# Gaussian Process in Computational Biology: Covariance Functions for TFAs and Gene Expression Clustering


**Muhammad Arifur Rahman**

Department of Computer Science

University of Sheffield

This dissertation is submitted for the degree of

*Doctor of Philosophy*

September 2015

# Abstract

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. But quantitative estimation is complex due to some reasons. Many of the transcription factors' activity and their own transcription level are post transcriptionally modified; very often the levels of the transcription factors' expressions are low and also contain noise. So, from the expression levels of their target genes it is useful to infer the activity of the transcription factors. Here we develop a Gaussian process based regression to infer the exact TFAs from a combination of mRNA expression level and DNA protein binding measurement.

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 background.

In this paper we develop a new clustering method that allows each cluster to be parameterised according to whether the behaviour of the genes across conditions is correlated or anti-correlated. By specifying correlation between such genes we gain more information within the cluster about how the genes interrelate.

Amyotrophic lateral sclerosis (ALS) is an irreversible neurodegenerative disorder that kills the motor neurons and results in death within 2 to 3 years from the symptom onset. Speed of progression for different patients are heterogeneous with significant variability. It is already reported that $SOD1^{G93A}$ transgenic mice from different backgrounds ($129Sv$ and $C57$) showed consistent phenotypic differences for disease progression. Here we used a hierarchy of Gaussian processes to model condition-specific and gene-specific temporal covariances. This study demonstrated about finding some significant gene expression profiles and clusters of associated or co-regulated gene expressions together from four groups of data ($SOD1^{G93A}$ and $Ntg$ from $129Sv$ and $C57$ backgrounds). Further gene enrichment score analysis and ontology pathway

analysis of some specified clusters for a particular group may lead toward identifying features underlying the differential speed of disease progression. Our study shows the effectiveness of sharing information between replicates and different model conditions when modelling gene expression time series.

# Table of contents

# List of figures

# List of tables

# Chapter 1

# Introduction

## 1.1 System Biology

The prime goal of Biology is to get the 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 radically changed the way of study and development of biology and biological systems. They explained the biological phenomena with the help of molecular basis. This new concept helps to explain different aspects of biology like heredity, different diseases, various evolutionary aspects as well as development with more firm theoretical ground. Since then, biology became a framework of knowledge governed by some basic and fundamental laws of physics.

Due to the enormous advancement of molecular biology, at present we have in-depth knowledge of elementary processes like evolution, heredity, disease, development etc. These mechanisms also includes other biological features like replication, transcription and translation. Accomplishment of symbolic DNA sequencing helped to reveal large number 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 profile are available at the mRNA level. Even measurement of protein level and their different subsequent actions are also making progress.

Undoubtedly understanding at the molecular level will accelerate to understand the biological systems but these knowledge aren't sufficient to understand biological systems as systems. Genes and protein are few components of a whole system. It is necessary to understand what constitutes the system, but even only this knowledge is not sufficient to understand the complete system. System biology is a new field of

biology able to acquire understanding up to system level of 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 disciplines research area like molecular biology, high-precision measurement technology, mathematics, computer science, control theory and other engineering and scientific field. Kitano (2002) mentioned the main four key areas to be carried out for further research: (1) genomic and other molecular biology research, (2) various 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 clearly depicts requirement of multi-disciplinary research effort to get the knowledge of biological systems as systems. Indeed the abstract concept of system is more than a collection of multi-disciplinary research components. It is also essential to know what happens during the period or processes when any stimuli and/or disruptions take place to obtain the proper insight of system beside the detailed description of the components.

Identification of the system structure is the primary requirement to understand biological systems. Some of the key structures might have different regulatory relationship of genes and interactions with proteins that show the metabolism pathway and signal transduction, physical structure of chromatin, cells, organisms and other components. Though it is very critical to monitor biological processes in bulk using high-throughput DNA micro-array, real-time polymerase chain reaction (RT-PCR), protein chips and other methods thereafter methods to identify genes and metabolism network have to be established. Once a system structure is established up to a certain degree, it is required to find out the behaviour. To understand the behaviour properly a number of analysis methods can be used. For example, consider that we want to know the sensitivity of a specified behaviour against some external perturbations, and its time to return its normal state since the stimuli take place. This type of analysis provides the system level characteristics as well as uncover important insights of medical treatments by revealing cell responses to certain chemical affinities.

To apply the knowledge obtained from system structure and behaviour understanding, further research is required to control the state of the biological systems. All these phases lead toward the establishment of technologies that allow us to design biological system which can provide cures for different diseases. Some futuristic example could be organ cloning technique for the treatment of diseases that require organ transplants or

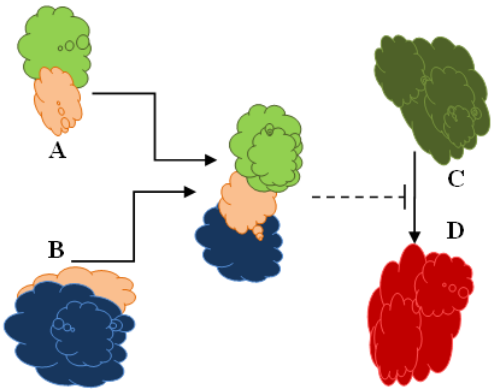


Fig. 1.1 A 'cartoon' model of protein protein interaction. Two different molecular species A and B bind to form a complex molecular. The newly formed complex hinder the rate at which molecules of species C are transformed to species D.

building biological materials for engineering, specially robotics, having self-sustaining and self-repairing capabilities.

## 1.2 Dynamic Mathematical model: what and why in System biology?

Any models are abstractions of reality. Models mostly designed to focus on specific aspects of the objects for a certain kind of study. Usually other aspects are abstracted away. Biologists are almost regularly making use of tangible 'real world'models. Some of them are very simple, like molecular ball-and-stick, again some of them are 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 communicate by diagrams. These diagrams are usually constructed with a set of components and the ways they interact with each other. While representing knowledge of cellular or different other processes, these interaction diagrams, or 'cartoon'models may play a central role (Ingalls (2012)).

A major drawback of these cartoon models is that, while considering system behaviour, they could be significantly ambiguous. It is more, if there is any interaction network related with feedback. Complexity increases even further when the number of components and their corresponding interactions in the network grow. Sometimes it becomes very difficult to get the intuitive understanding of the system behaviour. But a

mathematical model or description of the same model can eliminate uncertainty of the model behaviour. The mathematical model will consider the quantitative representation of 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 to 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 interaction. But a well studied mechanisms with sufficient data might be capable to show 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 systems 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 part 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 are also hypothesis. Though the real cellular behaviour cannot be predicted by simulation, but 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 about 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 result such as bifurcation diagram or phase portfolio. The structure of the system should include structure of gene 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. Beside 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

In modern molecular biology the biological systems like cells are treated as a 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. But the actual behaviour of biological systems are 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. But to explain the biological systems' behaviour only these are not 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 to reach 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 Microarray and Gene Expression Data for Genomics

Living cells contain thousands of genes. These genes code for one or more proteins. Expression of these genes are regulated by many of these proteins through a very complex regulatory pathway. Usually regulation occurs to accommodate the change of the environment, as well as at the cell cycle of the development process. Gene expression is the process where information contained in the gene is used to synthesise 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 DNA microarray or DNA chip which is also known as biochip.

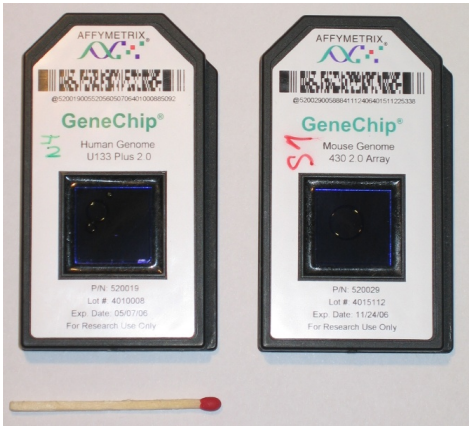


Fig. 1.2 Gene Expression Data: Affymetrix Micro Array (Image courtesy Wikipedia. http://en.wikipedia.org/wiki/DNA_microarray)

Figure 1.2 shows two Affymetrix chips which contain DNA microarray. A match is shown at the bottom for the purpose of size reference of a microarray. The solid-phase DNA macroarray 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 locations. Every strand constructs with 25 probes or bases. The microarray is rinsed and washed with fluorescent stain. To accomplish a DNA test, two types of samples are used: one is the test sample and the other one is controlled sample. After extracting mRNA from DNA, copies are made

from mRNA by reverse transcription. Two different fluorescent tagged with cyanide are used to differentiate between control sample and 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 hybridisation process a green spot represents a hybridisation with the control targets only, a red spot indicates hybridisation with the test targets only, yellow represent hybridisation both with the control targets and test targets, while black represents hybridisation with the neither samples, i.e. no hybridisation. Over the last couple of decades these gene expression data became one of the key resource 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.

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 using a Dynamic Bayesian network and trying to gain the insight of regulatory pathway. By analysing gene expression data Friedman et al. (2000) were the first to determine the properties of transcriptional program for Baker's yeast using a Bayesian network.

Many of the recent studies already established the fact that the gene function of 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 to quantitative measure and analyse the mechanisms regulating 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 of the mRNA expression levels of transcription factors (Sanguinetti et al. (2006)).

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. Now biologists can acquire 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

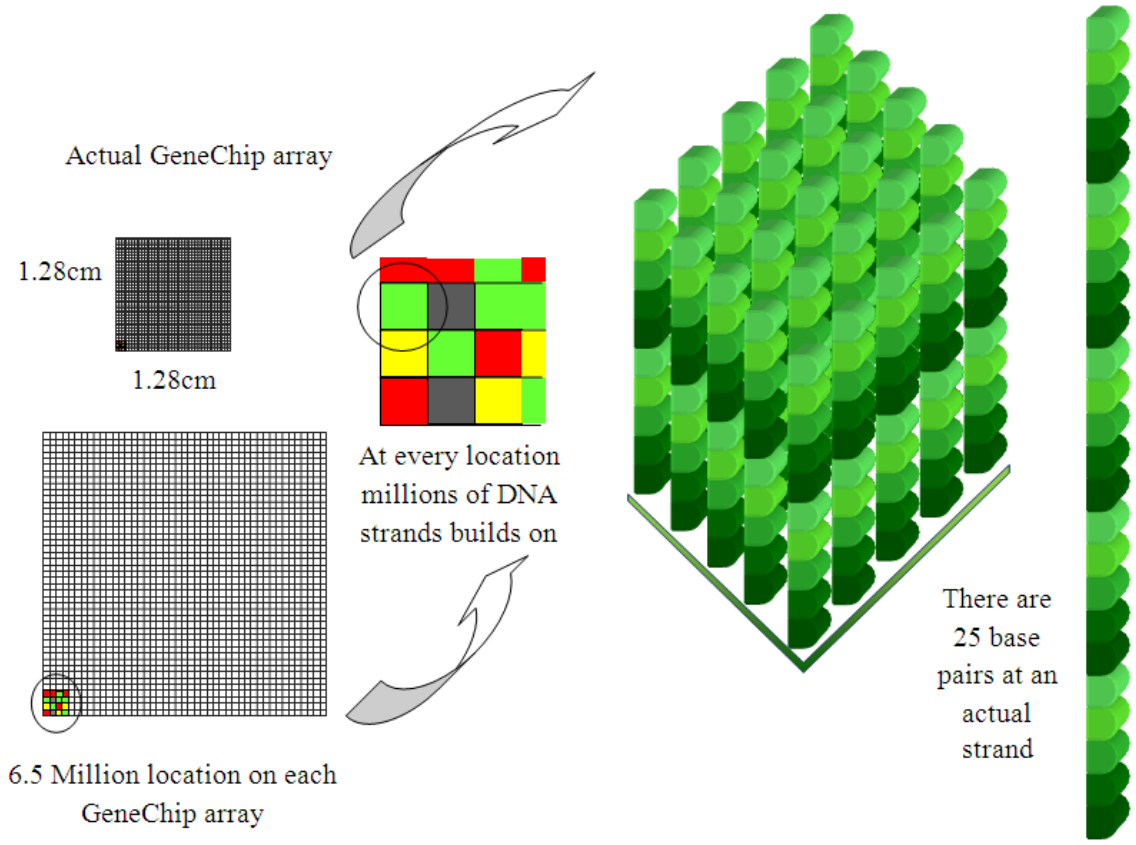


Fig. 1.3 Gene Expression Data: Affymetrix Micro Array

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.

### 1.4.2 *Caenorhabditis elegans*

*Caenorhabditis elegans* is a nonparasitic, soil dwelling, small nematode worm. *C. elegans* and other *Caenorhabditis* species are found through all over the world. It can easily colonize mostly in the rotting materials with other microorganism. *C. elegans* is easy to maintain in the petri dishes at the laboratory. At 25 °C *C. elegans* completeits life cycle in just 2.5 days from fertilized embryos to egg-laying adult through 4 larval stages. Its typical life span 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)).

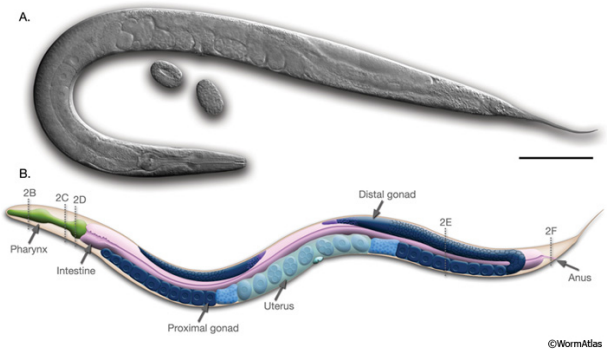


Fig. 1.4 Anatomy of an adult hermaphrodite(*C.elegans*). A. DIC image of an adult hermaphrodite, left lateral side. Scale bar 0.1 mm. B. Schematic drawing of anatomical structures, left lateral side (Courtesy of WormAtlas http://www.wormatlas.org/hermaphrodite/introduction/IMAGES/introfig1leg.htm).

*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 was facilitated by its simple body plan. It is an eukaryote and it shares cellular and molecular structures and control pathways with higher organisms. *C. elegans* is multicellular, an adult wild type consist of 959 somatic cells and among these 302 are neurons (Palikaras and Tavernarakis (2013); Sulston and Horvitz (1977)). It's developmental process (e.g. embryogenesis, morphogenesis) goes through a complex process to develop into an adult. Yet 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 disease *C. elegans* used as a vital model. There are between $\sim 20,250$ to $\sim 21,700$ 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 $\sim 38\%$ of the 20,250 protein-coding genes predicted in *C. elegans*. When human genes introduced into *C. elegans*, human genes replaced their homologs. On the contrary, many *C. elegans* genes can function with great deal of similarity to human like mammalian genes. So, the

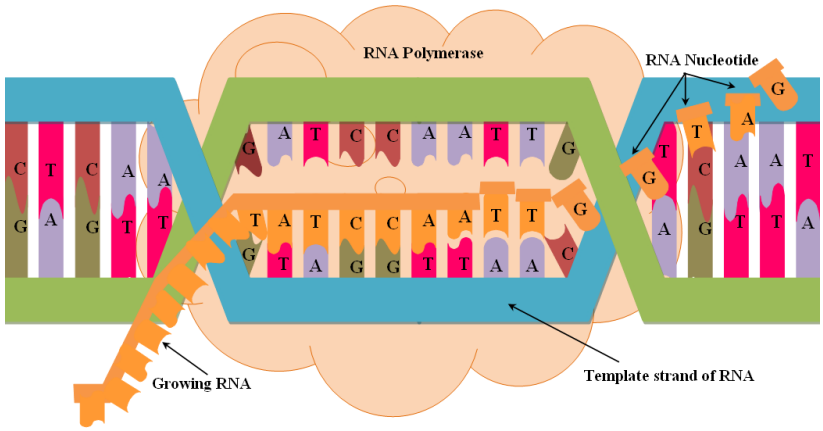


Fig. 1.5 A 'cartoon model'of DNA Transcription

biological insight acquired from *C. elegans* may be directly applicable to more complex organism like human one.

### 1.4.3 Transcription

All the processes in the body take places through some proteins. So, cells need protein continuously. As a consequence, proteins are required to be manufactured at every moment in the body. Inside the cell, proteins are manufactured from the DNA. When any cell in the body is in need of a protein, a special signal is sent to the DNA using hormones. Then proteins reside in DNA start to manufacture depending on the received codes. The way that the enzymes find the information required for protein construction is extremely complicated.

DNA (Deoxyribonucleic acid) transcription is a process that involves the transcribing of genetic information from DNA to a complementary RNA (Ribonucleic acid). Protein is produced from a copy of DNA by the transcription process. This production of proteins and enzymes are controlled by the coding of cellular activity. Even the conversion of DNA to proteins is not straight forward. A RNA polymerase reads the sequence of DNA, which produces a complementary RNA. 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 to the process of DNA transcription are :

**RNA polymerase binding to DNA:** In order to initiate the DNA transcription, RNA polymerase and sigma factor form a holoenzyme. Transcription process starts at

the promoter region of a double-stranded DNA. Sigma factor can recognize the DNA and its 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 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 opened, transcribed and reclosed with minimum stress on the DNA molecule. At any certain time, many RNA polymerase can transcribe a single DNA sequence, which can manufacture a large quantity of protein at once.

### 1.4.4 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 factors 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 proteins. 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)).

