mathbf{y} \\ &= \mathbf{y}^\top \mathbf{L}^{-\top}\mathbf{L}^{-1}\mathbf{y} \\ &= \| \mathbf{L}^{-1}\mathbf{y} \|^2\end{aligned} \tag{A.28}$$

The vector $\mathbf{L}^{-1}\mathbf{y}$ can be computed with the computational cost $\mathcal{O}(N^2)$ by forward substitution as $\mathbf{L}$ is triangular. $\boldsymbol{\Sigma}^{-1}\mathbf{y}$ can be computed using back substitution $\mathbf{L}^{-\top}\mathbf{L}^{-1}\mathbf{y}$ and $|\boldsymbol{\Sigma}|$ is computed as $|\boldsymbol{\Sigma}| = \prod_n \mathbf{L}_n^2 n$.