Gaussian Process for Modelling Transcription Factor Activity and Clustering Gene Expression

Muhammad Arifur Rahman

Department of Computer Science

University of Sheffield

This dissertation is submitted for the degree of

Doctor of Philosophy

August 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 principle and detail of biological systems. More than six decade ago, Watson and Crick discover 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 help to explain different aspect of biology like heredity, different disease, 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, translation, and so on. 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 isn't sufficient to understand biological systems as systems. Genes and protein are few components of a whole system. It is necessary to understand what constitute the system, but even only this knowledge is not sufficient

to understand the complete system. System biology is a new field of biology 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 technique may be may be required for each individual research target. Very often it demands combined effort from multiple discipline 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 carried out the 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. The abstract concept of system yet more than a collection of multi-disciplinary research components. To obtain the proper insight of system beside the detail description of the components it is also essential to know what happens during the period or processes when any stimuli and/or disruptions take place.

Identification of the system structure is the primary requirement to understand biological system. Some of the key structure might be different regulatory relationship of genes and interactions with protein that shows 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 method can be used. For example, if 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 response 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 leads toward the establishment of technologies those allow us to design biological system which can provide cures for different diseases. Even some futuristic example could be organ cloning technique for the treatment of diseases what require

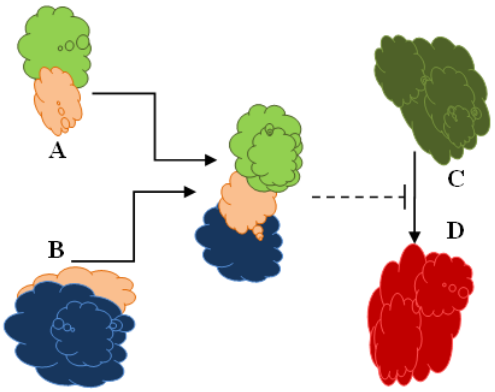

Fig. 1.1 A 'cartoon' model of protein protein interaction. Two different molecular species A and B bind to form a complex molecular. The newly formed complex hinder the rate at which molecules of species C are transformed to species D.

organ transplants or 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 of specific aspects of the objects for certain kind of study. usually other aspects are abstracted away. Biologists 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 model 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 even 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 grows. Sometimes it become 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, but in many other cases binding and unbinding rates might be also required. The cartoon model or traditional knowledge cannot provide a quantitative description rather 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 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 language are used for this simulation purposes.

As mathematical model is a hypothesis, so the outcome or result of the model hypothesis are also hypothesis. Though the real cellular behaviour definitively cannot predict by simulation, but they 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 model or system.

1.3 The Systeome Project

"Systeome" is an collection of system profile for all genetic variations and environmental stimuli response. 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 practice. This Systeome project involves many other major engineering projects for developing the measurements, as well as software platform.

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 nonlinearly to generate coherent instead of complex 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 etc. are also the systems of components to consider and 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, but the detail functional insights of biological systems may not give 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 contains thousands of genes. These genes codes for one or more protein. Expression of these genes are regulated by many of these protein through a very complex regulatory pathway. Usually this regulation occurred to accommodate the change of the environment, as well as through the cell cycle of the development process. Gene expression is the process where information contains in the gene is used to synthesis a functional gene product. The genetic code stored in the DNA usually expressed or interpreted by gene expression which represents the phenotype. These gene expression data are 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 contains 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 schematic 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 construct 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 controlled sample and another is test sample. After extracting mRNA from DNA, copied are made from mRNA by reverse transcription. Two different fluorescent tagged with cyanide are usually used to differentiate between controlled samples and test samples. In general, green is used for controlled 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 controlled targets only, a red spot indicates hybridisation with the test targets only, yellow represent hybridisation both with the controlled targets and test targets, and 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 also in the field of 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 tried 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 measurement and analysis of the mechanisms regulating mRNA transcription. Though using high throughput techniques it is comparatively easier to measure the output of transcription, but 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

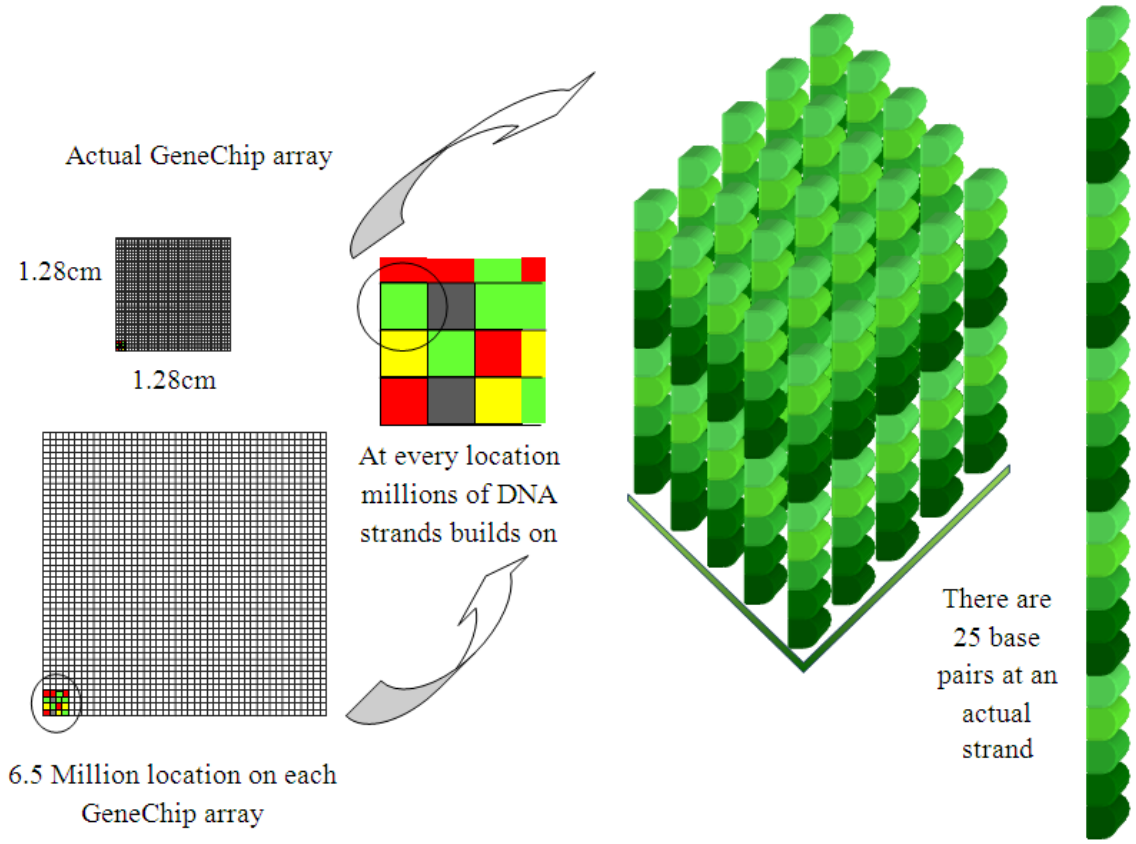

Fig. 1.3 Gene Expression Data: Affymetrix Micro Array

regulatory network. Lee et al. (2002) determined the transcriptional regulatory network of yeast using chromatin immunoprecipitation(ChIP). They tried to figure out how yeast transcriptional regulators bind to promoter sequences across the genome. By calculating a confidence value (P value) and setting up specific threshold they considered 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

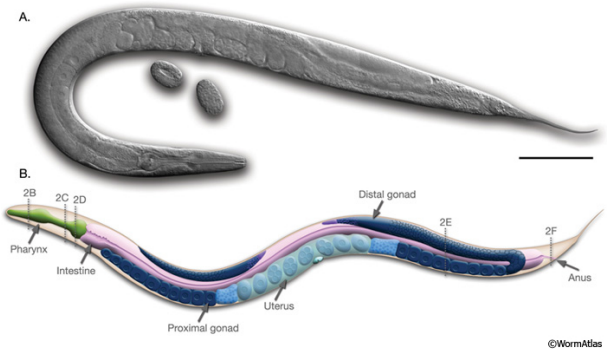

Fig. 1.4 Anatomy of an adult hermaphrodite(*C.elegans*). A. DIC image of an adult hermaphrodite, left lateral side. Scale bar 0.1 mm. B. Schematic drawing of anatomical structures, left lateral side (Courtesy of WormAtlas http://www.wormatlas.org/hermaphrodite/introduction/IMAGES/introfig1leg.htm).

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)).