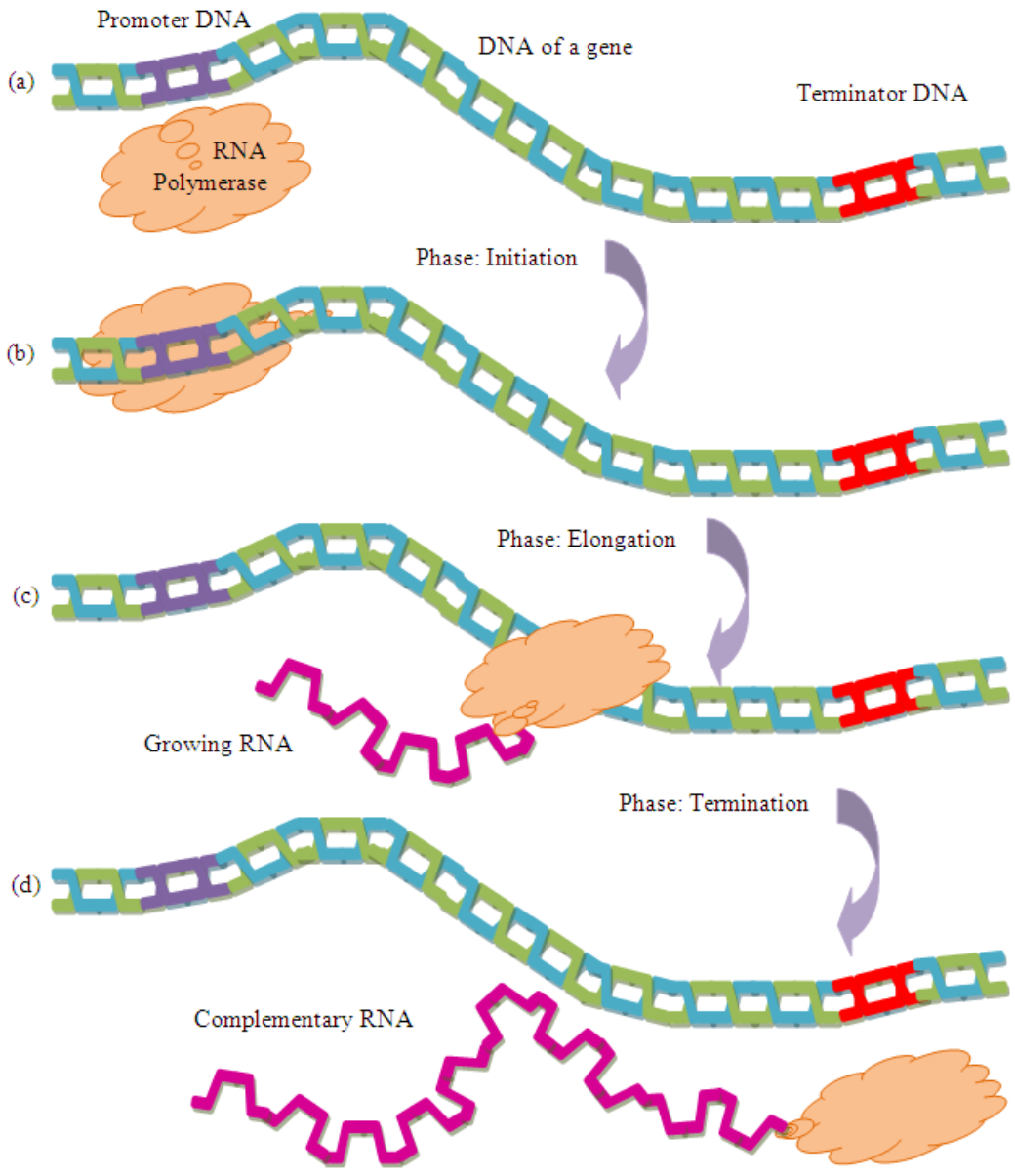


Fig. 1.6 A 'cartoon model'of Transcription Process: DNA Transcribed in mRNA

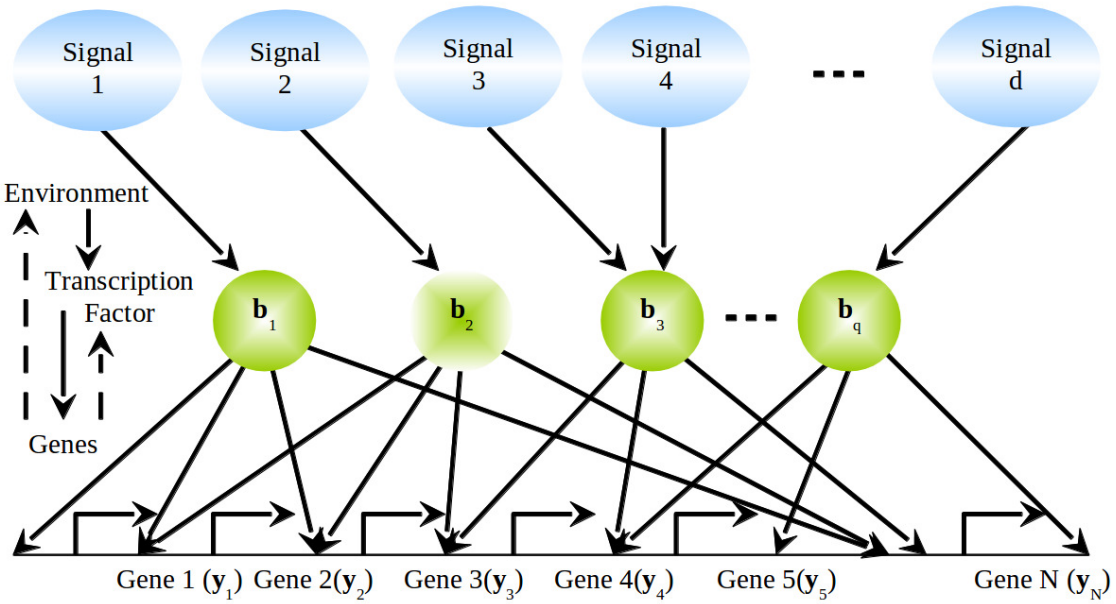


Fig. 1.7 The mapping between environmental signal, transcription factors and the genes that they regulate (Alon (2006)).

### 1.4.5 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 murinae models also show the heterogeneity in the disease progression for the clinical phenotype. In a previous 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 murinae models could be interesting to examine the the speed of propagation of ALS.

## 1.5 Gaussian Processes

Gaussian processes (GPs) are general class of models of functions. GPs are one of the most simple and widely used families of stochastic processes. As a general setting,

Gaussian process of many types have been studied and incorporated in research for decades, specially 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 Gaussian process at Chapter 3.

## 1.6 Main goal of the thesis

First we set some 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 different stages

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

## 1.7 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 bellow-

- Muhammad Arifur Rahman and Neil D. Lawrence, "A Gaussian Process Model for Inferring the Dynamic Transcription Factor Activity", *International Conference on Intelligent Systems for Molecular Biology and European Conference on Computational Biology*, Ireland, July 2015.

- Muhammad Arifur Rahman 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.

## 1.8 Road Map

The thesis is structured in the following chapters:

**Chapter 1**: This document start 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 basis concepts of 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) but 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 Gaussian process, regression problem and regression with Gaussian process. Choice of an appropriate kernel is one of the key issue while modelling with Gaussian process. In this chapter we briefly describe about 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 was justified at the later section of this chapter.

**Chapter 4**: We note that the linear Gaussian model is equivalent to a Gaussian process 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 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 neurons and results in death within 2 to 3 years from the symptom onset. 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 parametrised according to whether the behaviour of the genes across conditions are correlated or anti-correlated. By specifying 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 on 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.

## 1.9 Notation, Symbols and Acronyms

### 1.9.1 Acronyms

| | |
|---|---|
| **cDNA** | Complementary Deoxyribonucleic Acid |
| **C.elegans** | Caenorhabditis elegans |
| **ChIP** | Chromatin Immunoprecipitation |
| **DIC** | Differential Interference Contrast |
| **DNA** | Deoxyribonucleic Acid |
| **EDGEdb** | C. elegans Differential Gene Expression Database |
| **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 |
| **LVM** | Latent Variable Model |
| **mRNA** | messenger Transfer Ribonucleic Acid |
| **PCA** | Principal Component Analysis |
| **PLS** | Partial Least Square |
| **PPCA** | Probabilistic Principal Component Analysis |
| **RBF** | Radial Basis Function |
| **RNA** | Ribonucleic Acid |
| **RT-PCR** | Reverse Transcription Polymerase Chain Reaction |
| **SE** | Squared Exponential |
| **SVD** | Singular Value Decomposition |
| **TF** | Transcription Factor |
| **TFA** | Transcription Factor Activity |

### 1.9.2 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 |
| $\mathbf{x}\_\mathrm{i}$ | 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 |
| $\mathrm{diag}\left(\mathbf{x}^\top\right)$ | 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 |
| $\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ | A Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ |
| $\mathcal{N}\left(.\lvert\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ | The probability distribution of a Gaussian random variable 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 |

# Chapter 2

# Probabilistic TFA of *C. elegans*

The data – information – knowledge – wisdom (DIKW) hierarchy is one of the fundamental and widely recognized hierarchy in the information and knowledge literature. This hierarchy contextualizes data, information, knowledge and wisdom, with respect to one another to identify and describe the processes involved in the transformation of 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 the vital role behind 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 real world suffer from quality issues for various reasons. Acquisition errors are very likely to be included even in the controlled environment. Dealing with noise or added uncertainty of the data is troublesome. Within the constraints, probabilistic modelling is a dominant approach with added flexibility and capability to deal with uncertainty.

## 2.1 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 representation of unobservable factors/covariates, missing data, random effects, finite mixtures, variations in hierarchical data, clusters and many more. Figure **??** shows an analogy of latent variable model where marionette's different movement and dynamics are observed, whereas these

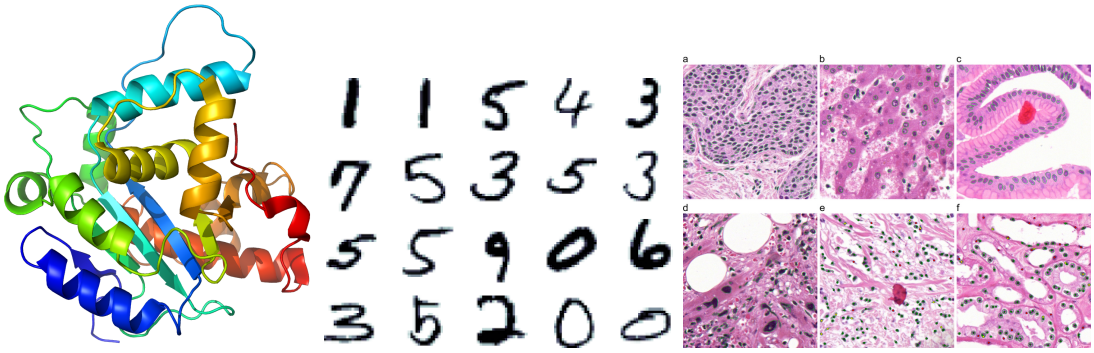

Fig. 2.1 Examples of high dimensional data form types and nature. Left: 3D model of a protein structure. Centre: Multiple samples of hand written digits from MNIST dataset. available at http://yann.lecun.com/exdb/mnist/. Right: Multiple image patches from breast cancer, liver, gastric mucosa, bone marrow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

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 $\epsilon_i$. Typically it is assumed that the noise has a Gaussian distribution $\epsilon_i \sim \mathcal{N}(0, \beta^{-1})$, where the term $\beta$ 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 the Equation 2.1 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 $\beta$ tends to infinity, PCA is recovered in the limit. The conditional

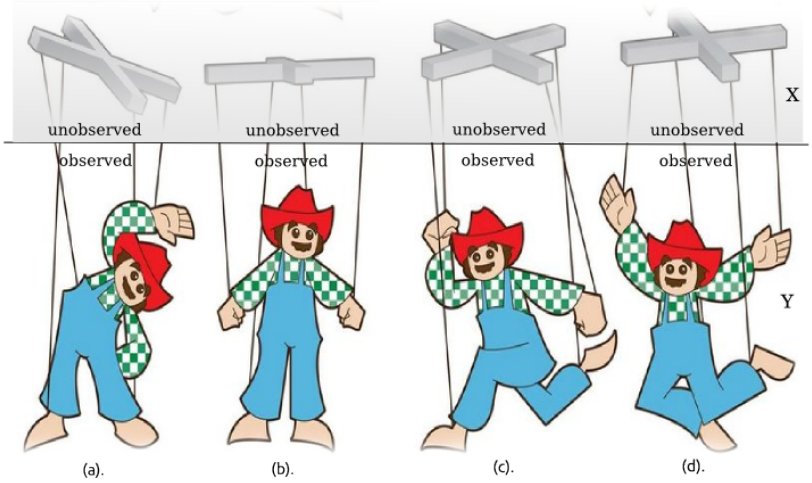


Fig. 2.2 Marionette analogy of latent variable model: Marionette's different dynamics are observed (represented by $\mathbf{Y}$). But these movements are controlled by the puppeteer's control bar which is unobserved (dynamics represented by $\mathbf{X}$).

probability of the data given the latent space is

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

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

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

where $N$ is the total number of data. Even for finite precision $\beta$ the maximum likelihood solution for $\mathbf{W}$ spans the principal sub-space of the data (Tipping and Bishop (1999)). This approach is can be applicable for both linear (e.g. Silva et al. (2005)) 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 maximum likelihood. Lawrence (2005) used an alternative but similar approach by first marginalising the parameters and then optimizing the latent variables.

## 2.2 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 of 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's $\mathbf{m}_1$ 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 Eq. 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 Eq. 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 Eq. 2.9 and Eq. 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.

## 2.3 Modelling of TFAs

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

In recent years an idea that has gain a lot of interest to infer regulatory mechanism from the expression level 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 are sum 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 have 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 (SVD) 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 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 above mentioned methods is that transcription factor activities do not hold any information regarding the strength of the regulators interactivity between the transcription factor 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 large portion of ChIP data suffers form 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) build 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.

## 2.4 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 $R$ 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 has 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.

## 2.5 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 his 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 $d$ the time points or number of experiments.

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

The connectivity measurements are collected in a binary matrix $\mathbf{X} \in \mathbb{R}^{N \times q}$, where $q$ is the number of transcription factors. We assume that $\mathbf{X}_{i,j}$ is one if transcription factor $j$ can bind gene $i$, zero otherwise.

Sanguinetti et al. (2006) used a latent variable model (as we described in Section: 2.1). TFAs were obtained by regressing 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.

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 is- $\mathbf{b}_{nt}$ has the Markov property (Explanation add in appendix and fix the ref...) and hence gene specific TFA $\mathbf{b}_{nt}$ at time $t$ depends solely on the gene specific TFA at time $(t-1)$.
- The second assumption is- 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 will be

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

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 will be 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 this 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 $\gamma$ 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 $\boldsymbol{\epsilon}_n \sim \mathcal{N}\left(0, \boldsymbol{\sigma}^2\mathbf{I}\right)$ we have

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

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

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

where

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

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

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

TFAs can be estimated a posteriori using Bayes's theorem,

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

where the posterior covariance is given by

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

where

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

and posterior mean is given by

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

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

## 2.6 Datasets

Sanguinetti et al. (2006) did their experiments on yeast's 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 an 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* used as a vital model. We provided introductory information about this model organism Chapter 1. To find out the TFAs of *C. elegans* we had to work with three type of datasets

1. Gene expression time series data
2. A list of transcription factors

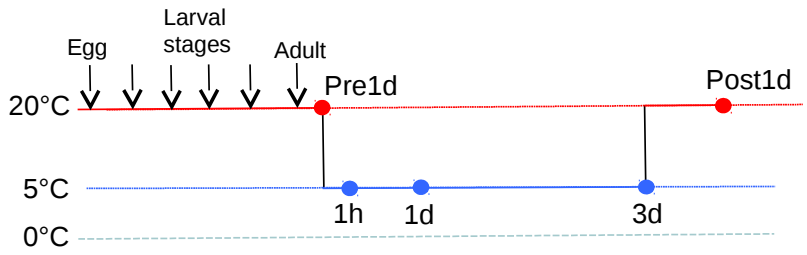


Fig. 2.3 Temperature and time settings for the gene response to chill exposure experiments: 'Pre1d '- The first experimental data within one day of *C. elegans*'s adulthood at the temperature 20 °C. '1h '- second experimental data were collected after one hour of the reduction of the temperature to 5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

3. Connectivity information between genes and transcription factors.

### 2.6.1 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. 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 were 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 were collected after one hour of the reduction of the temperature. Third experimental data were collected after 24

[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.

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

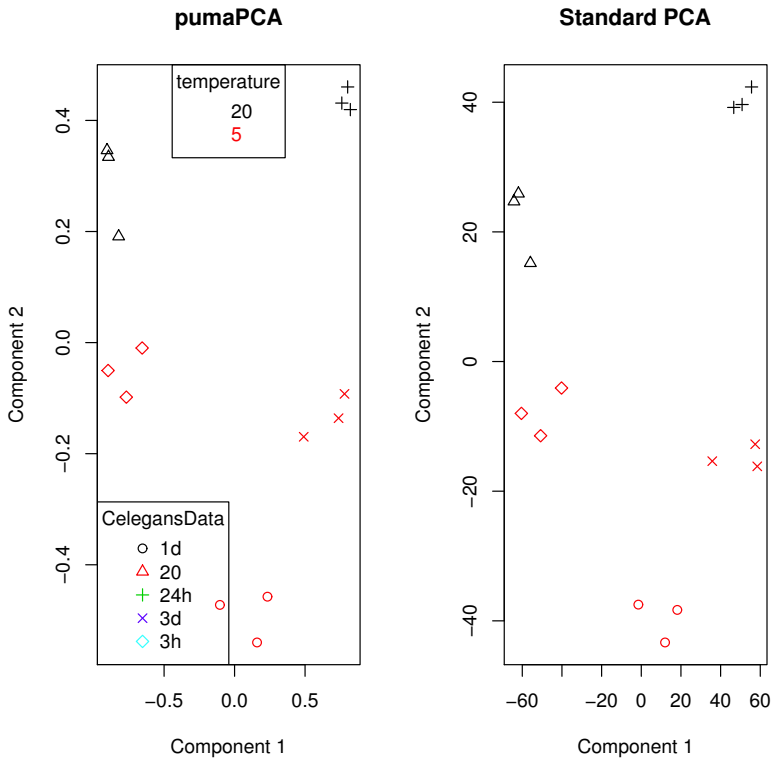


Fig. 2.4 Principal component analysis of gene expression time series data. Figure in the left shows the *puma* PCA which incorporates the uncertainty of the point expressions. Figure in the right shows the standard PCA which does not include any uncertainty of the point expressions. It appears from these 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 higher quality of the data

hours (1 day) of the temperature reduction. Fourth experimental data were 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 were collected after one day of the rise of the temperature. In the wet lab the full experiment were repeated twice maintaining the similar setup. So, three independent replicates of the experiments were available.

Figure 2.4 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 of figures have a very similar representations. 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.

### 2.6.2 Transcription Factors

From different data sources, we found different number/list of transcription factors for *C. elegans*. Inmaculada et al. (2007) build 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.

### 2.6.3 Connectivity Information

Network motifs are the simplest units of transcriptional regulatory network's architecture. Particular regulatory mechanism such as positive and/or negative feedback loop can be well studied by these network motifs. Based on size and nature network motifs can grow in numbers and complexity. Autoregulation, multicomponent loop, single input, multiple inputs, feedforward and regulators chain are some of the simplest and well known network motifs. Figure. 2.5 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 represents a true functional linkage between a pair of genes (Lee et al. (2007)). The linkage between two genes were measured based on the evidence codes shown in Table. 2.1.

We have constructed the connectivity matrix between genes and associated transcription factors from the gene to gene linkage and log-likelihood scores. We choose 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

| **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).

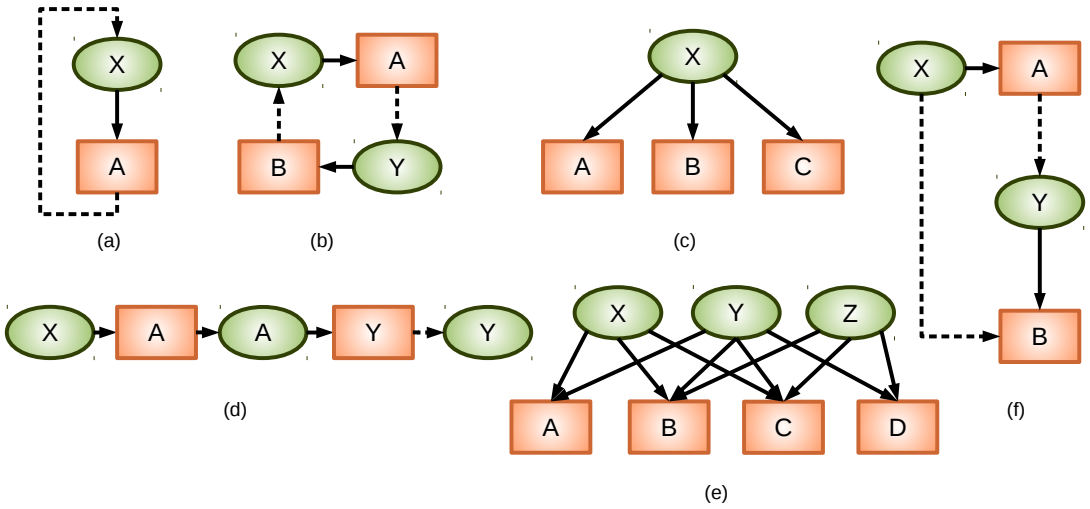


Fig. 2.5 Some basic examples of network motif: (a) autoregulation, (b) multi-component loop, (c) single input motifs, (d) regulators chain, (e) multiple inputs and (f) feedforward motifs. Here green ovals represents regulators and orange rectangles represents gene promoters. Solid lines represent binding of a regulator to a promoter while dashed arrow represents gene encoding regulators binding their respective regulators.

start our experiments. But if needed we can consider any of the evidence to reconstruct the connectivity matrix. From the gene list we have 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.

## 2.7 Result Analysis

We have 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 functionality of this tool and working pathway. There is no established benchmarks 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 motif, i.e. autoregulation, multi-component loop, feedforward loop, single input, multi-input motif and regulator chain were visible for transcription factor activity. So it was a

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

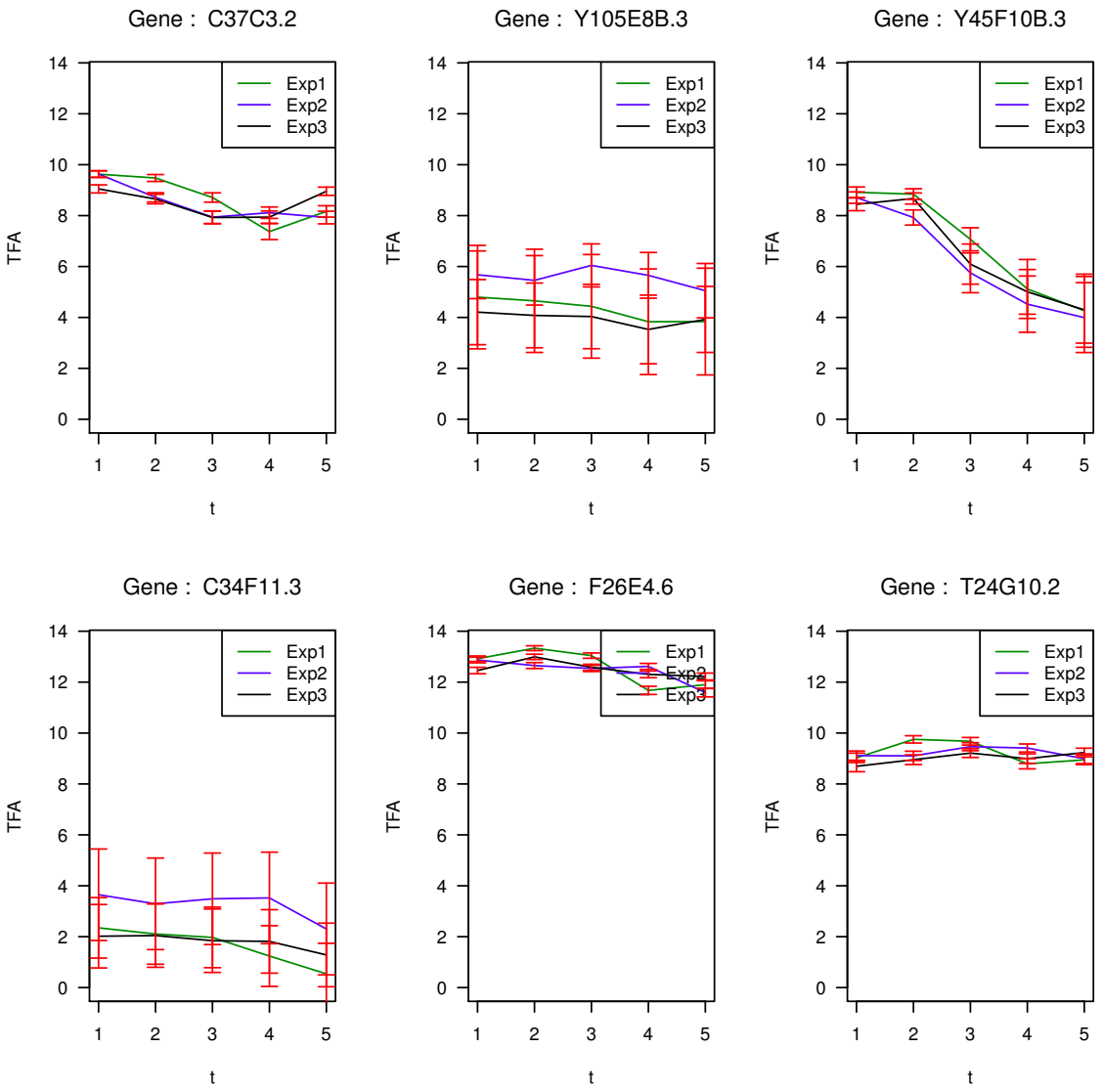


Fig. 2.6 Gene Specific transcription factor activity of ZK370.2 on (top left to right) C37C3.2, Y105E8B.3, Y45F10B.3 and (bottom left to right) C34F11.3, F26E4.6, T24G10.2. $x$-axis represent the time stage of the experiments and $y$-axis represent the gene expression level for transcription factor activities. Three different line represent TFA for three replication and red vertical lines are error bars.

mammoth task to choose all the transcription factors and show their activity. Rather we chose some random transcription factor and tried to find out its activity on different genes.

As a random sample, we chose transcription factor ZK370.2 and find its activity on different genes. Figure 2.6 shows that transcription factor ZK370.2 can regulate C37C3.2, Y105E8B.3, Y45F10B.3, C34F11.3, F26E4.6 and T24G10.2. In the dataset we had three replication of same experimental setup and its outcome. We performed our *in-silico* experiments for individual replicates and collected the results. Later we presented all the outcome together by plots (some examples are Figure:2.7, Figure 2.6). From our experimental result we can say that the dynamics for some of the gene specific regulations (ı.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 2.7 shows its activity on different genes.

### 2.7.1 Gene with multiple regulators

For the case of 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, ene 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 significance level can be determined by using the posteriori variance. Though for some cases the expression level is quite low and noise margin is significantly high (examples of binding with insignificant regulation) but by ranking the transcription factors according to the maximum gene specific TFA. We can also rank these genes using the ranking method proposed by Kalaitzis and Lawrence (2011) to rank the differentially expressed gene expressions.

### 2.7.2 Different clusters and related active TF

Clustering of genes is used to identify 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 visualize the data and also help to facilitate the functional

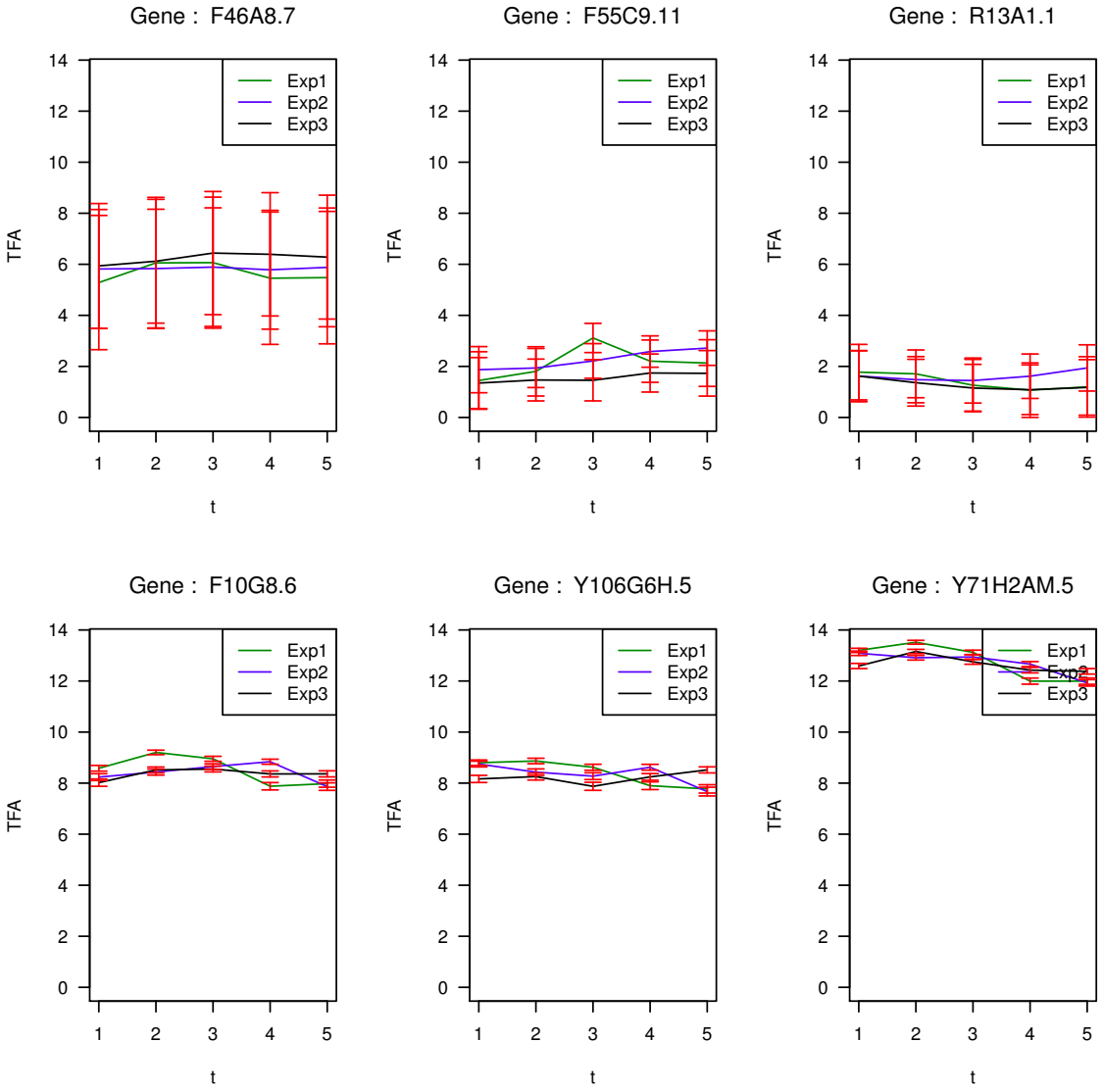


Fig. 2.7 Gene Specific transcription factor activity of T20B12.8.3 on (top left to right) F46A8.7, F55C9.11, R13A1.1 (bottom left to right) F10G8.6, Y106G6H.5 and Y71H2AM.5. $x$-axis represent the time stage of the experiments and $y$-axis represent the gene expression level for transcription factor activities. Three different line represent TFA for three replication and red vertical lines are error bars.

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

Table 2.2 Example of genes regulated by multiple TF

annotation of the not yet characterized genes. If an uncharacterised gene belongs to a cluster, then it could possibly have similar function and may dominated by genes of same function (Pe'er (2003)). Cossins et al. (2007) performed some cluster analysis of genes based on different phenotypes. They constructed the basic clusters with the following phenotype properties

**Cluster 1 - Chill upregulated:** basically 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 organization and biogenesis, DNA packaging, chromatin architecture chromatin modification, negative regulation of developmental process, chromatin remodelling, regulation of developmental process, DNA metabolic process larval development (sensu Nematoda), organelle organization and biogenesis, post-embryonic 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'.

Figure 2.8 shows the heat map generated from DNA microarray data to reflect the gene expression values at different temperature and their basic clusters (Cossins et al. (2007)). Based on the above clusters, we also investigate about the transcription


Fig. 2.8 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.

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

Table 2.3 Active TF on different clusters

factors active in different clusters. Table 2.3 shows the number of active transcription factors for each clusters. For further analysis we can present the full list.

## 2.8 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 have 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 ranking score of different samples.

Figure 2.9 shows the Pearson correlation between different ranking scores. The correlation coefficient for all three relations (between sample 1 and sample 2, sample 2 and sample 3 and sample 3 and sample 1) were 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.

## 2.9 Discussion

In this chapter, at 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*. Our tool can infer the gene specific transcription factor activity as a form of latent variable modelling. Earlier our gene expressions were collected as the gene response to chill exposure for *C. elegans*. We constructed our connectivity information between genes and transcription factors from the evidence of gene to gene interaction to model the TFAs. Earlier the dynamics of TFAs were obtained for a unicellular microorganism (yeast), but our tool is capable to determine transcription

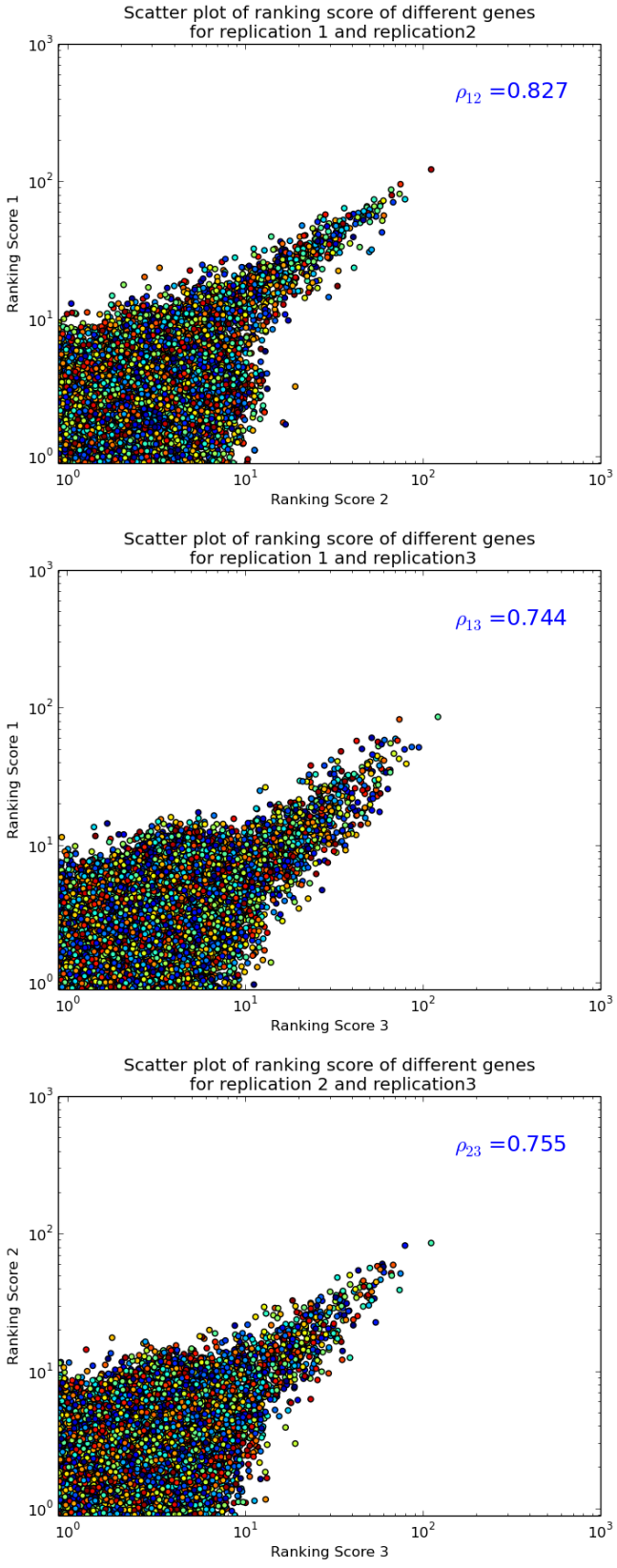


Fig. 2.9 Pearson's correlation between different ranking scores. For each figure number at the top right position represent Pearson's correlation

factor activities for a multicellular eukaryote (*C.elegans*). 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 processes is one of the most widely used families of stochastic processes for modelling dependent data observed over time, or space, or time and space 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) used Gaussian process for time series prediction date backs to the 1940's. But probably the history of 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 vector which have a normal distribution. 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 when he was serving as a assistant to professor H. L. d'Arrest at the Copenhagen Observatory, Denmark.

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 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 Gaussian process for machine learning and most of the discussed in this chapter can be found there in detailed form.

## 3.2 The Regression Problem

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

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

Supervised learning may be further sub-categorized in two fundamental tasks: regression and classification. Regression problem deals with estimating 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 some prediction of a continuous output variable at a desired input, based on a training input output data set. The input data can be any type of object 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 for a desired input. In the case of Bayesian inference, a prior distribution over functions is required. Then the model goes through training process and update the prior, based on the training data set. Let's 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 $\mathbf{y} \equiv \{y_n\}_{n=1}^N$, $\mathbf{y}_i \in \mathbb{R}$ are the training outputs. Now a key question arises, how can we consider a distribution over an infinite dimensional object as a function?