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. Its an eukaryote and it shares cellular and molecular structures and control pathways with higher organism. *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)). Its developmental process like embryogenesis, morphogenesis goes through a complex process to growth to an adult. Yet monitoring of the cellular process and recording of cell division pattern is comparatively easier as its body is transparent. C Elegan's complete cell lineage at the electron microscopy level has been completed. Its already been established and this 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 is using 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 were resulted in that list and this represents $\sim 38\%$ of the 20,250

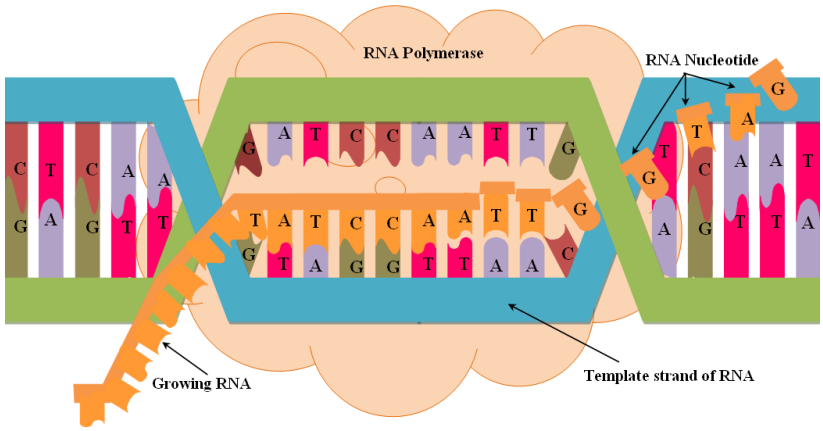

Fig. 1.5 A 'cartoon model' of DNA Transcription

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

1.4.3 Transcription

All the process in the body take places through some proteins. So, cells needs protein continuously. On the consequences, protein are required to be manufactured at every moment in the body. Inside cell protein is manufactured from the DNA. When any cells in the body is in need of protein, a special signal is sent to the DNA using hormones. Then proteins reside in DNA start to manufacture based on the received codes. The way that the enzymes finds 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 the 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. An RNA polymerase read the sequence of DNA, which produce an 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 binds 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 factors called transcription factors then 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 transcribe 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 protein such as coactivators, deacetylases, chromatin remodelers, kinases, histone acetylases, and methylases also play crucial roles in gene regulation but due to lack of DNA-binding domains they are not classified as transcription factors (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)).

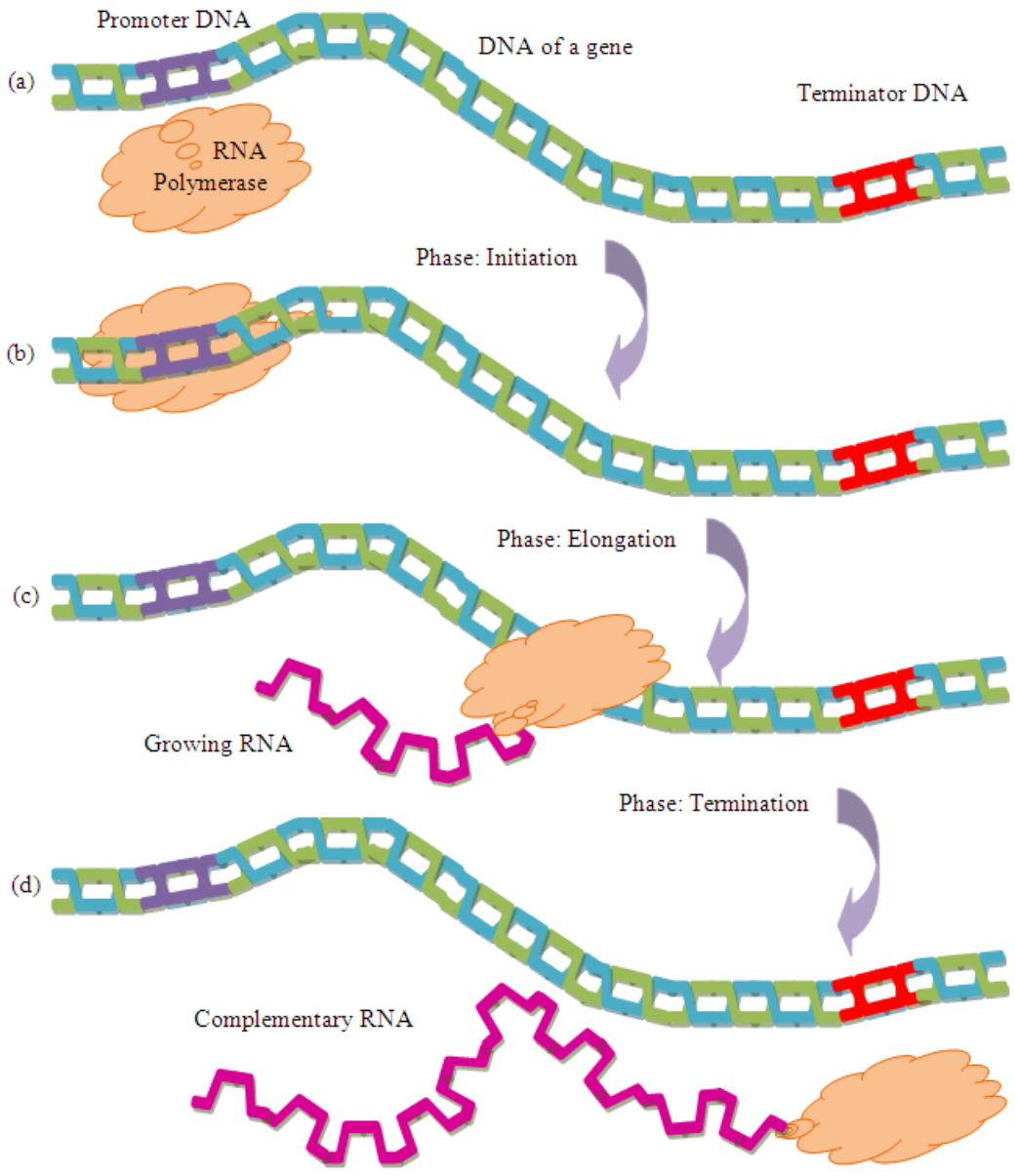

Fig. 1.6 A 'cartoon model' of Transcription Process: DNA Transcribed in mRNA

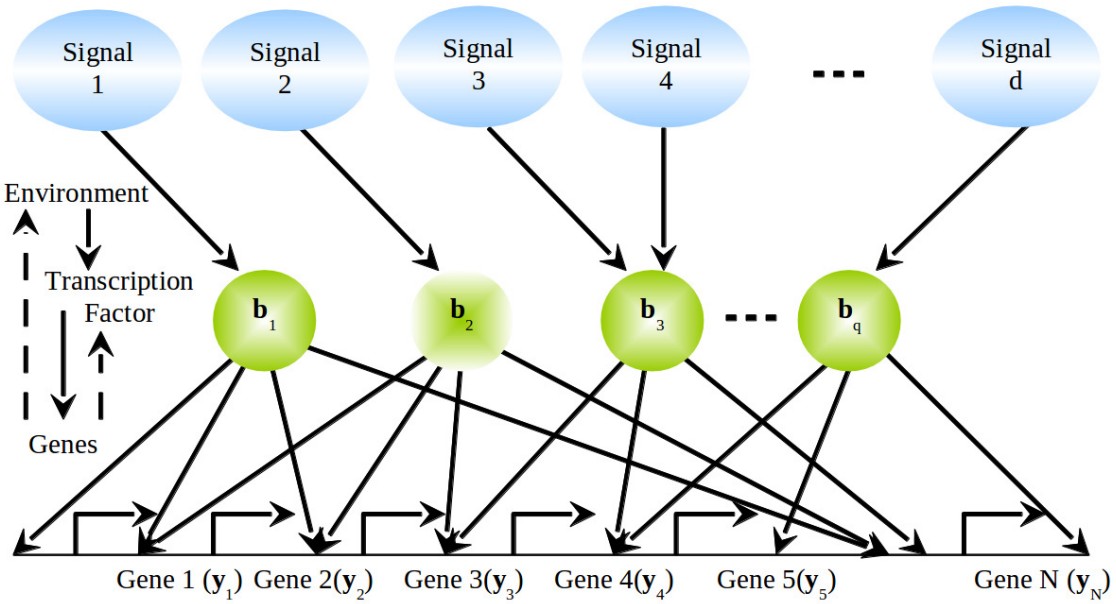

Fig. 1.7 The mapping between environmental signal, transcription factor inside the cell the genes that they regulate (Alon (2006)).

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 view point 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 every observations in the input space are random variables from a Gaussian distribution. We included the introductory concepts of Gaussian process at Chapter 3.

1.6 Summary of the main goal of the thesis

We will set some generic goals for this thesis and later we will try to achieve them chronologically:

Generic goal 1: We will aim to develop a tool (with programming language R) using probabilistic model for transcription factor activities. To step forward from a unicellular microorganism (e.g. yeast) to a multicellular eukaryote (e.g. *C.elegans*) we will build the connectivity information and analyse the transcription factor activities using our tool.