Although using plain and simple statistics regression problem can be solved, to model a more complex and specific learning task with improved reliability and robustness Gaussian process has proven as a better selection. Theoretically Gaussian process regression corresponds to Bayesian linear regression with infinite number of basis functions. In practice, the number of basis function is very high and doesn't not vary with the parameters. So, we can say Gaussian process models are non-parametric. A Gaussian process model can be used for regression model having an object featuring

infinite dimensionality. Even Gaussian processes has advanced beyond the regression model and now using 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 considered at the training level from a 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 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 Gaussian process is a fully probabilistic Bayesian model, and definitely will 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

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

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 Sec. 2.2 showed how marginalization and conditioning is related with 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 vector. Let's $\mathbf{f} = \{f_1, f_2, f_3, ..., f_N\}$ corresponds 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:

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

where $\boldsymbol{\mu}$ is the mean and $\mathbf{K}$ is the covariance matrix. Both are 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

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

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

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

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

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

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

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 $\mathbf{t} = \mathbf{t} - \overline{\mathbf{t}}$, where $\overline{\mathbf{t}}$ 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 a kernel, kernel function or covariance kernel) characterizes the properties or nature of the sample drawn from the Gaussian process. The covariance function ..... the modelling assumptions we wish to incorporate in our application. The mandatory requirement of a covariance matrix is to be symmetric positive semi-definite. So, as long as the covariance function generates symmetric positive semi-definite[1] 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. Our main goal of this thesis is to develop a covariance functions suitable for transcription factor activity analysis and clustering gene expressions. In this chapter we will discuss about 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 at Rasmussen and Williams (2006).

Any form of covariance function is acceptable, provided it satisfy 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

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

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 depends on the absolute location of the corresponding inputs, rather it depends on distance separating them.

If the covariance does not only depend on the distance between the data points in the input space, rather model need to adapt to functions where smoothness varies with the inputs, a non-stationary covariance functions will be required. There are many interesting non-stationary covariance functions. Depending on the nature or trend a careful selection of appropriate covariance function is essential. One of the simplest example of non-stationary covariance function which have a linear trend can

[1] A matrix $\mathbf{C}$ is called positive-semidefinite 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 $\mathbf{z}$.

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 at 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 parameter for output variance $a$ and lengthscale parameter $l$ govern the property of sample functions and are commonly known as hyperparameters. 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 function was generated for a given input range by drawing a sample from the multivariate Gaussian using Eq. 3.2 with zero mean. The smoothness of the sample function depends on the Eq. 3.10. Function variable located closer in the input space are highly correlated, whereas function variable located at 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.

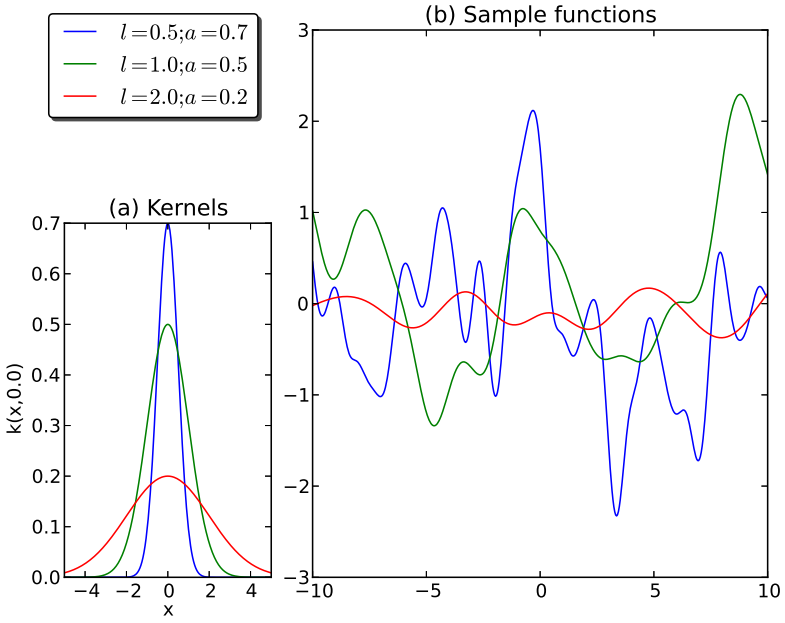


Fig. 3.1 Exponentiated quadratic kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left

### 3.4.2 Rational Quadratic Covariance Function

The rational quadratic covariance function is equivalent to adding together multiple exponentiated quadratic covariance functions having different lengthscale. Gaussian process prior kernel function expects smooth function with many lengthscale. In the Equation 3.11 the parameter $\alpha$ 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 $\alpha \to \infty$, then 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:

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

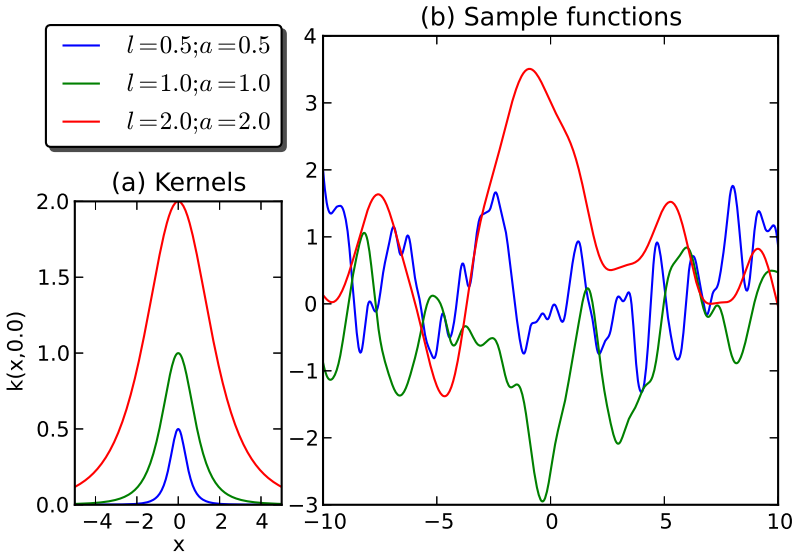


Fig. 3.2 Rational quadratic kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left

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

### 3.4.3 The Matérn Covariance Function

The Matérn class of covariance function 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 $a, l, \nu$ are positive hyperparameters, $K_{\nu}$ is a modified Bessel function and $\Gamma(.)$ is the Gamma function. Hyperparameter $\nu$ controls the roughness of the function and as 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 $\nu$ the sample functions are significantly rough.

The simpler form of Matérn covariance function is obtained when $\nu$ 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

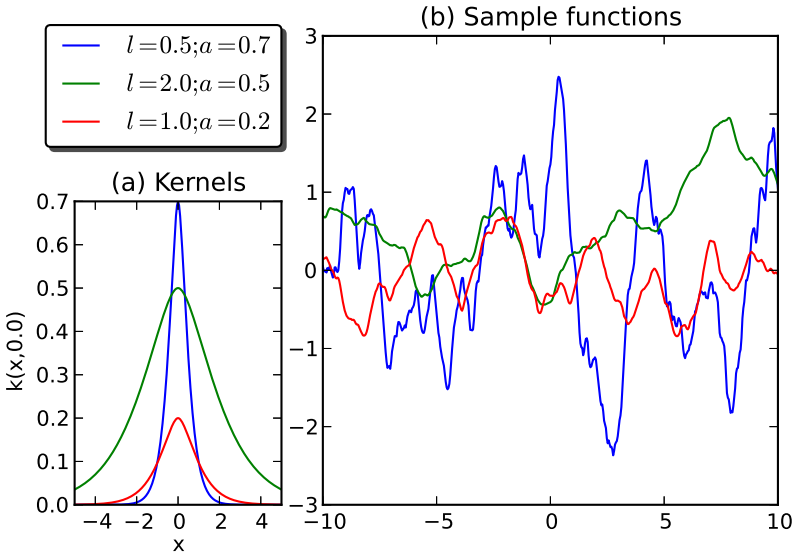


Fig. 3.3 The Matérn32 kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left

(1965) derived the general expression as follows-

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

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

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

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

### 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.

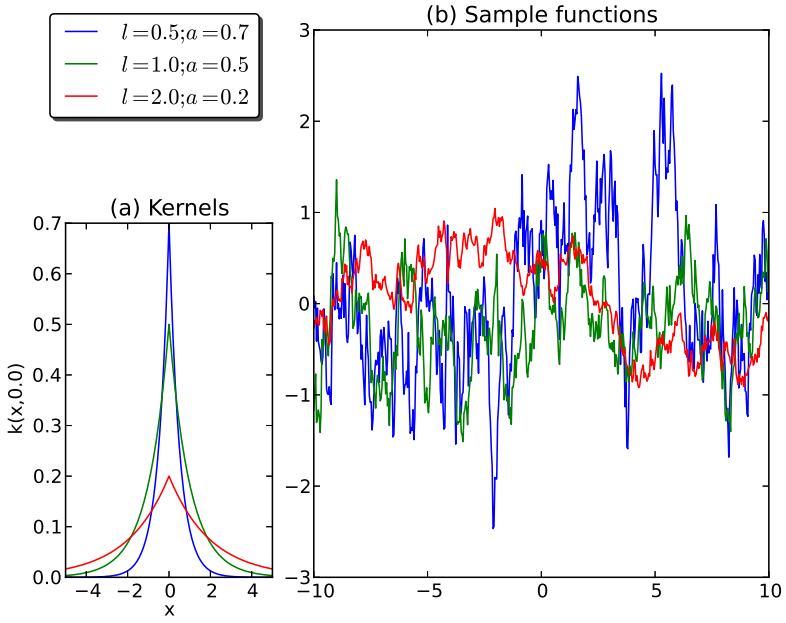


Fig. 3.4 The OU kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left

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 having the exactly same amplitude parameter $a$ and lengthscale parameter $l$.

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

### 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 $\lambda$ is a positive constant, $\xi$ and $\xi'$ are equidistributed and uncorrelated random variables. Using the basic trigonometry Eq. 3.17 can be written as

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

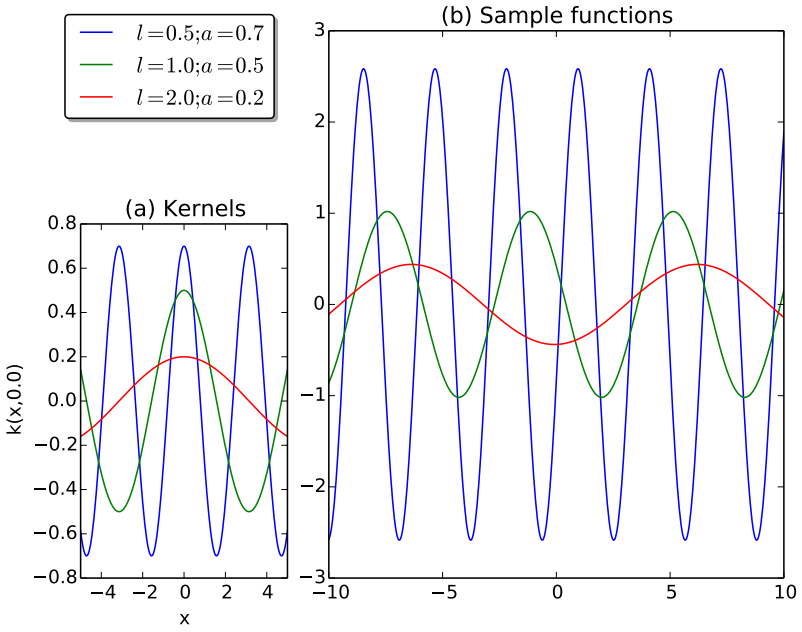


Fig. 3.5 The Cosine kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left

where $R^2 = \xi^2 + \left(\xi^{'}\right)^2 \geqslant 0$ and $\psi = \arctan\left(\frac{\xi}{\xi^{'}}\right) \in (-\pi, \pi)$. Let's $\mathbb{E}[\xi] = 0$, 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 $\xi$ and $\xi'$ are equidistributed and uncorrelated. The cosine kernel is a stationary kernel regardless of the distribution of $\xi$. 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 Gaussian process, yet we may need to model a complicated structure which is not expressed very well by any known kernel. To model such s structure, we may built our own 'customized kernel'

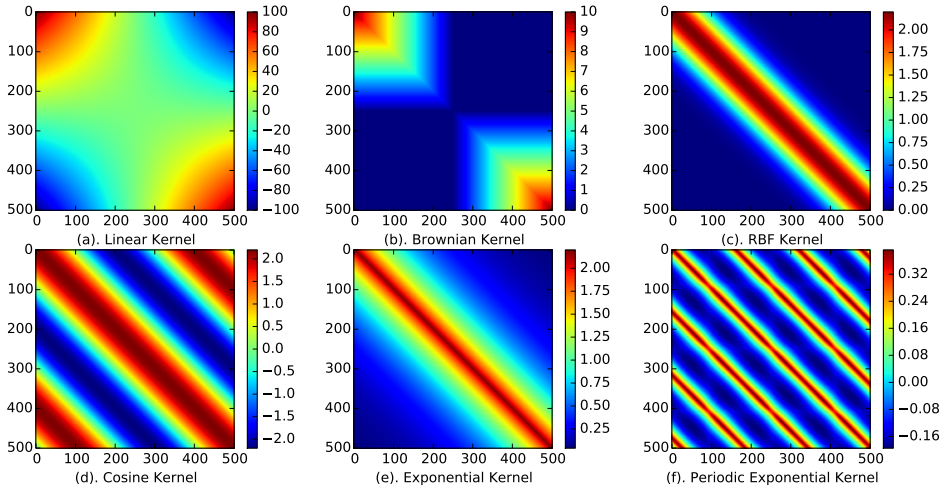


Fig. 3.6 Representation of some basic kernels using the same lengthscale and variance (a) Linear kernel (b) Brownian kernel (c) Exponentiated Quadratic kernel (d) Cosine kernel (e) Ornstein-Uhlenbeck kernel and (f) Periodic Exponential kernel

with the requirement of the structure or, desired properties. Addition of two Gaussian variables is a Gaussian. Again, scaling a Gaussian also leads to a Gaussian. There two basic mathematical properties helps to develop a range of kernels from a very simple to complex one. Appendix.. shows the addition and multiplication properties of Gaussian processes.

Figure 3.6 shows the 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). Exponential kernel (f). Periodic Exponential kernel. These kernels are the realization of different covariance function[3]. These kernels (including other) facilitate to construct new kernels or customize 'on demand 'of the structure with the desired properties.

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

[3]We have not describe the Linear, Brownian and Periodic Exponential kernels here. Detail description is available at Rasmussen and Williams (2006) and their *Python*-based implementation is available at Authors (2014).

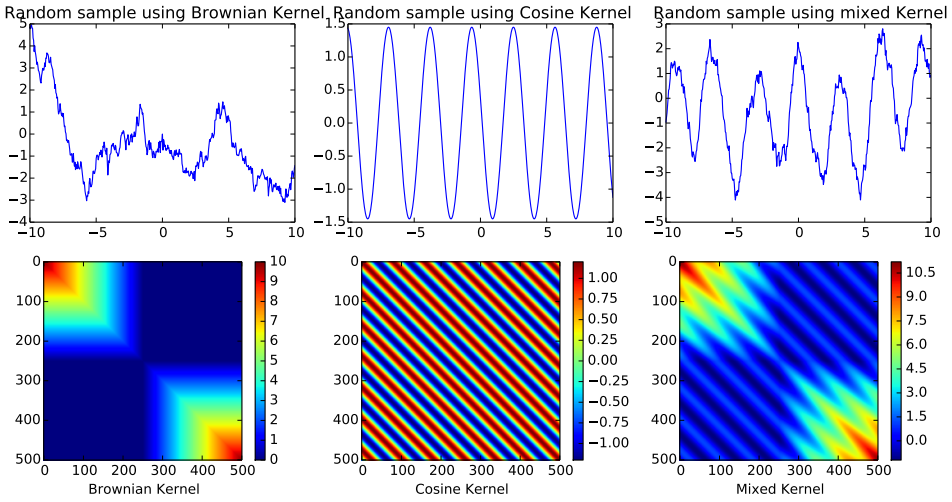


Fig. 3.7 Construction of 'made by order'kernel adding two basic kernels: an example of a univariate data which is globally periodic and locally governed by some random motion. (top-left) sample taken using Brownian kernel, (top-middle) sample taken using Cosine kernel, (top-right) sample taken using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

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 multivariate Gaussian distribution. Let's consider that we have observations $\mathbf{f}$ of a function at observation points $\mathbf{x}$. Now we wish to predict the values of that function at observation points $\mathbf{x}_\star$, which we are representing by $\mathbf{f}_\star$. Then the joint probability of $\mathbf{f}$ and $\mathbf{f}_\star$ can be obtained from equation 3.19-

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

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

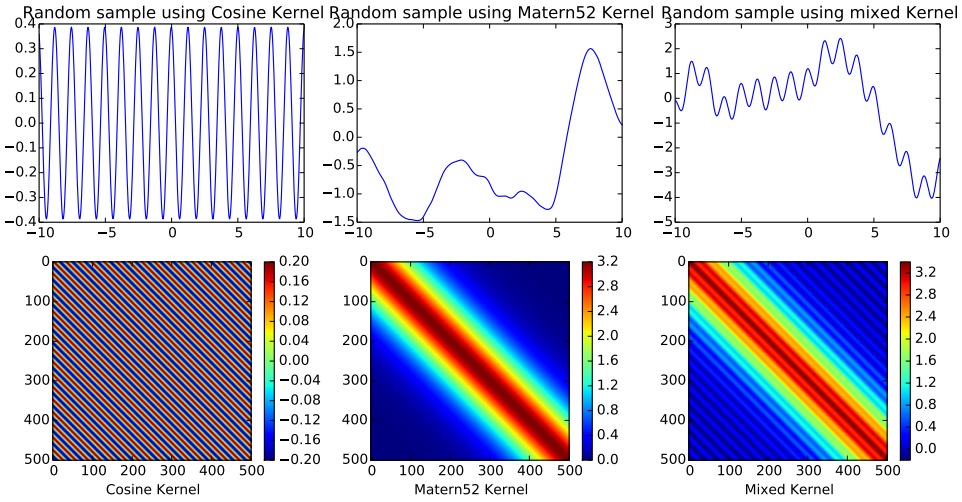


Fig. 3.8 Construction of a new kernel using Cosine kernel and Matérn kernel. (top-left) sample taken using Cosine kernel, (top-middle) sample taken using Matérn kernel, (top-right) sample taken using newly constructed kernel by adding two kernels, (bottom-left) Cosine kernel, (bottom-middle) Matérn kernel,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

property of a multivariate Gaussian is used to perform regression. The conditional property can be represented by the equation 3.20

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

In an ideal case the observations $\mathbf{f}$ is noise free but in practice it is always corrupted with some noise. Let's consider $\mathbf{y}$ is the corrupted version of $\mathbf{f}$. If we consider this noise as Gaussian noise then we can write $p\left(\mathbf{y}|\mathbf{f}\right) = \mathcal{N}\left(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}\right)$, where $\sigma^2$ is the variance of the noise and $\mathbf{I}$ is the identity matrix with appropriate size and marginalise the observation $\mathbf{f}$. Then the joint probability of $\mathbf{y}$ and $\mathbf{f}_{\star}$ can be represented by the equation 3.21.