Generic goal 2: We will target to find an pathway to overcome the limitation of the parametric Markovian assumption based linear Gaussian model to non-parametric Gaussian process with a particular covariance function.

Generic goal 3: We will design a kernel or covariance function of Gaussian process to model the dynamic behaviour of transcription factors for given gene expression time series data and connectivity information between genes and transcription factors.

Generic goal 4: Our final goal of this thesis is to develop a new clustering method based on hierarchy of Gaussian processes to model condition-specific and gene-specific temporal covariances which allows each cluster to be parametrised according to whether the behaviour of the genes across conditions is correlated or anti-correlated.

1.7 Road Map

The thesis is structured in the following chapters:

Chapter 1: This documents starts with some basic concepts and general terminology to the field of interest to addressed 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 was solved for a unicellular microorganism (yeast) but we have forwarded

the mathematical model of transcription factors activates 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. Choosing 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 are 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 is 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.

1.8 Notation, Symbols and Acronyms

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

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 contextualize 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 playing 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 likely to suffer from quality issue for various reasons. Even in the controlled environment due to several reasons acquisition errors might be included. In the noisy environment it might be even more. Dealing with these noise or added uncertainty of the data is troublesome. In a particular context within the constraints probabilistic modelling turns out to be a dominant approach with added flexibility and capability of dealing with uncertainty in many forms.

2.1 Latent Variable Model

Latent variable models (LVMs) (Bishop (1999)) explain complex relations between multiple variables by simple relations between the variables and an underlying unobservable, i.e. latent structure. Latent variable are typically included in statistical model for different statistical concepts, including representation the unobservable factors/covariates, missing data, random effects, finite mixtures, variations in hierarchical data, and clusters and many more. Figure 2.2 shows an analogy of latent variable model where Marionette's different movement and dynamics are observed whereas

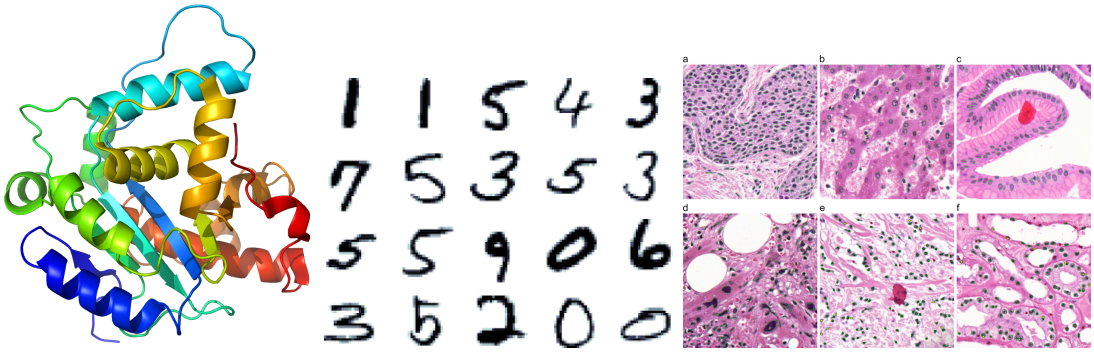

Fig. 2.1 Examples of high dimensional data form types and nature. Left: 3D model of a protein structure. Centre: Multiple samples of hand written digits from MNIST dataset. available at http://yann.lecun.com/exdb/mnist/. Right: Multiple image patch from breast cancer, liver, gastric mucosa, bone marrow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

these movements are taking place or controlled by the 'control bar'. The dynamics of the control bar is usually unobserved.

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

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

$$y_{i,j} = w_j(\mathbf{x}_i) + \epsilon_i, \tag{2.2}$$

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

Given the prior distribution over the latent variables has a Gaussian distribution, then the precision β can be infinity and PCA is recovered in the limit. The conditional

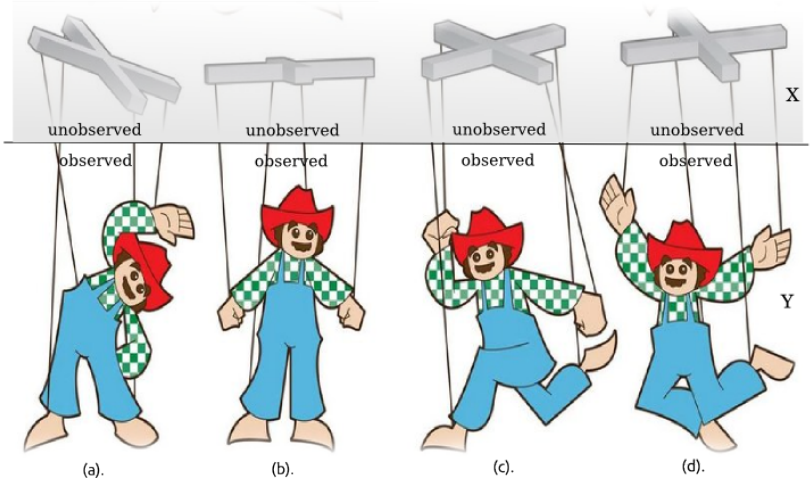

Fig. 2.2 Marionette Analogy of Latent Variable Model: Marionette's different dynamics are observed (represented by $\mathbf{Y}$). But these movements are controlled by the control bar which is unobserved (dynamics represented by $\mathbf{X}$).

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:

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

Even for finite precision β the maximum likelihood solution for $\mathbf{W}$ spans the principal sub-space of the data (Tipping and Bishop (1999)). This approach 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 this latent variables is first they are marginalised then other parameters are optimized using maximum likelihood model. Lawrence (2005) used an alternative but similar approach by first marginalising the parameters and then optimized the latent variables.

2.2 Modelling of TFAs

In recent years most methods 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. Many of the researcher used different ways or algorithm to find out these TFAs. For example, Liao et al. (2003) developed a data decomposition technique with dimension reduction and introduced '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 dose 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. But it is expected that depending on the experimental conditions transcription factor activities can vary from gene to gene. Even it is also expected that different transcription factors may bind the same gene. In most of the cases, realistic information about the intervals may not be true as they were not based on 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 using discrete random variables for protein concentrations and binding affinities. Though the model was more realistic but the computational complexity for genome-wide analysis can be expensive.

2.3 Our Goal

In this chapter we will build a dynamic 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 for a multicellular eukaryote (*C.elegans*). By nature this model will be a latent variable model which will

be developed based on probabilistic approach. We will build a tool using programming language *R* and it will be available on GitHub[1]. At Chapter:4 we will model the temporal changes in the gene-specific TFAs from time-series gene expression data using Gaussian process (a stochastic process whose consciousness comes from random values 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; Chapter:3 contains detail explanation). The covariance structure of the transcription factors will be shared among all genes. This approach will lead to a manageable parameter space and figure out useful information about the correlation of TFAs.

2.4 Probabilistic TFAs

We have developed our *R* based tools *chipDyno* based on the probabilistic approach of Sanguinetti et al. (2006). First we will give a brief introduction of that approach then we will present results.

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. 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; element (i, j) of $\mathbf{X}$ is one if transcription factor j can bind gene i, zero otherwise.

In Sanguinetti et al. (2006), TFAs are obtained by regressing the gene expressions using the connectivity information, giving the following linear model-

$$\mathbf{y}_n = \mathbf{B}_n\mathbf{x}_n + \boldsymbol{\epsilon_n} \tag{2.5}$$

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. But through marginalization by prior distribution on the rows of $\mathbf{B}_n$ these parameters can be dealt. Two plausible assumptions for selecting the prior distribution will be helpful to determine the gene specific TFAs. Firstly, $\mathbf{b}_{nt}$ has the Markov property and hence gene specific TFA $\mathbf{b}_{nt}$ at time t depends solely on the gene specific TFA at time $(t-1)$ and the second assumption is, the prior distribution to be stationary in time.