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

Regression with Gaussian process is a Bayesian method. From the knowledge of a *prior* over a function, we proceed to 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 observed

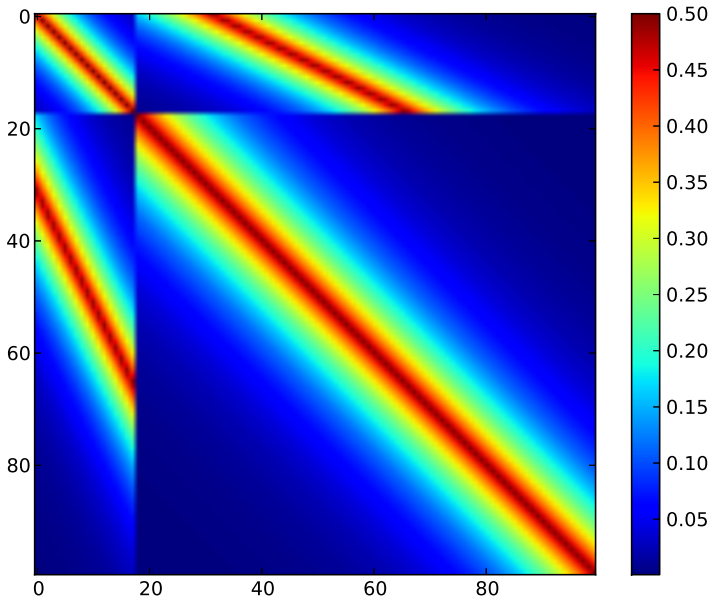


Fig. 3.9 Overall representation of covariances between training and test data

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 prediction

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 prediction 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 then the conditional distribution will be $p(\mathbf{f}_\star|\mathbf{y})$. Both of the distribution are also Gaussian,

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

The mean of the conditional distribution in Equation 3.22 is

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

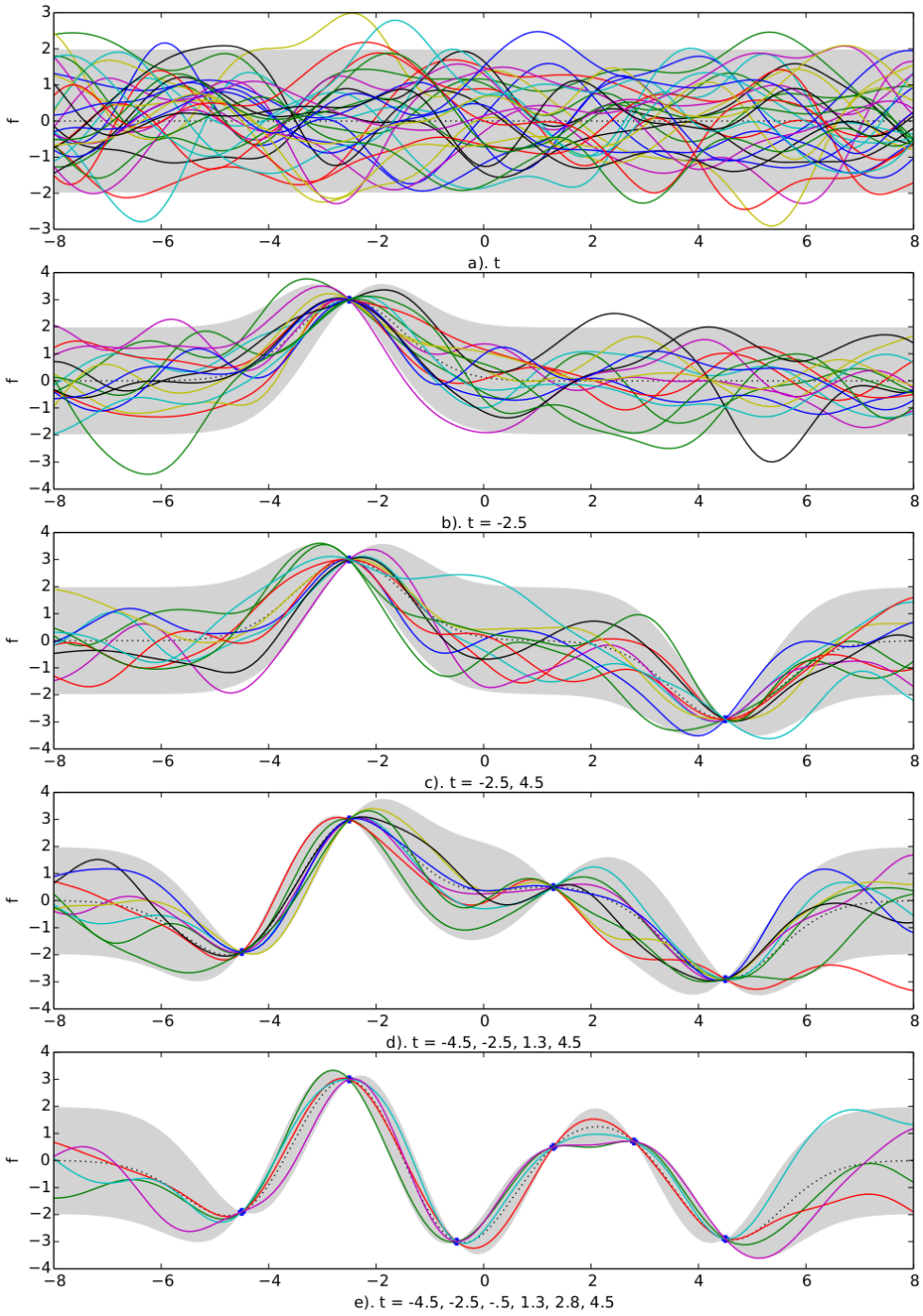


Fig. 3.10 A visual representation of Gaussian process regression: Modelling a one dimensional function using Gaussian process. Coloured solid lines represent different samples from the process and dotted line is the mean function. The shaded area is 95% confidence interval. (a). A Gaussian process not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

and its covariance is given by

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

These results can be calculated using block matrix inverse rules. The derivation can be found in appendix section (Appendix ??). 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 dotted line is the mean function. The shaded area is 95% confidence interval. (a) of Figure 3.10 represent a Gaussian process not conditioned on any data points. Without any observations, the prior uncertainty about the underlying function is constant everywhere. (b). to (e). of Figure 3.10 shows some posterior samples after conditioning on different amount of data.

### 3.6.2 Hyperparameter Learning

To construct the covariance function still we need to consider the hyperparameters and optimize those. 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 maximum likelihood. If we consider all the hyperparameters $\alpha$, $\sigma^2$ and $l$ in to a vector $\boldsymbol{\theta}$, then 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:

$$p\left(\mathbf{y}|\boldsymbol{\theta}\right) = -\frac{D}{2}log2\pi - \frac{1}{2} \times log\left|\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right| - \frac{1}{2}\mathbf{y}^T\left[\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{y} \tag{3.25}$$

We can have the Log maximum likelihood by:

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

## 3.7 Toward the GP model of TFA

We are interested to develop 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 ??) based probabilistic approach of Sanguinetti et al. (2006). From Chapter

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

2 Section 2.5 we have the probabilistic gene specific TFAs as

$$\mathbf{b}_{n(t+1)} \sim \mathcal{N}(\gamma \mathbf{b}_{nt} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}) \tag{3.27}$$

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

$$\mathbf{b}_{n(k+1)} \sim \mathcal{N}\left(\gamma \mathbf{b}_{nk} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}\right), \tag{3.28}$$

and

$$\mathbf{b}_{n_1} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \tag{3.29}$$

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

$$u_{k+1} \sim \mathcal{N}\left(\gamma u_k + (1-\gamma)\,\mu, (1-\gamma^2)s\right), \tag{3.30}$$

where $\mu$ 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

$$t_k = kDt. \tag{3.31}$$

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

$$du = -\lambda\,(u-\mu)\,dt + q^{1/2}dB, \tag{3.32}$$

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

$$u(t_k) = au(t_{k-1}) + b\mu + w_{k-1}, \tag{3.33}$$

where $w_{k-1} \sim \mathcal{N}(0, c)$ with
$a = \exp(-\lambda Dt)$

$b = \int_0^D t\exp(-\lambda(Dt-s))ds$
$= 1 - \exp(-\lambda Dt)$

$c = \int_0^D t\exp(-\lambda(Dt-s))q\exp(-\lambda(Dt-s))ds$

$= q \int_0^D t \exp(-2\lambda(Dt - s))ds$
$= [q/(2\lambda)][1 - \exp(-2\lambda Dt)]$

That is,

$$u_{k+1} \sim \mathcal{N}\left(au_k + b\mu, c\right). \tag{3.34}$$

We can now match the coefficients:

$$a = \exp(-\lambda Dt) = \gamma \tag{3.35}$$

$$b = 1 - \exp(-\lambda Dt) = 1 - \gamma \tag{3.36}$$

$$c = (1 - \gamma^2)s = [q/(2\lambda)][1 - \exp(-2\lambda Dt)] \tag{3.37}$$

Equation 3.35 quite luckily has a nice solution $\gamma = \exp(-\lambda Dt)$ and from Equation 3.37 we will 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 arbitrarily 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 we get

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

When we start from variance $s = q/[2\lambda]$, then the process will indeed be 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

$$\mathbf{K}_b(t, t') = \mathbf{\Sigma}\boldsymbol{\gamma}^{|t-t'|} \tag{3.38}$$

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

$$\mathbf{db} = -\lambda(\mathbf{b} - \boldsymbol{\mu})\mathbf{dt} + Q^{1/2}\mathbf{dB}. \tag{3.39}$$

## 3.8 Gaussian Process: Pros and Cons

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

There are several issues which may make Gaussian processes sometimes difficult to use. The generic inference and learning algorithm where we need to inverse the matrix 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 data-points. An exact inference for typical 'Big data'could be nightmare. 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 and sparse approximations could be challenging while working with Gaussian process. However, Gaussian process framework GPy (Authors (2014)) can automatically deal with the last two issues.

## 3.9 Discussion

In this chapter we briefly describe Gaussian process, regression problem and regression with 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 clustering gene expressions. In this chapter we briefly describe about some commonly used kernels. We also mentioned about hyperparameter learning. Finally we justified the rationale behind choosing the Ornstein-Uhlenbeck kernel to model the transcription factor activity using Gaussian process. At the next chapter 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 for reconstructing transcription factor activities given gene expression profiles and a connectivity matrix (binding data) between genes and transcription factors. 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.

We note that the linear Gaussian model is equivalent to a Gaussian process with a particular covariance function. We therefore build a model directly from the Gaussian process perspective to achieve the same effect. We introduce a computational trick, based on judicious application of singular value decomposition, to enable us to efficiently fit the Gaussian process in a reduced 'TF activity' space.

First we load in the classic Spellman et al. (1998) Yeast Cell Cycle data set. The cdc15 time series data has 23 time points. We can load this gene expression data in with GPy.

Time series of synchronized yeast cells from the CDC-15 experiment of Spellman et al. (1998). Two colour spotted cDNA array data set of a series of experiments to identify which genes in Yeast are cell cycle regulated. We can make a simple helper function to plot genes from the data set (which are provided as a pandas array).

Our second data set is from ChiP-chip experiments performed on yeast by Lee et al. (2002). These give us the binding information between transcription factors and genes. In this notebook we are going to try and combine this binding information with the gene expression information to infer transcription factor activities.

## 4.1 Model for Transcription Factor Activities

We are working with *log* expression levels in a matrix $\mathbf{Y} \in \Re^{n \times T}$ and we will assume a linear (additive) model giving the relationship between the expression level of the gene and the corresponding transcription factor activity which are unobserved, but we represent by a matrix $\mathbf{F} \in \Re^{q \times T}$. Our basic assumption is as follows. Transcription factors are in time series, so they are likely to be temporally smooth. Further we assume that the transcription factors are potentially correlated with one another (to account for transcription factors that operate in unison).

**Correlation Between Transcription Factors**: If there are $q$ transcription factors then the correlation between different transcription factors is encoded in a covariance matrix, $\mathbf{\Sigma}$ which is $q \times q$ in dimensionality.

**Temporal Smoothness**: Further 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 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 desigingin kernel using *coregionalization* at Chapter 5 in Section 5.2.2.

## 4.2 Relation to Gene Expressions

We now assume that 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

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

## 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 tell us that

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

where

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

This results in a covariance function that is of size $n$ by $T$ where $n$ is number of genes and $T$ is number of time points. However, we can get a drastic reduction in the size of the covariance function by considering the singular value decomposition 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 zero 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 $O(T^3n^3)$ computational complexity. We are going to use a rotation trick to get the likelihood.

## 4.4 The Main Computational Trick

### 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 paper 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 $\mathbf{S}$, but 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 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

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

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

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

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

$$L=L_u+L_q+\text{const} \quad (4.13)$$

where

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

and

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

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 $\sigma^2$ on $\hat{\mathbf{y}}_u$ alone using $L_u$. Once this is done, fix the value of $\sigma^2$ in $L_q$ and optimize 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 and joint maximisation of the likelihood of $L_u$ and $L_q$.

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 with over smoothness. Figure 4.1 shows activities of different transcription factors while the model was developed considering Exponentiated Quadratic kernel with White kernel in additive form.

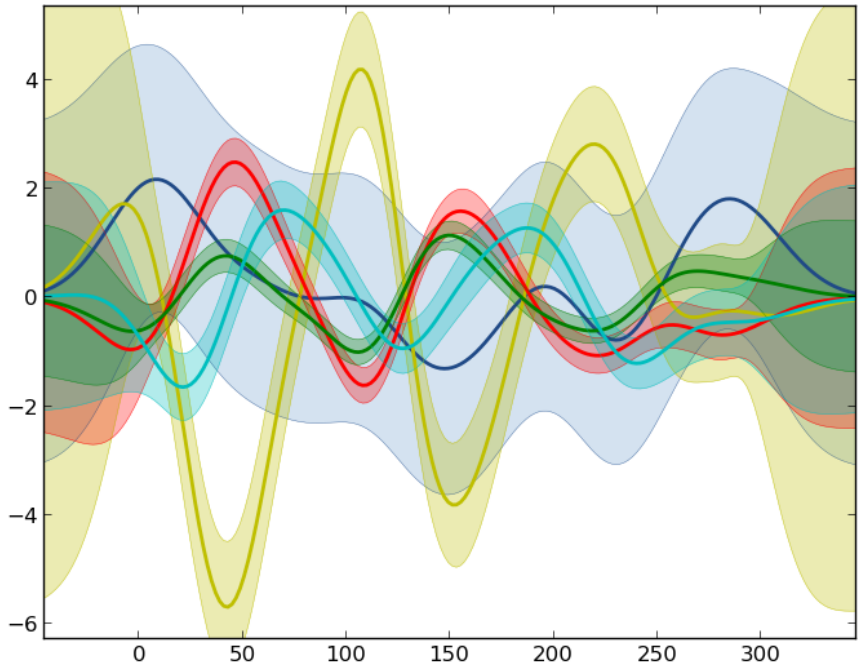


Fig. 4.1 Variation of activities of Transcription factors RBF+white kernels

Figure 4.2[1] shows transcription factor activity of ACE2. While developing the model we choose Ornstein-Uhlenbeck kernel and White kernel in additive form. We believe that 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 4.3 shows the pictorial representation of intrinsic coregionalization kernel (Equation 4.5) $\mathbf{K}_f$ considering 20 transcription factors where covariance matrix $\mathbf{\Sigma}$ of was constructed using Ornstein-Uhlenbeck kernel and White kernel in additive form.

Figure 4.4 shows some examples of transcription factors activities where model was developed with Ornstein-Uhlenbeck kernel and White kernel.

[1]The complete model is still in progress. After the final model exact figure might be updated from this one

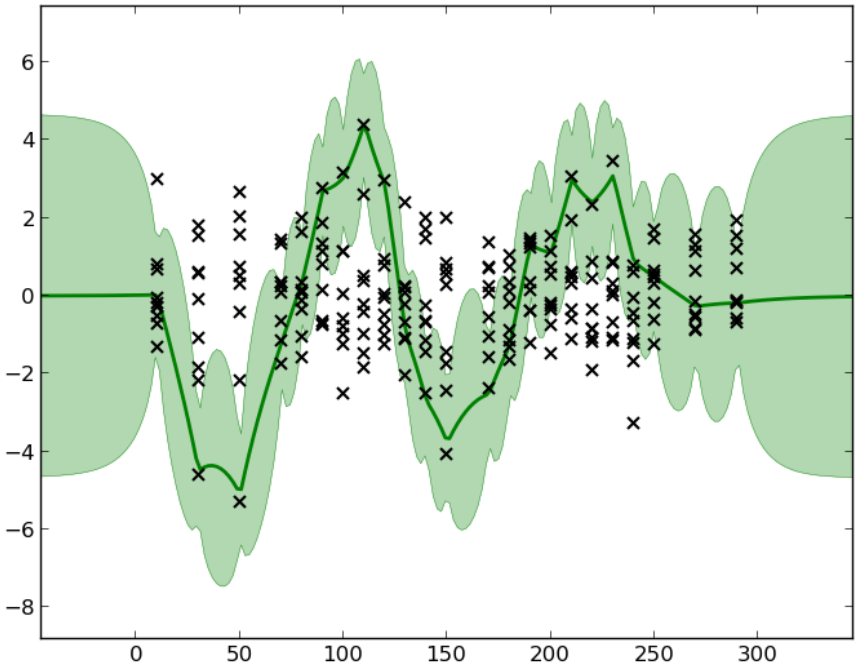


Fig. 4.2 Transcription factor activity of ACE2

## 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:

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

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

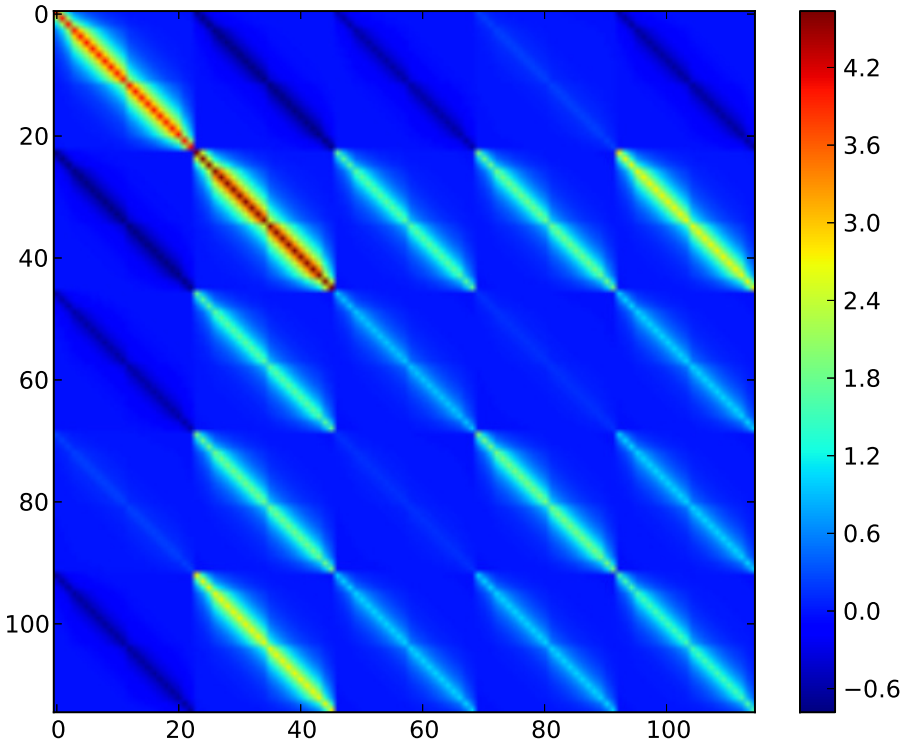


Fig. 4.3 Kernel of Intrinsic Coregionalization model $\mathbf{K}_f$ considering 6 Transcription factors where covariance matrix $\mathbf{\Sigma}$ of (*Equation*4.5) was constructed using Ornstein-Uhlenbeck kernel and White kernel in additive form

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

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

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

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

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

Now we can find the conditional distribution of $g$ for given $y_q$ 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:

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

The mean of the conditional distribution of Equation 4.16 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 of Equation 4.16 given by:

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

## 4.6 Discussion

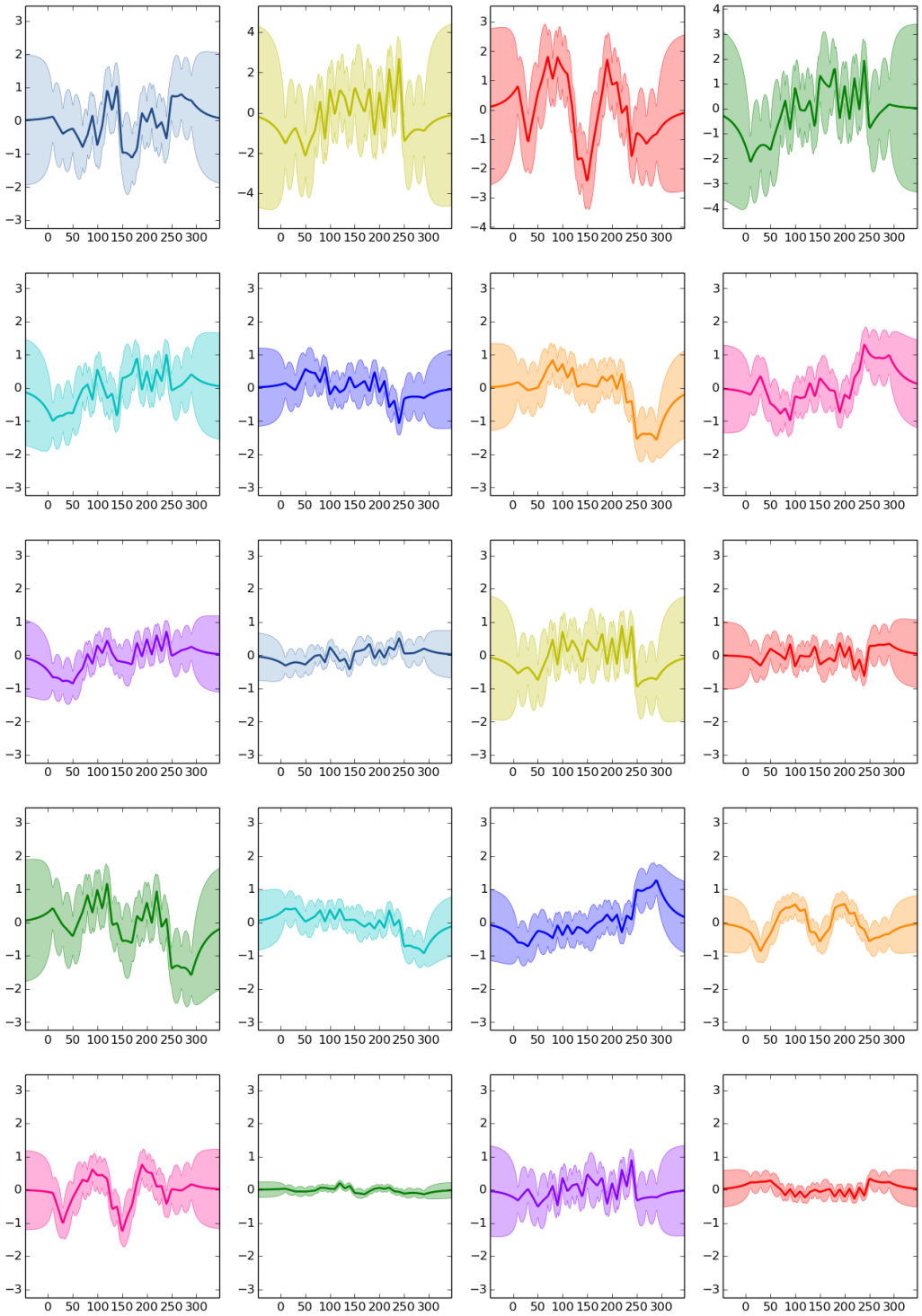

Fig. 4.4 Transcription factor activity of different TF using Ornstein-Uhlenbeck kernel and White kernel in additive form

# 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 key requirement 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 effect the speed of ALS progression by building a new model. We used Gaussian processes and here we introduce *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 optimized these models and compared them based on likelihood scores and select 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 onset and heterogeneous with variable severity in terms of speed

of progression of the disease course. Injury and cell death of motor neurons 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% patients live longer than 5 years and less than 10% 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 characterized 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 variability 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 $129 \pm 5$ days, while the $C57$ mouse strain can survive $180 \pm 16$ 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 mouse 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 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 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. Our functional annotation clustering and pathway analysis reveal some interesting information for a group of genes which might have

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

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)), 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 exchange of information. Inference on the whole structure of data is always preferable than 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 require a large number of variables optimization. Previously, Ng et al. (2006) also Medvedovic et al. (2004) used clustering method to model replicates using hierarchical structure. Both of the model 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)). Whilst Gaussian process also overcome the requirement of evenly spaced time points for time expression data. As a part of 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 background ($129Sv$ and $C57$ with transgenic and non-transgenic). We also considered their replicates (four in our case) and build 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 background or strains and there are four biological replicates. So for every individual gene we can incorporate these in an hierarchical fashion. Let's, the vector of measurements $\mathbf{y}_{nr}^{i}$ denotes 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 $i$ represents 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

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

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 $\Sigma_n$ for two genes $n$

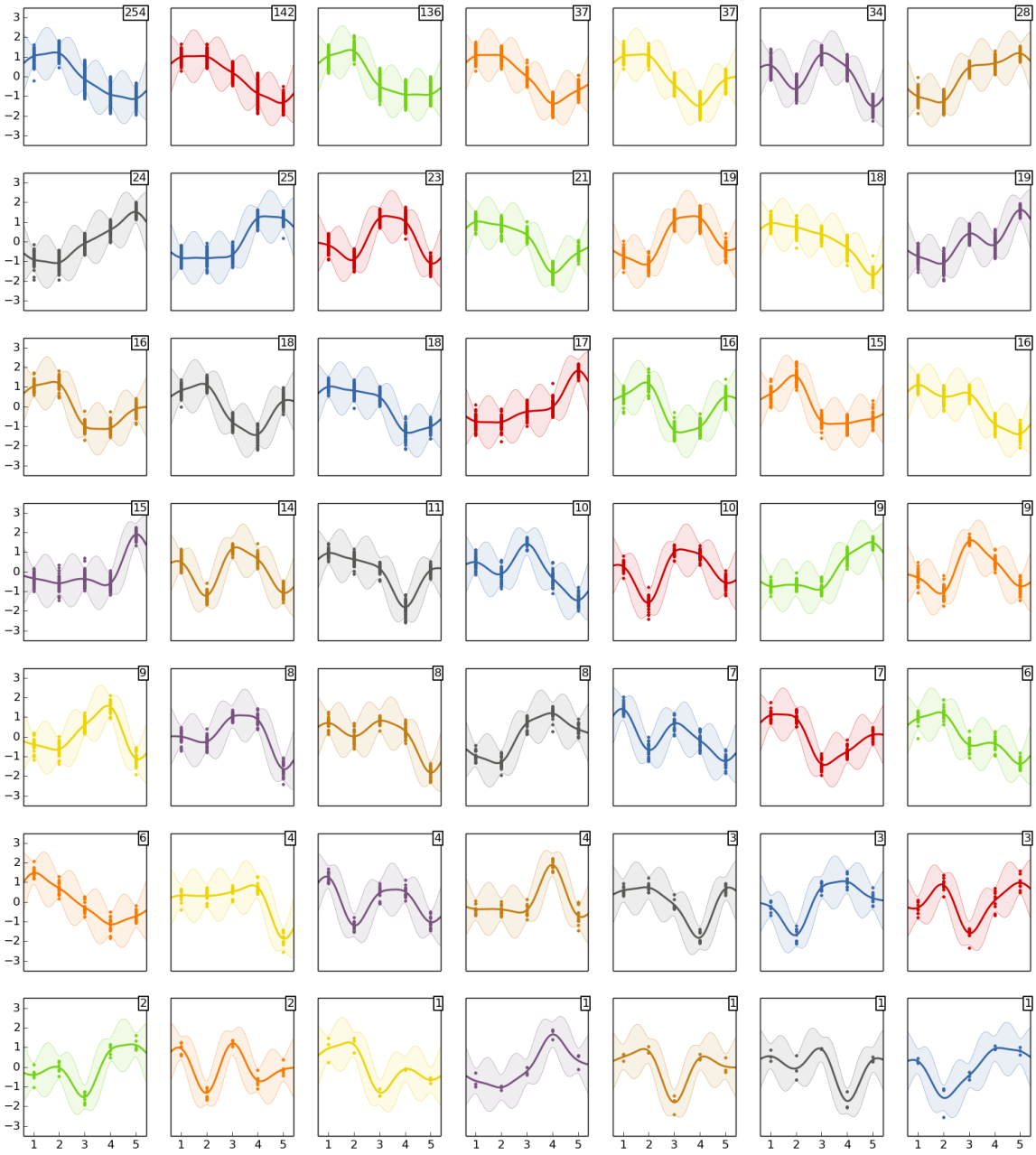


Fig. 5.1 Clustering gene expression data of *C.elegans* using the method proposed by Hensman et al. (2013b). Along $x$-axis the five time points are as described in Section 2.6.1 (shown in Figure 2.3). Number at the corner indicates number of genes belong to this cluster. Solid line represents a posterior mean and shaded area represents 95% confidence interval. We used top most 10,000 differentially expressed genes from the total 15,000+ genes.

and $n'$ are given by

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

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

### 5.2.2 Kernel Design with Coregionalization

Gaussian process models have been used already to capture structure in the data arising from temporal correlation. Our innovation is to realise that there is actually 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 multi-variate 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

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

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 $a_{d,q}$

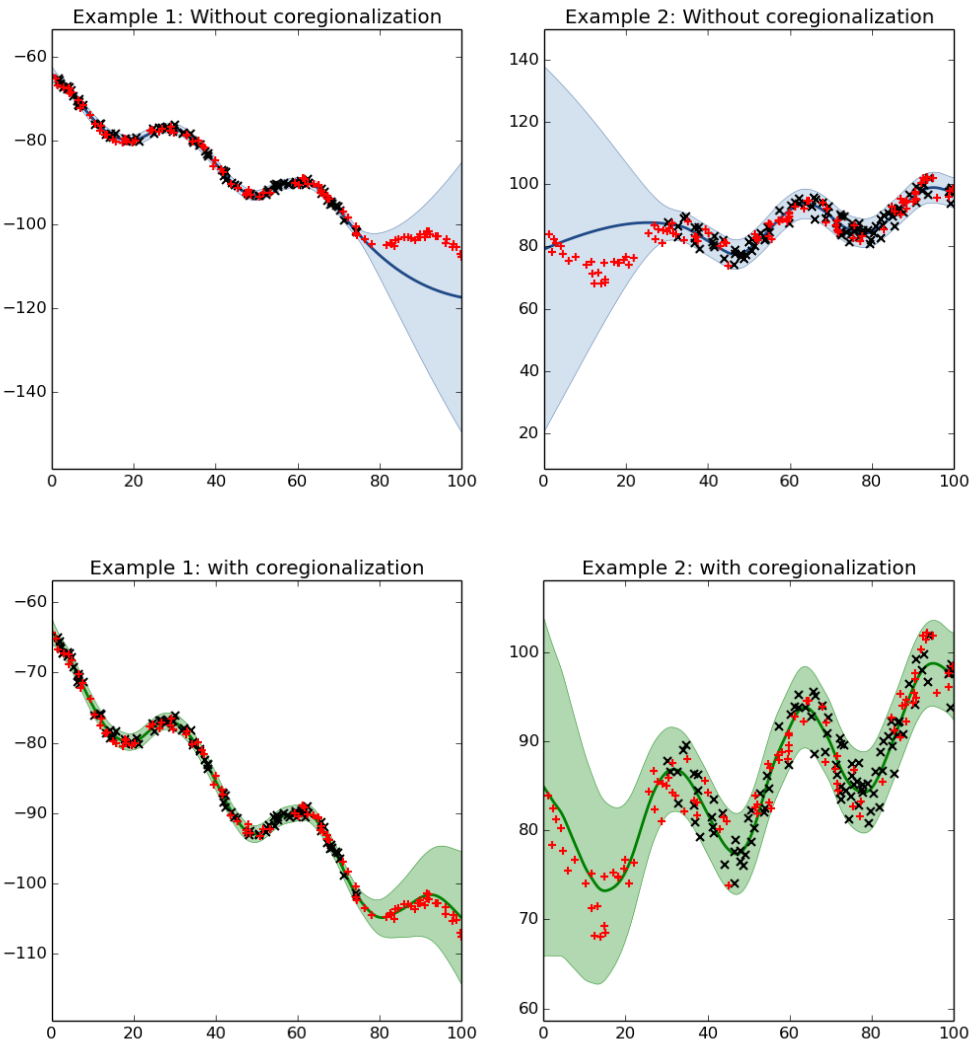


Fig. 5.2 Simple demonstration of regression using *coregionalization* model with Gaussian process. Here, training data marked with black, while red represents test data. Solid line represents a posterior mean function and shaded area represents 95% confidence interval. The independent models (top-left 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).

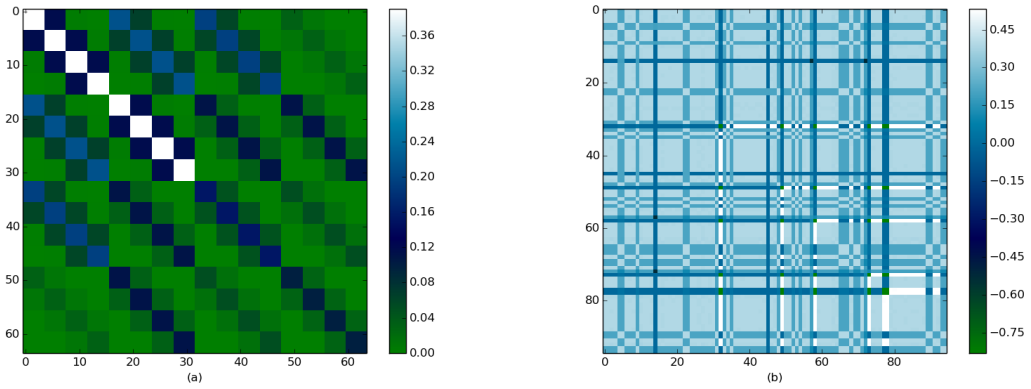


Fig. 5.3 Simple representation of kernels- (a). *Coregionalization* kernel in the input space with $64 \times 64$ dimensions; (4 strains ($129Sv-SOD1, 129Sv-Ntg, C57-Ntg$ and $C57-SOD1$) $\times$ 4 replicates $\times$ 4 time points or stages of the disease). With a closer look, we can find four primary 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 stage or time points (b). kernel after optimization considering top most 100 (an arbitrary suitable number for visualization) differentially expressed genes. This is a realization of covariance between genes, where every pixel is computed using the *coregionalization* kernel.

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, $\mathbf{A}_q \epsilon \mathbb{R}^{D \times R_q}$ 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 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 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 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 the measurements could be corrupted by noise, incorporated either at the biological origin or introduced in the measurement process. Figure 5.3 shows the representation of the coregionalized kernel in the input space and the representation of an optimized kernel where we considered only 100 (top most differentially expressed; an arbitrary number suitable for visualization) genes.

### 5.2.3 Clustering

Our aim was 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.

[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.

Partitioning genes into clusters are done by inference. Using Dirichlet process prior for mixing coefficients and partitioning Dunson (2010) proposed a method where Gaussian process was used to model the function within a cluster. This mechanisms 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 maximize the log marginal likelihood of time series data. Hyperparameters are optimized 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)).