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

To satisfy these conditions then there will be two limiting conditions of prior distributions- Assume all the $\mathbf{b}_{nt}$ are identical, then the first limiting case will take place. So that

$$\mathbf{b}_{n1} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2.6}$$

and

$$\mathbf{b}_{n(t+1)} \sim \mathcal{N}(\mathbf{b}_{nt}, \mathbf{0}) \tag{2.7}$$

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

$$\mathbf{b}_{nt} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2.8}$$

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:2.6, Equation:2.7 and Equation:2.8)

$$\mathbf{b}_{n(t+1)} \sim \mathcal{N}(\gamma \mathbf{b}_{nt} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}) \tag{2.9}$$

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

$$p\left(\mathbf{Y}|\mathbf{B},\mathbf{X}\right) = \prod_{n=1}^{N} p\left(\mathbf{y}_n|\mathbf{B}_n, \mathbf{x}_n\right). \tag{2.10}$$

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

$$p\left(\mathbf{y}_n|\mathbf{B}_n, \mathbf{x}_n\right) = \mathcal{N}\left(\mathbf{y}_n|\mathbf{B}_n\mathbf{x}_n, \boldsymbol{\sigma}^2\mathbf{I}\right). \tag{2.11}$$

Factorizing the likelihood along the experiments with the assumption of spherical Gaussian noise distribution we can rewrite the Equation 2.10 as

$$p\left(\mathbf{Y}|\mathbf{B}, \mathbf{X}\right) = \prod_{t=1}^{d}\prod_{n=1}^{N} p\left(\mathbf{y}_{nt}|\mathbf{b}_{nt}, \mathbf{x}_n\right) \tag{2.12}$$

where

$$p\left(\mathbf{y}_{nt}|\mathbf{b}_{nt}, \mathbf{x}_n\right) = \mathcal{N}\left(\mathbf{y}_{nt}|\mathbf{b}_{nt}^{\top}\mathbf{x}_n, \sigma^2\right). \tag{2.13}$$

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

$$p\left(\mathbf{y}_n|\sigma,\Sigma,\boldsymbol{\mu},\gamma,\mathbf{x}_n\right)=\int\prod_{t=1}^{d}d\mathbf{b}_{nt}\mathcal{N}\left(\mathbf{y}_{nt}|\mathbf{b}_{nt}^{\top}\mathbf{x}_n,\sigma^2\right)\times\left(\prod_{t=2}^{d}p\left(\mathbf{b}_{nt}|\mathbf{b}_{n(t-1)}\right)\right)\mathcal{N}\left(\mathbf{b}_{n1}|\boldsymbol{\mu}\Sigma\right). \tag{2.14}$$

TFAs can be estimated as a posteriori using Bayes's Theorem. The detail explanation is available at Sanguinetti et al. (2006).

2.5 Datasets

Sanguinetti et al. (2006) has done their experiments on yeast's cell cycle data of Spellman et al. (1998) which is a unicellular microorganism. One of our research key question was can we step forward to find out the transcription factor activities from a unicellular microorganism to multicellular eukaryote. *C. elegans* is a established multicellular eukaryotic model organism (Chapter:1 contains some introductory information about this model organism). To find out the TFA of *C. elegans* basically we had to work with three type of datasets. *i*). Gene expression time series data *ii*). Transcription Factors *iii*). Connectivity information between genes and transcription factors.

2.5.1 Gene Expression Time series data

The gene expression Affymetrix single colour GeneChip data[2] on point estimate of expression level came without estimates of uncertainty level. To extract this data we used Bioconductor[3] package *puma* (Pearson et al. (2009)). The experiments were done at five different stages (i.e. our experimental dataset will have five time points). Apart from the temperature rest of the environmental conditions were same with the target of consistent result. The experimental data was collected within one day of its adulthood at the temperature 20 °C. With the target to measure the gene response to chill exposure the temperature was reduced to 5 °C and samples were collected after one hour, then after 24 hour (1 day), then after 72 hours (3 days). For final

[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

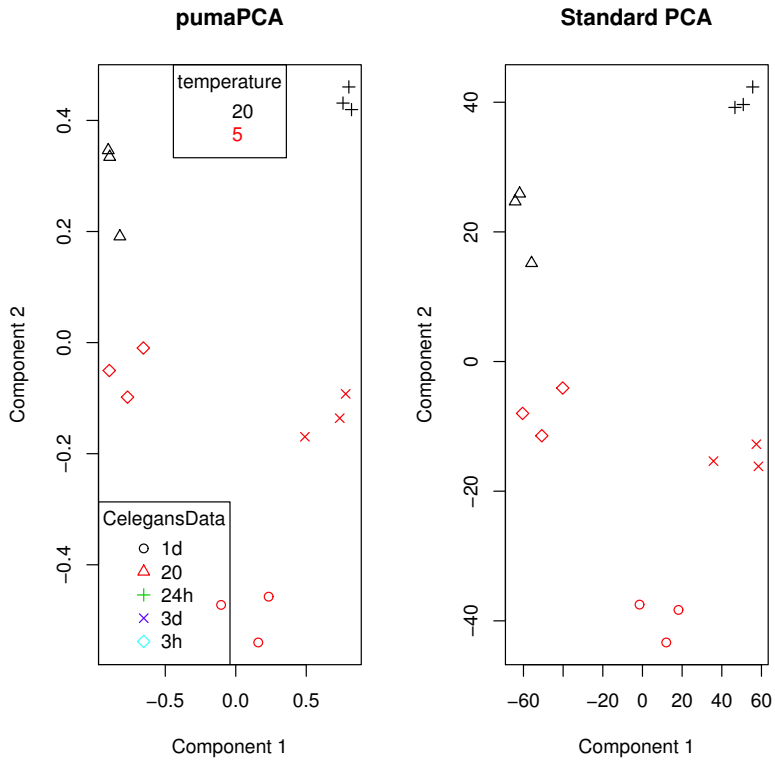

Fig. 2.3 Principal component analysis of gene expression time series data

experiments the temperature was brought back to 20 °C and samples were collected within one day of rise of temperature. All the experiments were repeated two more times which provide us 3 independent replicates of similar experiments. Figure 2.3 shows the PCA analysis of the time series data.

2.5.2 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 we also kept the opening to deal with any number of transcription factor depending on the requirement/ update of the sequence information of transcription factors.

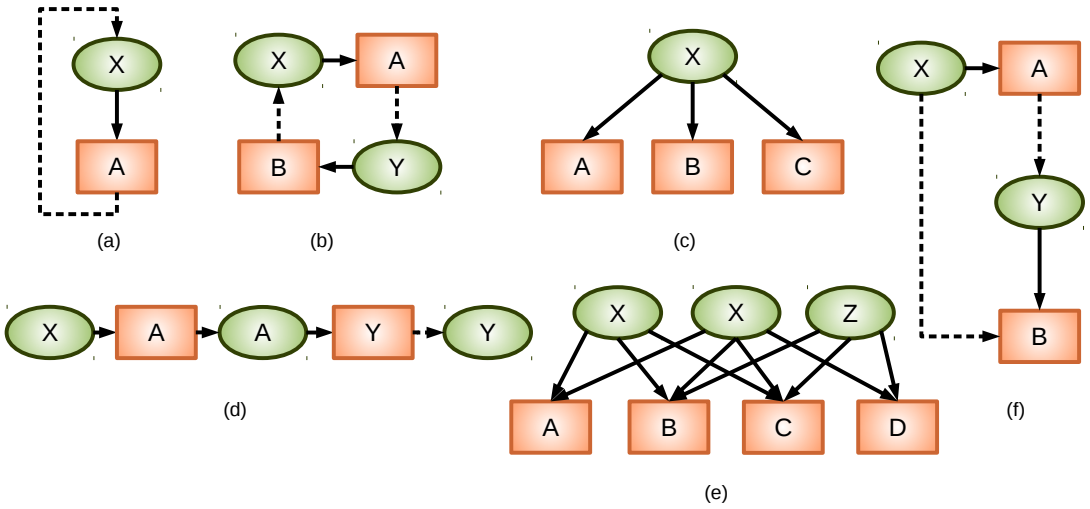

Fig. 2.4 Some basic examples of network motif: (a). autoregulation, (b). multicomponent loop, (c). single input motifs, (d). regulators chain, (e). multiple inputs, (f). feedforward motifs. Here green ovals represents regulators and orange rectangles represents gene promoters. Solid lines represent binding of a regulator to a promoter while dashed arrow represents gene encoding regulators binding their respective regulators.

2.5.3 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, regulators chain are some of the simplest and well known network motifs. Figure:2.4 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) and provide information about whether a certain transcription factor can bind the promoter region of a gene or not.

WormNet (2014) 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 following evidence codes-

- 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

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 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.

2.6 Result Analysis

We have developed a *R* based tool *chipDyno* for the identification of 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 which to compare to our results of gene specific TFA for *C. elegans*.

According to WormNet (2014) the number of gene 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, regulator chain were visible for transcription factor activity. So it was a mammoth task to choose all the transcription factors and show their activity. Rather we choose some random transcription factor and tried to find out its activity on different genes.

As a random sample we choose transcription factor ZK370.2 and find its activity on different genes. Figure 2.5 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 replications and collected the results. Later we visualize all the outcome together by some plots. From the outcome of our experimental result we can say that the dynamics for some of the gene specific regulations (i.e. F26E4.6 and T24G10.2) are very flat and not that much informative but for some genes TFAs varies notably over time (i.e. C37C3.2 and Y45F10B.3). Perhaps these are the genes which regulates significantly by this transcription factor. For some cases the error bar is quite high. These are the examples of bindings where there regulation is not that much significant or false positive could be an issue here. The magnitude of TFA also

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

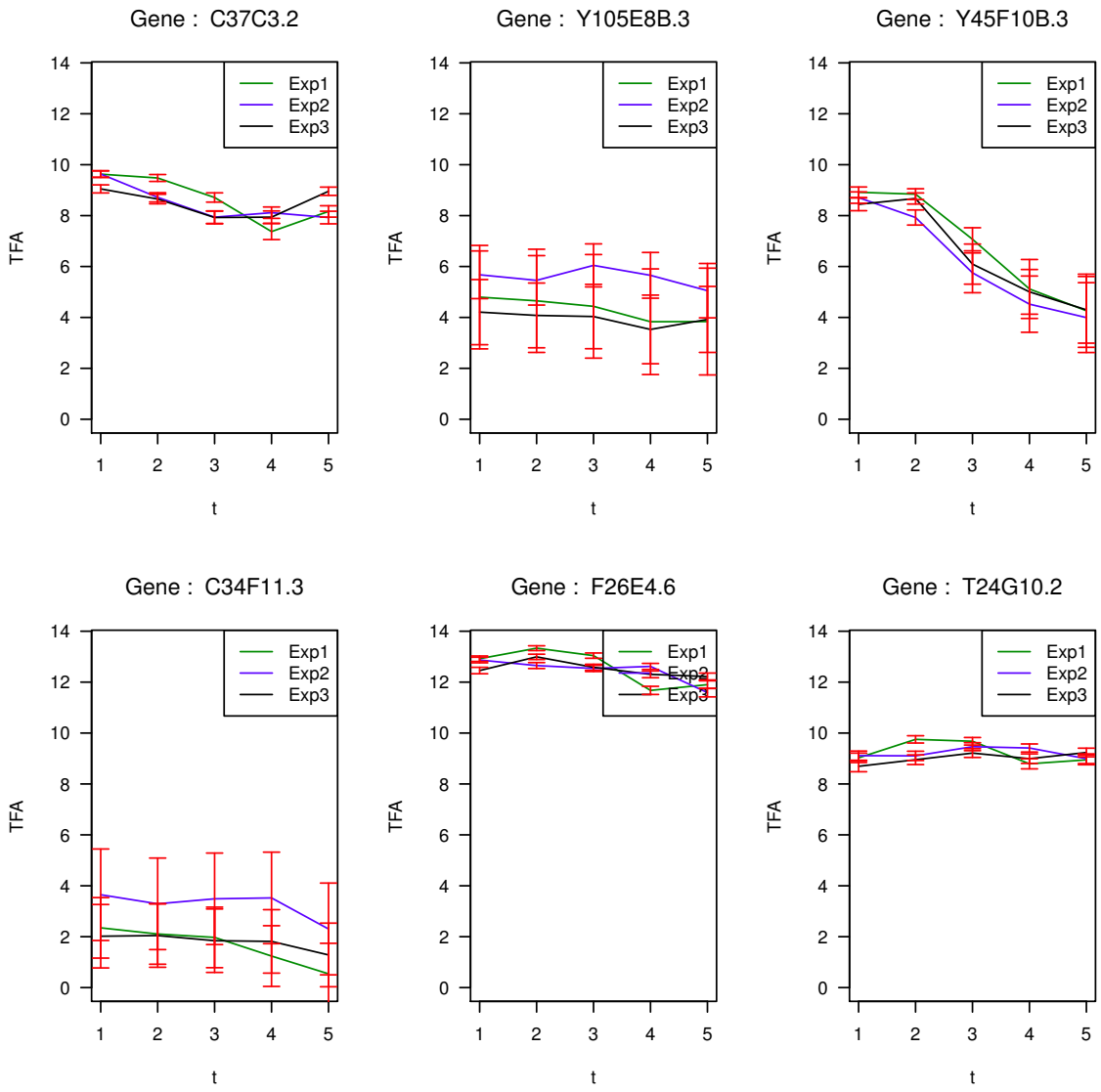

Fig. 2.5 Gene Specific transcription factor activity of ZK370.2 on (top left to right) C37C3.2, Y105E8B.3, Y45F10B.3 and (bottom left to right) C34F11.3, F26E4.6, T24G10.2. x-axis represent the time stage of the experiments and y-axis represent the gene expression level for transcription factor activities. Three different line represent TFA for three replication and red vertical lines are error bars.

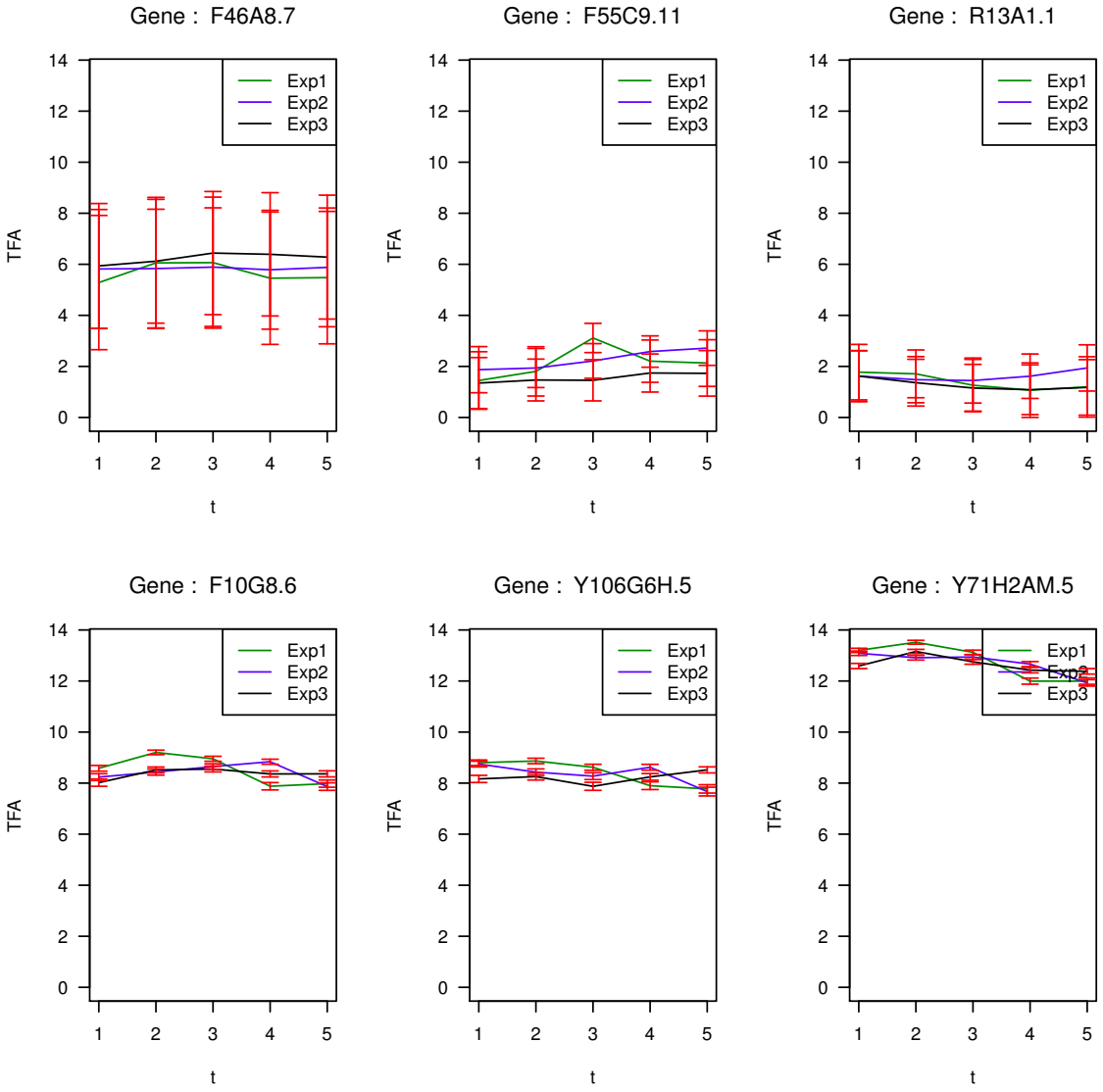

Fig. 2.6 Gene Specific transcription factor activity of T20B12.8.3 on (top left to right) F46A8.7, F55C9.11, R13A1.1 (bottom left to right) F10G8.6, Y106G6H.5 and Y71H2AM.5. x-axis represent the time stage of the experiments and y-axis represent the gene expression level for transcription factor activities. 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 ± 3.493205, F33A8.3 = 1.415785 ± 3.492985
Y105E8B.3	Y54G2A.1 = 0.07157665 ± 1.2222137 F33D11.12 = 0.03861905 ± 0.7252534 ZK370.2 = $-1.20157055 \pm 2.0318513$
Y105E8B.3	T20B12.8 = 0.25474933 ± 2.5665869 F33A8.3 = 0.11619828 ± 3.5107742 Y116A8C.35 = 0.03289664 ± 3.8071374 F11A10.2 = 0.03016348 ± 1.7737585 C16A3.7 = 0.01883489 ± 0.9431105

Table 2.1 Example of genes regulated by multiple TF

differs from one to another. We picked another random transcription factor T20B12.8.3. Figure 2.6 shows its activity on different genes.