## 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] pacakge *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 filter the quiet or inactive genes from the differentially expressed ones using Gaussian process. In addition

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

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

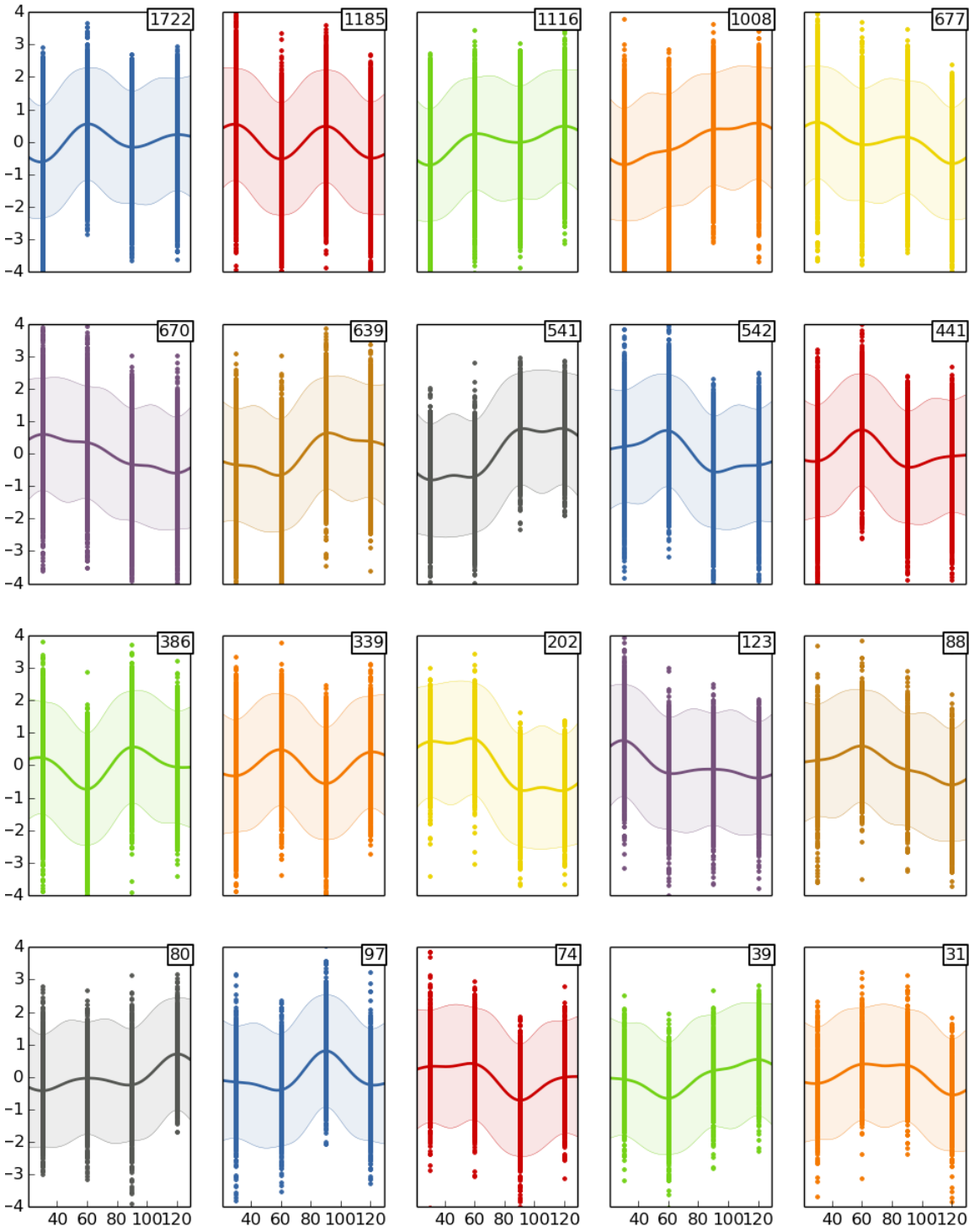


Fig. 5.4 Clustering gene expression data using the method proposed by Hensman et al. (2013b). Along $x$-axis the four time stages are pre-symptom, onset, symptom and end-stage (all the data points together formed like solid vertical lines). Number at the corner indicates number of genes belong to this cluster. Solid line represents a posterior mean and shaded area represents 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.

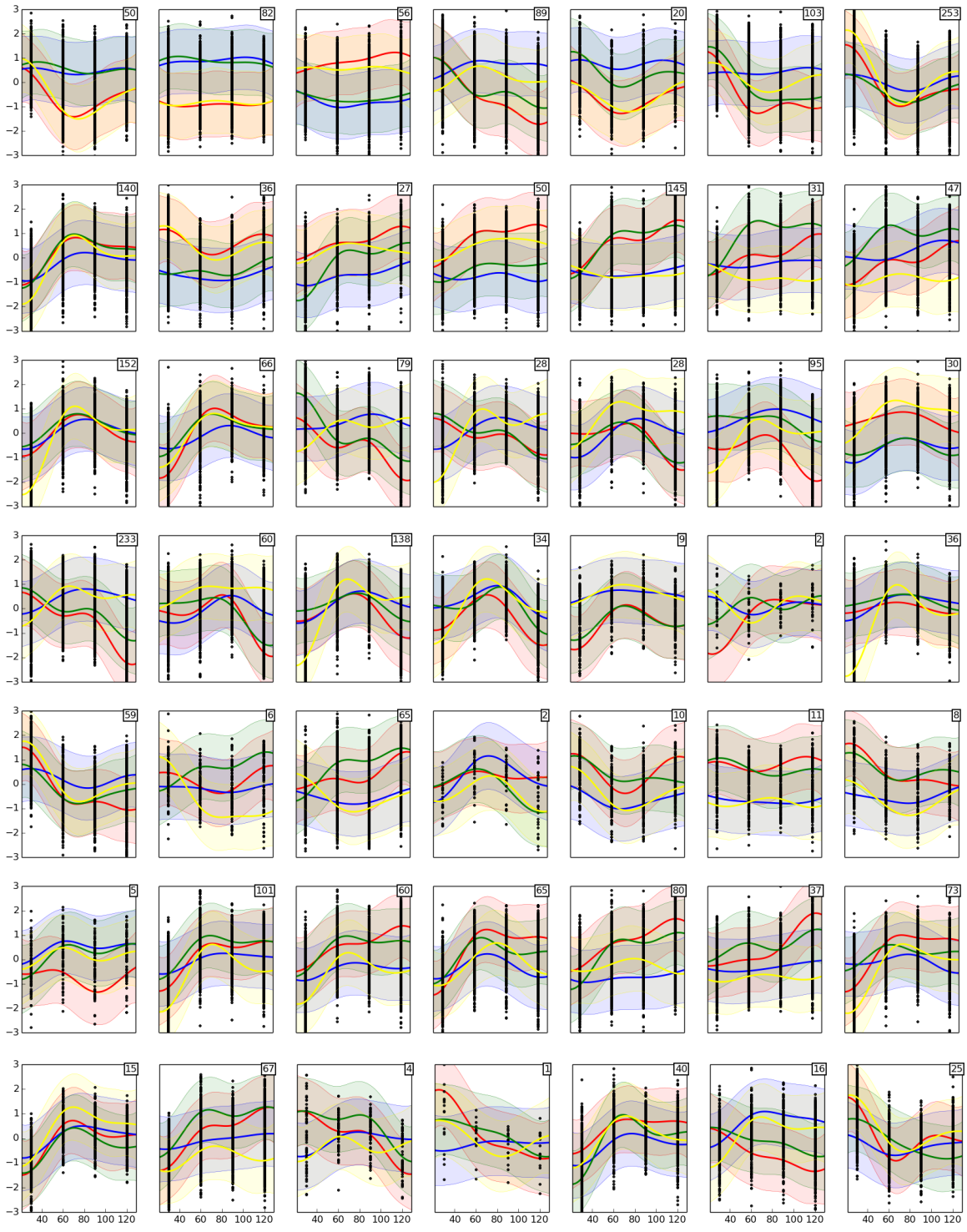

Fig. 5.5 Clustering genes expressions using hierarchy of Gaussian processes. Some representative clusters from the 203 clusters generated (top to bottom, left to right: cluster 1 to 7, 16 to 22, 31 to 37, 46 to 52, 61 to 67, 76 to 82 and 196 to 202). Along $x$-axis the four time stages are pre-symptom, 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. Number at the corner indicates number of genes belong to this cluster. Solid line represents a posterior mean function and shaded area represents 95% confidence interval.

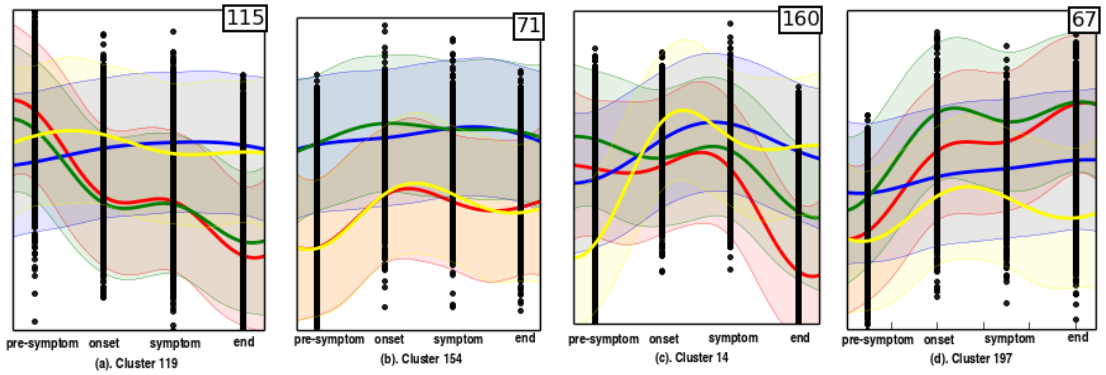


Fig. 5.6 Along $x$-axis of each individual figure four time stages are pre-symptom, onset, symptom and end-stage. Four different colours yellow, red, green and blue are representing four mouse strains $129Sv-SOD1, 129Sv-Ntg, C57-Ntg$ and $C57-SOD1$ respectively. Examples of clusters where genes from 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.

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 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 replicates were normalized to zero-mean over all the samples before the filtering.

**Cluster analysis:** In the previous selection stage we choose 10,000 genes from the total of 45,037 probe sets which were more dynamically differentially expressed. We applied our own hierarchical Gaussian process cluster model on these genes and collected the results for further experiments. Figure 5.5 shows a small part of our result. For any individual graph, along $x$-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 variation of number of genes for any individual cluster. All the clusters were indicating different dynamic behaviour of the gene set. We found a number of clusters were attractive for further analysis but the detail analysis of every cluster 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 color in our representation) has different characteristics and focussed our consideration for further analysis.

**Enrichment score analysis:** A typical biological process is regulated with 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, hyper-geometric distribution, Chi-square, Fisher's exact test. In a previous study, Huang et al. (2009a)

[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.

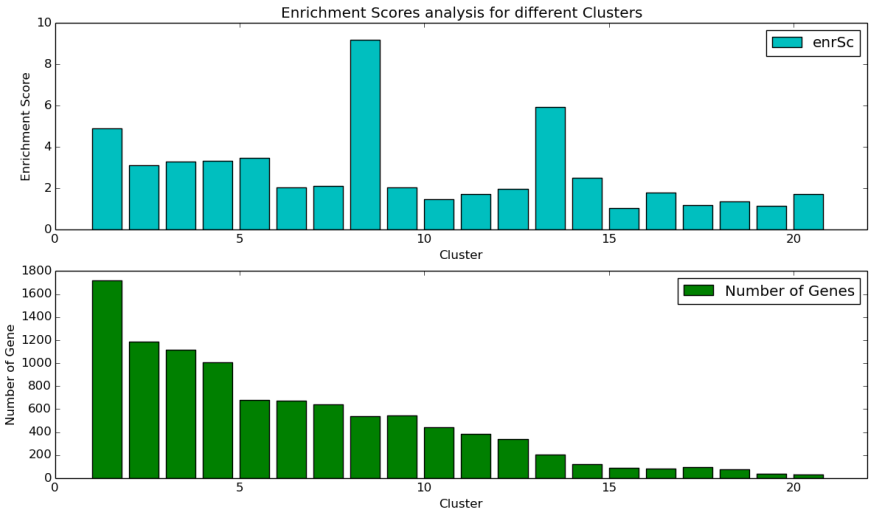


Fig. 5.7 Enrichment scores analysis without *coregionalization*. Figure at the top shows the enrichment score for different clusters, where $x$-axis is the cluster number and $y$-axis shows the enrichment score of that cluster. Figure at the bottom shows the number of genes belongs to any specific cluster.

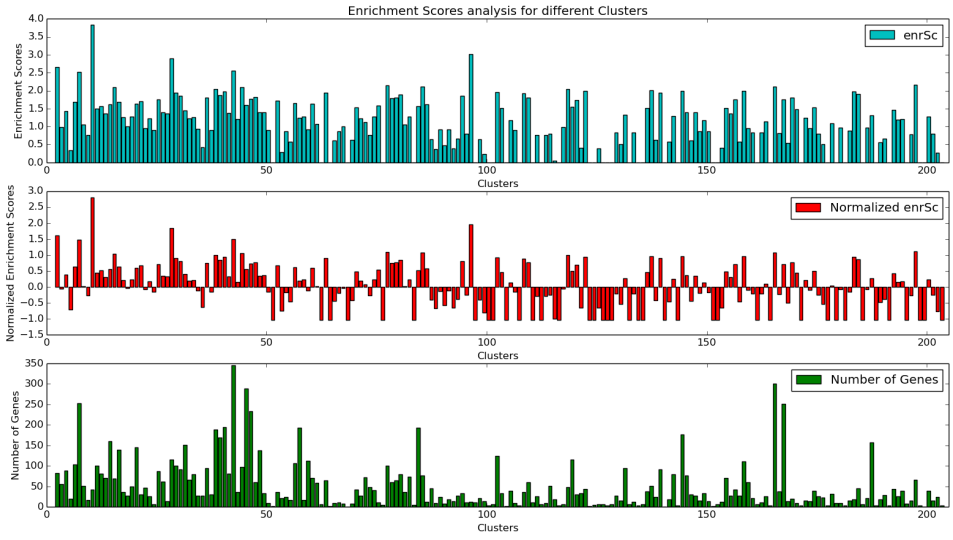


Fig. 5.8 Enrichment scores analysis. Figure at the top shows the enrichment score for different clusters where $x$-axis is the cluster number and $y$-axis shows the enrichment score of that cluster. Figure located at the middle shows the normalized score (mean enrichment score is 1.05). While the bottom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

reported about 68 Bioinformatics tools to compute the enrichment score and grouped them in 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 result the results of enrichment score analysis. In a functional annotation clustering 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 are enriched or not. Here for our 203 clusters the mean enrichment score is 1.05 and we have found at least 15 clusters have an enrichment score of $\geq 2$. 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 gene have any roles for a phenotype or a given phenomena. It is also used for the analysis of gene ontology, physical interaction networks, 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 cluster (cluster197; Figure 5.6) which was selected at the earlier stage for further analysis and has a relatively high enrichment score (2.16) is related with ALS. In previous study Brockington et al. (2013) reported about $SOD1$ related genes and ALS. One of the $SOD1$ gene, $Derlin-1$ (Prob ID: 1415693_at), can accumulate with other misfolded proteins and cause the neuron death and belongs to our chosen cluster. Figure 5.9 shows the pathway analysis that we have found for one of our cluster using tool $DAVID$. We have also found some other genes from the same cluster are responsible for cell death and related with some other neural disorder.

## 5.4 Discussion

We have performed genome-wide analysis to cluster genes systematically and analyse the rationale behind the variation in the speed of propagation for ALS. Our particular

| **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.

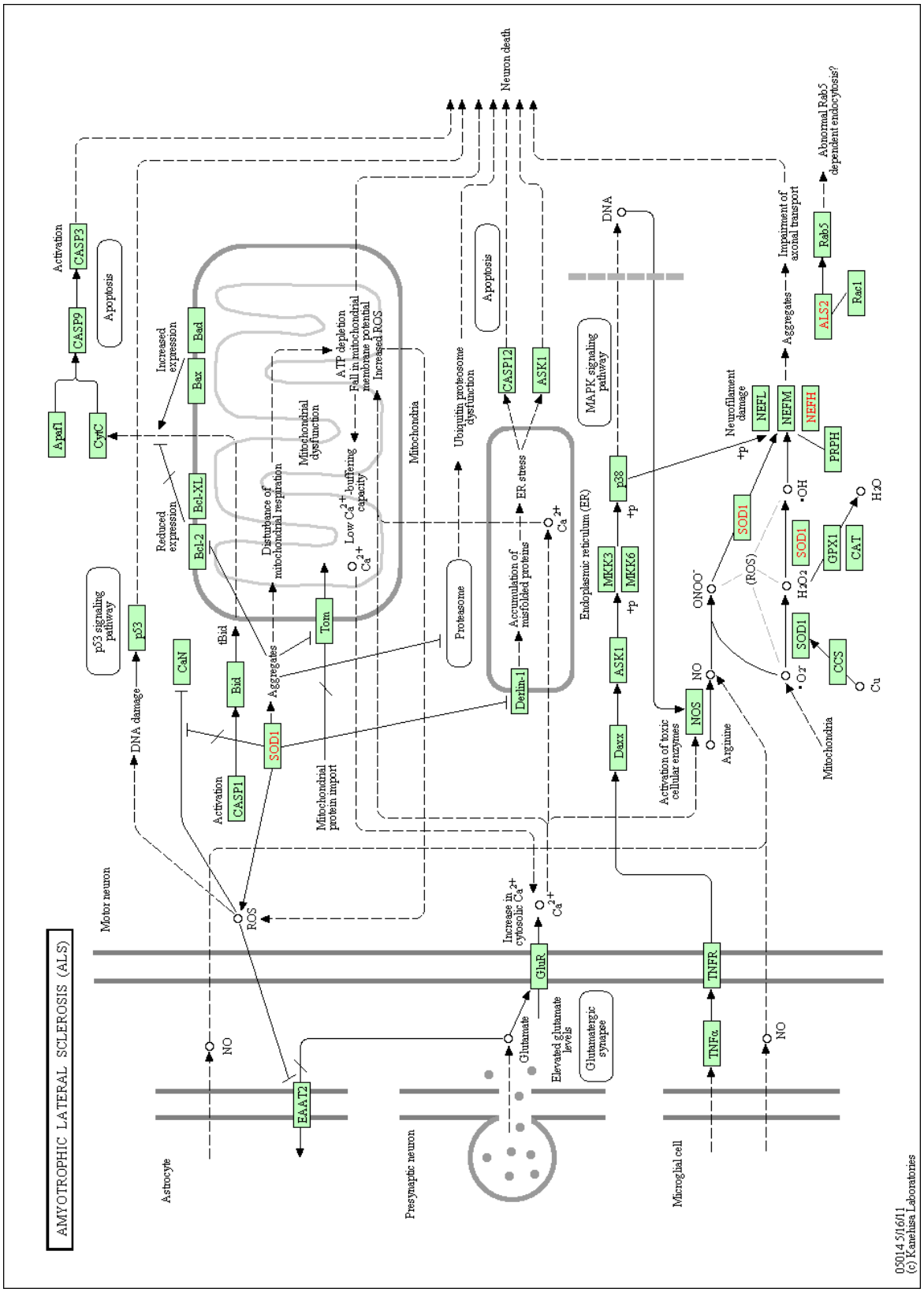


Fig. 5.9 Pathway analysis: KEGG_Pathway.