2.6.1 Gene with multiple regulators

For the case of multi input motif a single gene could be regulated by multiple transcription factor. Our developed tool can determine the posteriori of the relative weight for the different transcription factors regulating the genes. Table 2.1 shows some examples. Gene C44B12.5 can be regulated by transcription factor Y116A8C.35 and F33A8.3. While gene Y105E8B.3 is regulated by T20B12.8, F33A8.3, Y116A8C.35, F11A10.2 and C16A3.7. Though for some cases the expression level is quite low and noise margin is significantly high (examples of binding with insignificant regulation) but we can rank these gene using Kalaitzis and Lawrence (2011).

2.6.2 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 way to visualize the data and also helps to facilitated the functional annotation of the not yet characterized genes. If an uncharacterised genes belongs a cluster then the unknown gene could possibly have similar function and may dominated by genes of same function Pe'er (2003). Cossins et al. (2007) performed some clusters analysis of the genes based on different phenotype and its subsequent

Fig. 2.7 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.

activities of the cell properties. They constructed the basic clusters with the following phenotypes 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.

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

Table 2.2 Active TF on different clusters

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 2.7 shows 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 factors active in different clusters. Table 2.2 shows the numbers active transcription factor for each clusters. For further analysis we can present the full list.

2.7 Ranking Differentially expressed gene expressions

Kalaitzis and Lawrence (2011) analysed the time series gene expression and filter the quiet or inactive genes from the differentially expressed genes. They have developed the model considering the temporal nature of data using Gaussian process. We have used this model to rank our time series gene expression and ranked the differentially expressed gene expressions. We ranked the three replicates of our data separately and later determine the Pearson correlation between ranking score of different samples.

Figure 2.8 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. Which 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.

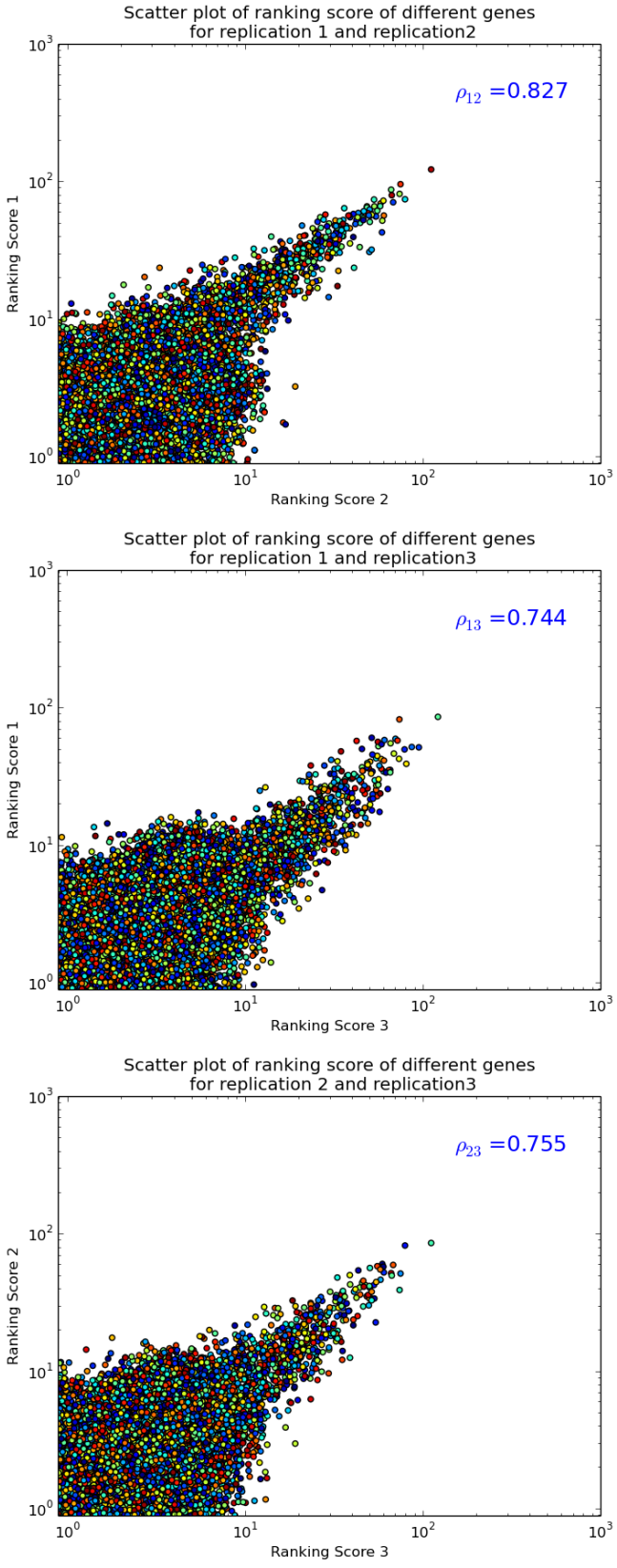

Fig. 2.8 Pearson's correlation between different ranking scores. For each figure number at the top right position represent Pearson's correlation

Chapter 3

Gaussian Process Regression

3.1 Brief History of Gaussian Process

The Gaussian processes is one of the most simple and widely used families of stochastic processes for modeling dependent data observed over time, or space, or time and space together. As a general setting, Gaussian process 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. The Brownian motion is a Gaussian process. This is because the distribution of a random vector is a linear combination of vector which have a normal distribution (Castaneda et al. (2012)). Thorvald N. Thiele was the first to propose the mathematical theory of Brownian motion. He also introduce 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 problem. For general purpose function approximators Bishop (1995) used 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. Rasmussen and Williams (2006) perhaps

the most widely used and cited article 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: to find hidden structure of unlabelled data
3. reinforcement learning: take action by maximizing the cumulative reward.

MacKay (2003), Bishop (2006) describes the concepts in detail. 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.

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 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 function is required. Then the model go through some training process and update the prior, based on the training data set $\mathcal{D}$ 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, but to model a more complex and specific learning task with improved reliability and robustness Gaussian process is a better selection. Gaussian process models can be used for regression model having an object featuring infinite dimensionality. Even at present Gaussian process have been advanced beyond the regression model and now using for classification, unsupervised learningcite, reinforcement learning cite and many more.

We assume the outputs considered at the training level may contain some noise and observed from the underlying mapping function $f(\mathbf{x})$. 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 leads to an exercise in reasoning under uncertainty. 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 from data Bayesian inference can be used to make predictions:

$$p(f|\mathcal{D}) = \frac{p(\mathcal{D}|f)p(f)}{p(\mathcal{D})} \tag{3.1}$$

The dynamic activity of transcription factors can be viewed as a regression task.

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 random variables indexed by a continuous variable: $f(\mathbf{x})$ chosen from a random function variables $\mathbf{f} = \{f_1, f_2, f_3, ..., f_N\}$, with corresponding 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 normally distributed and as a whole can be represent as a multivariate Gaussian distribution:

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

where $\boldsymbol{\mu}$ is the mean and $\mathbf{K}$ is covariance of Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$ 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 process, a Gaussian process is completely defined by its mean function and covariance function given in 3.3 and 3.4 respectively. Usually the $m(\mathbf{x})$ 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-

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

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

3.4 GP: Covariances

For convenience, we often define the mean of the prior of the GP as zero but the posterior mean of the GP $p(f|\mathcal{D})$ obtained from the GP regression is not a zero mean process.

Based on our problem we are free to design our covariance function. The mandatory requirement of a covariance matrix is symmetric positive semi-definite. So as long as the covariance function generates symmetric positive semi-definite matrix, we can use that function for a Gaussian process. Smoothness, periodicity, amplitude, lengthscale etc. are the basic properties while choosing a Gaussian process covariance function. It is very crucial to choose a appropriate function for further Gaussian process Modelling. The main goal of this thesis is to develop a covariance function able to solve our problem, hopefully more robust and flexible way. Here first we will discuss about some of the very well known and widely used covariance functions. The in detail description will be found at Rasmussen and Williams (2006).

3.4.1 Exponentiated Quadratic covariance function

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-

$$K_{EQ}(r) = a^2 \exp\left(-\frac{r^2}{2l^2}\right) \tag{3.6}$$

where $r = \|\mathbf{x} - \mathbf{x}'\|$. Here $\|\mathbf{x} - \mathbf{x}'\|$ is invariant to translation and rotation. So, 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 commonly known as hyperparameters. Parameter a determines

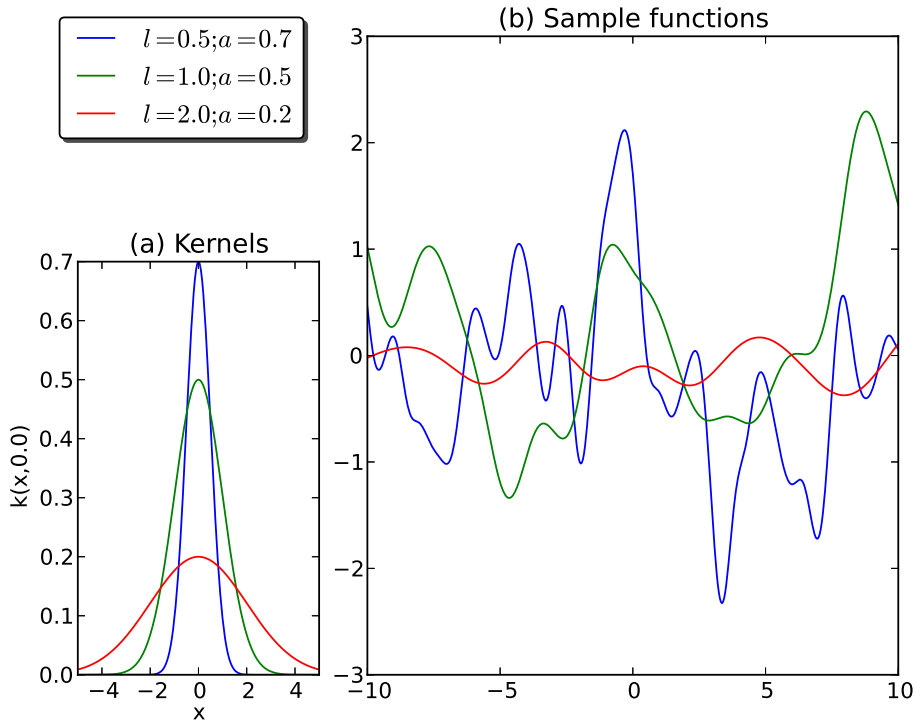

Fig. 3.1 Exponentiated Quadratic kernel and sample functions

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 Gaussian process using Exponentiated Quadratic covariance with different lengthscale and amplitude hyperparameter. The random function was generated for a given input range by drawing a sample from the multivariate Gaussian using equation 3.2 with zero mean. The smoothness of the sample function depends on the equation 3.6. 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 function could be interesting.

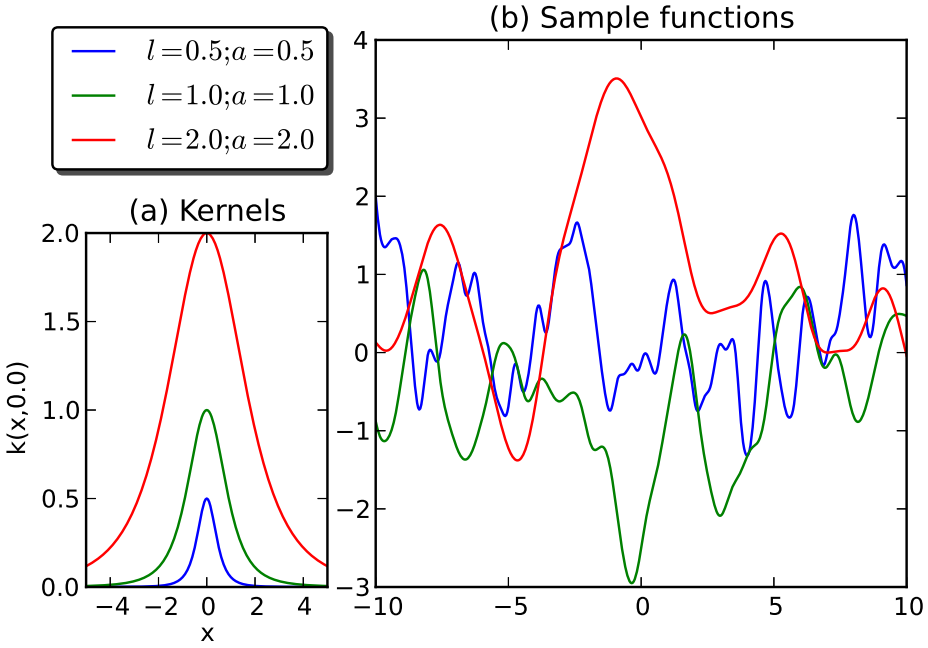

Fig. 3.2 Rational Quadratic kernel and random sample functions

3.4.2 Rational Quadratic covariance function

Rational Quadratic covariance function is equivalent to adding together multiple exponentiated quadratic covariance function having different lengthscale. Gaussian process prior kernel function expect smooth function with many lengthscale. Here the parameter α can control the relative weights for lengthscale variations. Exponentiated quadratic covariance function can be viewed as a special case of rational quadratic covariance function. If $\alpha \to \infty$, then both of the functions become identical.

$$K_{RQ}(r) = a^2 \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha} \tag{3.7}$$

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 3.8-

$$K_{Mat}(r) = a^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu} r}{l} \right) \tag{3.8}$$

where a, l, ν are positive hyperparameter, K_{ν} is a modified Bessel function and $\Gamma(.)$ is the Gamma function. Hyperparameter ν 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, but for finite value of ν the sample functions are significantly rough. The simpler form of Matérn covariance function is obtained when

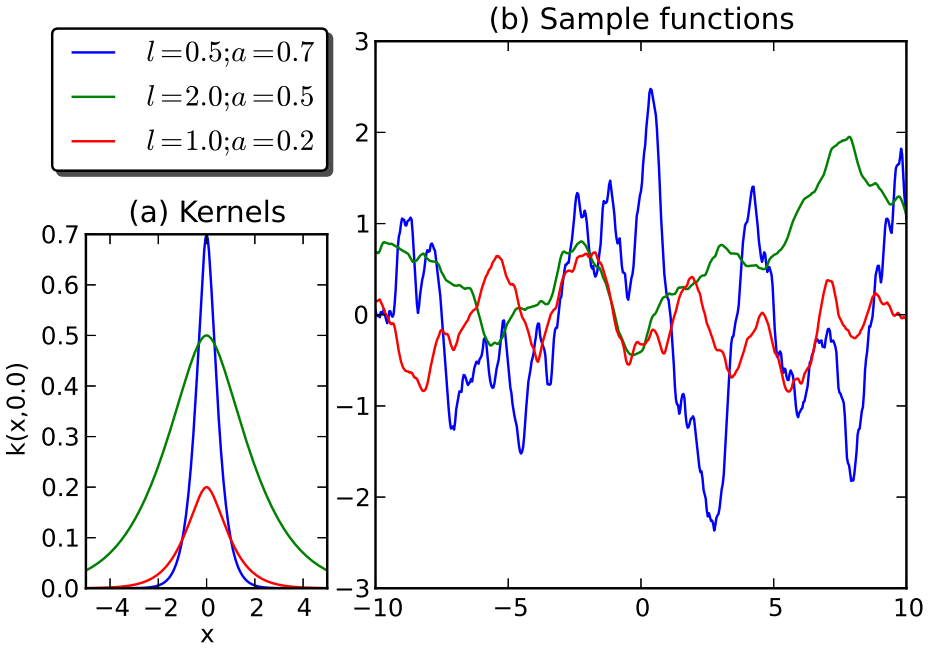

Fig. 3.3 The Matérn32 kernel and random sample functions

ν is half integer: $\nu = p + 1/2$, where p is a non-negative integer. The covariance function can be expressed as a product of an exponential and a polynomial of order p. Abramowitz and Stegun (1965) derived the general expression as follows-

$$K_{\nu=p+1/2}(r) = \exp\left(-\frac{\sqrt{2\nu} r}{l} \right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!\,(p-i)!} \left(\frac{\sqrt{8\nu} r}{l} \right)^{p-i} \tag{3.9}$$