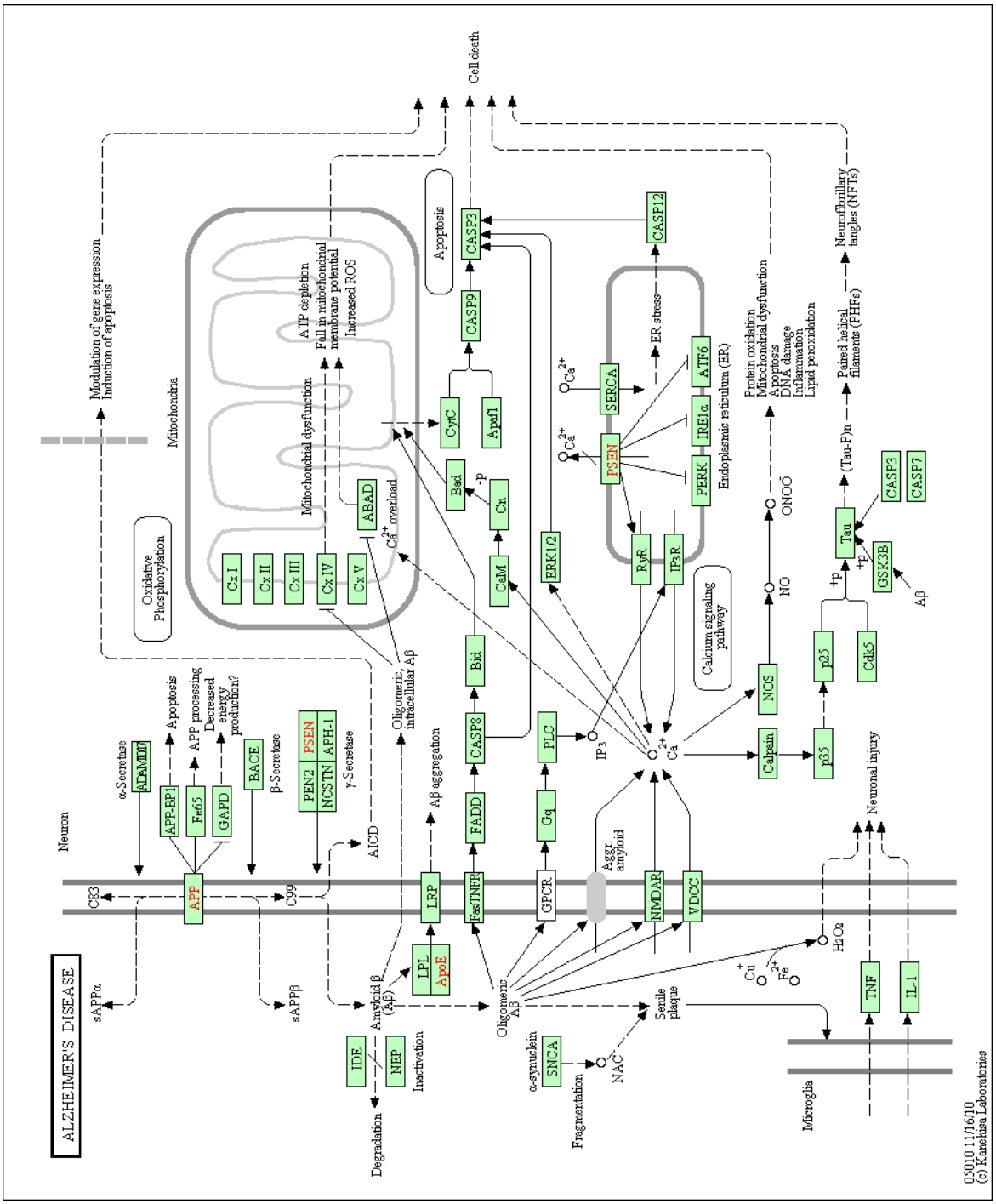


Fig. 5.10 KEGG_Pathway analysis of Alzheimer's disease.

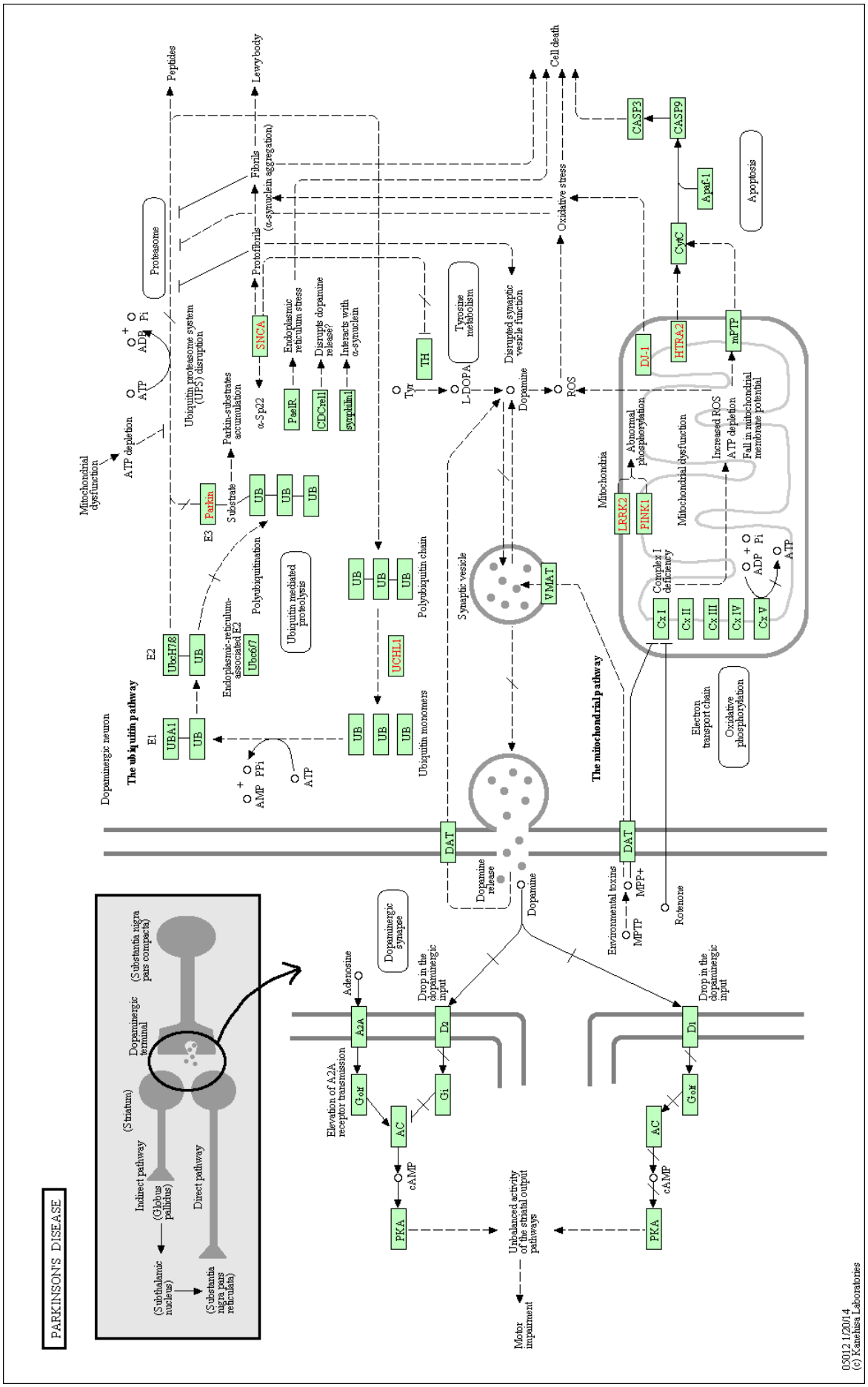


Fig. 5.11 KEGG_Pathway analysis of Parkinson's disease.

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 disease process. Here we started with whole genome set and ended with a single gene. This finding lead us to conclude that the model we have developed based on 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 is required.

# Chapter 6

# Conclusion and Future work

We have developed a tool based on programming language *R* named *chipDyno* using the model of Sanguinetti et al. (2006) which integrate the connectivity information between genes and transcription factors, and micro array 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 manage to determine the gene specific transcription factor activity for *C. elegans*, a multicellular eukaryote. We were also successful to filter out the quiet genes from the differentially expressed genes.

To elucidate pathways and processes relevant to human biology and disease *C. elegans* is been 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 were 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 great deal of similarity to human like mammalian genes. So, the biological insight acquire from *C. elegans* may be directly applicable to more complex organism like 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 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. Gaussian process is not restricted to equally spaced time series data. Already

Gaussian process regression have 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.

## 6.1 Future Work

Sanguinetti et al. (2006) model to infer the transcription factor activity is a linear-Gaussian state-space model. We believe that this linear Gaussian model is equivalent to Gaussian process with a specific covariance function. We have developed a model directly from Gaussian process to achieve the same goal. We are quite close to develop a valid covariance function for reconstructing transcription factor activities given gene expression profile and binding information between genes and transcription factors. Here we will introduce a computational trick using singular value decomposition and intrinsic coregionalization model. We believe this method will enable us to efficiently fit the Gaussian process in a reduced transcription factor activity space.

Amyotrophic lateral sclerosis (ALS), also known as "Lou Gehrig's Disease" or motor neurone disease (MND), is an irreversible progressive neurodegenerative adult onset that affects motor neurons in the brain and the spinal cord. Muscle denervation spreads over neuromuscular system and leads toward death by failure of the respiratory system with in few years of system onset (Peviani et al. (2010)). This lethal invariable disorder has median survival of less than 5 years, only 20% of the affected people can survive more than 5 years and 10% of the patients can survive more than 10 years. Mutation in the Cu/Zn superoxide dismutase (SOD1) gene is responsible for around 20% of the familial motor neurone disease Nardo et al. (2013). Transgenic mice can express human SOD1 mutation and nicely replicates different histopathological and clinical features of motor neurone disease. Mimic of these murine models of different clinical phenotypes observed from human MND patients are widely using by the researchers to determine the disease progression. But we didn't found any evidence of gene expression analysis that attempt to analyse motor neuron disease considering the genetic background on different phenotype of this murine disease. So gene expression analysis for different murine models could reveal interesting information. Nardo et al. (2013) used two mouse models to analyse fast and slow disease progression of ALS. We will use our Gaussian process based model to infer the transcription factor activity on gene expression data obtained from different murine models and try to find out some fascinating insights.

Clustering of gene expression time series is another major interest of the research to get the view of groups of co-regulated or associated genes. It is assumed that gene

involved in the same biological process will be expressed with a similarity sharing underlying time series. Cossins et al. (2007) did some additional cluster analysis (not published yet!) based on some phenotype properties. Again it is very common to have multiple biological replicates of the gene expression time series data. Just taking average of the replicates surely lead toward discarding insight. Recently Hensman et al. (2013b) used a hierarchy of Gaussian process to model a gene specific and replicate specific temporal covariance. They also used this model for clustering application. Using this Gaussian process based hierarchical clustering analysis of Hensman et al. (2013b) we will try to find some robust clusters for the gene expression data of *C. elegans*. Once if we can do so, it will easily lead us to find out the active transcription factors related with these clusters and their subsequent dynamic behaviour as well.

# References


Abramowitz, M. and Stegun, I. A. (1965). *Handbook of Mathematical Functions*. Dover, New York. (page 48)

Alon, U. (2006). An introduction to systems biology: design principles of biological circuits. *CRC Press*. (pages 11 and 13)

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 23)

Alvarez, M. A. and Lawrence, N. D. (2011). Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, (12):1459–1500. (pages 76 and 78)

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 62 and 76)

Authors, T. G. (2012–2014). GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy. (pages 52, 60, 71, and 77)

Barenco, M., Tomescu, D., Brewer, D., Callard, R., Stark, J., and Hubankcorresponding, M. (2006). Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(3):R25. (page 73)

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 72)

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press. (page 41)

Bishop, C. M. (1999). Latent variable models. In *Learning in Graphical Models*, pages 371–403. MIT Press. (page 19)

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York. (page 42)

Bonilla, E. V., Agakov, F. V., and Williams, C. K. I. (2007). Kernel multi-task learning using task-specific features. (page 76)

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). (pages 23 and 24)

Brenner, S. (1974). The genetics of caenorhabditis elegans. *Genetics*, 77:71–94. (pages 8, 9, and 27)

Brivanlou, A. H. and Darnell, J. E. (2002). Transcription signal transduction and the control of gene expression. *Science*, 295(5556):819–818. (page 11)

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 13, 72, and 86)

Byerly, L., Cassada, R., and Russell, R. (1976). The life cycle of the nematode caenorhabditis elegans. i. wild-type growth and reproduction. *Dev Biol*, 51(1):23–33. (page 9)

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. *J Neurol Sci*, 165:s21–s26. (page 72)

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 36 and 95)

Deisenroth, M. P. (2012). *Efficient Reinforcement Learning using Gaussian Processes*. PhD thesis, Faculty of Informatics, Karlsruhe Institute of Technology. (page 43)

Dunson, D. D. (2010). Nonparametric bayes applications to biostatistics. *Bayesian Nonparametrics, Cambridge University press.* (page 80)

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). *Proceedings of the National Academy of Sciences of the United States of America*, 95(25):14863–14868. (page 34)

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. (page 43)

Ferraiuolo, L., Kirby, J., Grierson, A. J., Sendtner, M., and Shaw, P. J. (2011). Molecular pathways of motor neuron injury in amyotrophic lateral sclerosis. *Nat Rev Neurol*, 7(11):616–630. (page 72)

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using bayesian networks to analyze expression data. *J. Comput. Biol.*, 7(3/4):601–620. (page 7)

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 23)

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman and Hall / CRC. (page 73)

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 9)

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 72)

Hensman, J., Fusi, N., and Lawrence, N. D. (2013a). Gaussian processes for big data. *arXiv:1309.6835*. (page 60)

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 73, 75, 80, 81, and 95)

Honkela, A., Girardotb, C., Gustafsonb, H., Liub, Y.-H., Furlongb, E. E. M., Lawrencec, N. D., and Rattrayc, 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 73)

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 72 and 84)

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 72 and 86)

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 72 and 86)

Ingalls, B. P. (2012). *Mathematical Modelling in Systems Biology: An Introduction*. MIT Press. (page 3)

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 30 and 32)

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 34, 38, 73, 80, 84, and 94)

Karin, M. (1990). Too many transcription factors: positive and negative interactions. *New Biol.*, 2(2):126–131. (page 11)

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

Kitano, H. (2002). *Systems Biology: Toward System-level Understanding of Biological Systems*. MIT Press. (pages 2, 4, 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 41)

Latchman, D. S. (1997). Transcription factors: an overview. *Int. J. Biochem. Cell Biol.*, 29(12):1305–1312. (page 11)

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816. (page 21)

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 30)

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 7 and 61)

Lee, T. I. and Young, R. A. (2000). Transcription of eukaryotic protein-coding genes. *Annu. Rev. Genet.*, 34:77–137. (page 11)

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. *PNAS*, 100(26). (pages 23, 24, and 30)

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 73)

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. (pages 42 and 44)

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 Comput Biol*, 10(5). (page 73)

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 72)

Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in Applied Probability*, 5:439–468. (page 41)

Medvedovic, M., Yeung, K., and Bumgarner, R. (2004). Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8). (page 73)

Menzefricke, U. (2000). Hierarchical modeling with gaussian processes. *Communications in Statistics - Simulation and Computation*, 29(4):1089–1108. (page 73)

Mitchell, P. J. and Tjian, R. (1989). Transcriptional regulation in mammalian cells by sequence-specific dna binding proteins. *Science*, 245(4916):371–378. (page 11)

Nachman, I., Regev, A., and Friedman, N. (2004). Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, 20(1). (page 24)

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). (pages 80 and 94)

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics No. 118, New York: Springer-Verlag. (page 41)

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 73)

Nickisch, H. and Rasmussen, C. E. (2008). Approximations for binary gaussian process classification. *Journal of Machine Learning Research*, 9(1). (page 43)

Nikolov, D. B. and Burley, S. K. (1997). Rna polymerase ii transcription initiation: A structural view. *Proc. Natl. Acad. Sci. U.S.A.*, 94(1):15–22. (page 11)

O'Hagan, A. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society B*, 40(1):1–42. (page 41)

Ong, I. M., Glasner, J. D., and Page, D. (2002). Modelling regulatory pathways in e. coli from time series expression profiles. *Oxford Univ Press*, 18(1):S241–S248. (page 7)

Palikaras, K. and Tavernarakis, N. (2013). Caenorhabditis elegans (nematode). (1):404–408. (page 9)

Papoulis, A. (1991). *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Companies, 3rd edition. (page 41)

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 28 and 80)

Pe'er, D. (2003). *From gene expression to molecular pathways*. PhD thesis, The Hebrew University. (page 36)

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 Neurol Disord Drug Targets*, 9(4):491–503. (pages 13, 72, and 94)

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 13 and 72)

Ptashne, M. and Gann, A. (1997). Transcriptional activation by recruitment. *Nature*, 386:569–577. (page 11)

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. (pages 41, 43, 45, 46, and 52)

Roeder, R. G. (1996). The role of general initiation factors in transcription by rna polymerase ii. *Trends Biochem. Sci.*, 21(9):327–335. (page 11)

Rogers, S. and Girolami, M. (2011). *A First Course in Machine Learning*. Chapman & Hall/CRC, 1st edition. (page 42)

Roweis, S. (1998). Em algorithms for pca and spca. In *in Advances in Neural Information Processing Systems*, pages 626–632. MIT Press. (page 20)

Rowley, J. (2007). The wisdom hierarchy: Representations of the dikw hierarchy. *J. Inf. Sci.*, 33(2):163–180. (page 19)

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 72)

Sanguinetti, G., Rattray, M., and Lawrence1, 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 7, 24, 25, 27, 38, 57, 61, 65, 73, 93, and 94)

Shaye, D. D. and Greenwald, I. (2011). Ortholist: A compendium of c. elegans genes with human orthologs. *PLoS ONE*, 9(1). (pages 9 and 93)

Silva, R., Glymour, C., and Spirtes, P. (2005). Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7:2006. (page 21)

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 27, 61, and 73)

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 64)

Sulston, J. E. and Horvitz, H. (1977). Post-embryonic cell lineage of the nematode, caenorhabditis elegans. *Developmental Biology*, 56(1):110–156. (page 9)

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 9)

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. (pages 20 and 21)

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 73)

Turner, B. J. and Talbot, K. (2008). Transgenics, toxicity and therapeutics in rodent models of mutant sod1-mediated familial als. *Prog Neurobiol*, 85(1):94–134. (page 72)

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Phys. Rev.*, 36:823–841. (pages 49 and 58)

Wackernagel, H. (2003). *Multivariate Geostatistics An Introduction with Applications*. Springer Science and Business Media. (pages 62, 76, and 78)

Watson, J. D. and Crick, F. H. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, (171):737–738. (page 1)

Wiener, N. (1949). *Extrapolation, Interpolation and Smoothing of Stationary Time Series.* (page 41)

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 20)

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 43)

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 41)

Wood, W. B. (1988). The nematode c. elegans. *Cold Spring Harbor Laboratory Press, New York*, pages 1–16. (page 9)

WormNet (2015). Wormnet. http://www.functionalnet.org/wormnet/about.html [Online; accessed 01-Sept-2015]. (pages 30, 31, and 32)

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 30)

# Appendix A

# Apendix 1

## A.1 GP property

## A.2 SVD

## A.3 Markov property

## A.4 ChipDyno ?