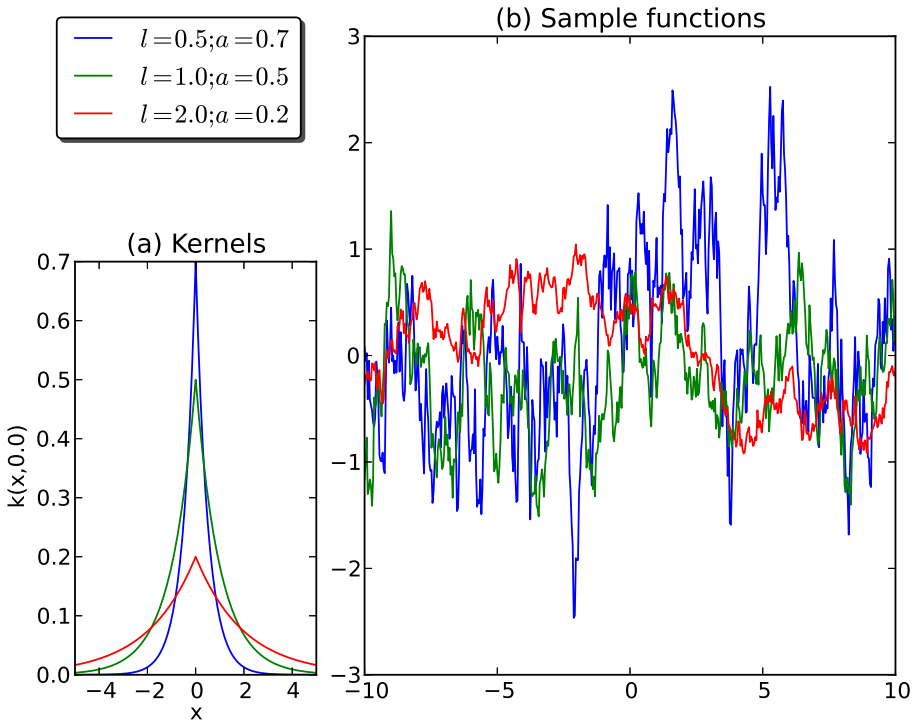

Fig. 3.4 The OU kernel and random sample functions

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

$$K_{\nu=3/2}(r) = \left(1 + \frac{\sqrt{3}r}{l}\right) \exp\left(-\frac{\sqrt{3}r}{l}\right) \tag{3.10}$$

$$K_{\nu=5/2}(r) = \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}r}{l}\right) \tag{3.11}$$

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 process was developed as a mathematical model of the velocity of a particle moving with Brownian motion. The OU process can be found setting up $\nu = 1/2$ and expressed as Equation 3.12. Figure 3.4(a) shows the kernel and Figure 3.4(b) shows the sample functions form the OU

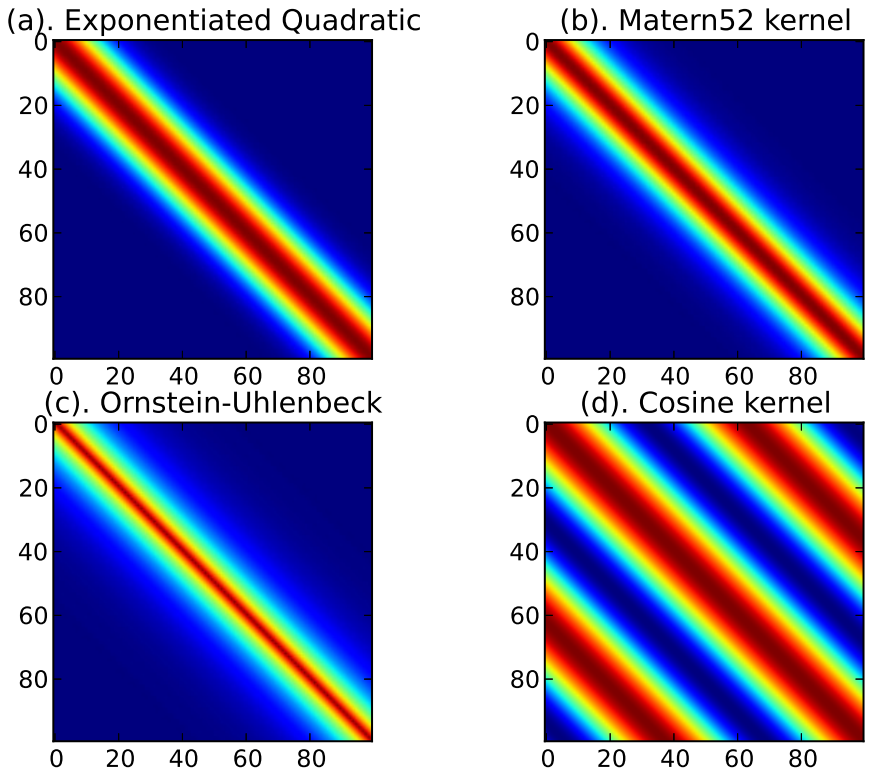

Fig. 3.5 Representation of some basic kernels (a). Exponentiated Quadratic kernel, (b). Matérn52 kernel (c). Ornstein-Uhlenbeck kernel (d). Cosine kernel

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

$$K_{\nu=1/2}(r) = \exp\left(-\frac{r}{l}\right) \tag{3.12}$$

Figure 3.5 shows examples of some basic kernels- (a). Exponentiated Quadratic kernel, (b). Matérn52 kernel (c). Ornstein-Uhlenbeck kernel (d). Cosine kernel. These kernels are the realization of Exponentiated Quadratic covariance function(Equation 3.6), Matérn52 covariance function (Equation 3.11), Ornstein-Uhlenbeck covariance function (Equation 3.12) and Cosine function[1] respectively.

[1]We have not described the Cosine covariance function here. The in detail description will be found at Rasmussen and Williams (2006)

3.5 Gaussian Process Regression

Gaussian process regression can be done using the marginal and conditional properties of multivariate Gaussian distribution. Lets consider that we have some observations $\mathbf{f}$ of a function at observation point $\mathbf{x}$. 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 3.13-

$$p\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_\star\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_\star\end{bmatrix}\middle|\mathbf{0}, \begin{bmatrix}\mathbf{K}_{\mathbf{x},\mathbf{x}} & \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}\\ \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}} & \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star}\end{bmatrix}\right) \tag{3.13}$$

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 property of a multivariate Gaussian is used to perform regression the. The conditional property is can be represented by the equation 3.14:

$$p(\mathbf{f}|\mathbf{f}_\star) = \mathcal{N}\left(\mathbf{f}_\star\middle|\mathbf{K}_{\mathbf{x}_\star,\mathbf{x}}\mathbf{K}_{\mathbf{x},\mathbf{x}}^{-1}\mathbf{f}, \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star} - \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}}\mathbf{K}_{\mathbf{x},\mathbf{x}}^{-1}\mathbf{K}_{\mathbf{x},\mathbf{x}_\star}\right) \tag{3.14}$$

In ideal case the observations $\mathbf{f}$ is noise free but in practice it is always corrupted with some noise. Lets consider $\mathbf{y}$ is the corrupted version of $\mathbf{f}$. If we consider this noise as Gaussian noise then we can write $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I})$, where σ^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 3.15.

$$p\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_\star\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_\star\end{bmatrix}\middle|\mathbf{0}, \begin{bmatrix}\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I} & \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}\\ \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}} & \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star}\end{bmatrix}\right) \tag{3.15}$$

Regression with Gaussian process is Bayesian method. From the knowledge of a *prior* over a function we proceed to a *posterior* and this happens in a closed from of equation 3.14.

Figure 3.6 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 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.

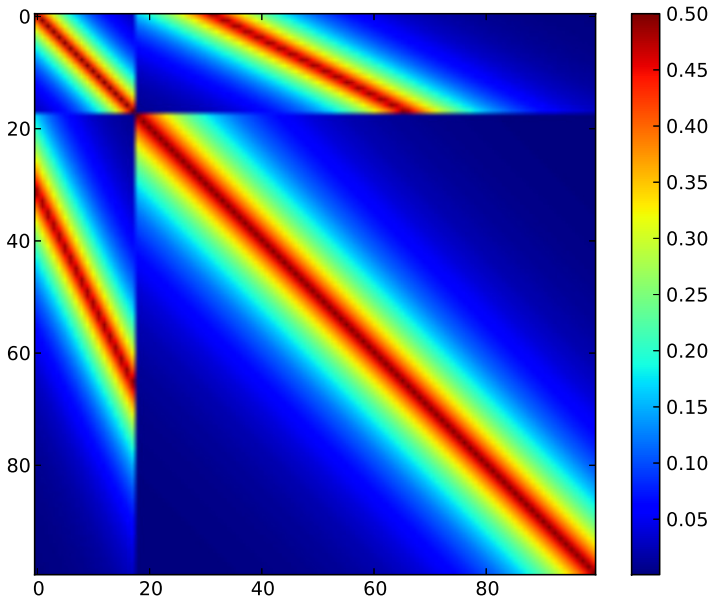

Fig. 3.6 Overall representation of covariances between training and test data

3.5.1 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 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,

$$\mathbf{f}_\star \sim \left(\boldsymbol{\mu}_f, \mathbf{C}_f\right) \tag{3.16}$$

The mean of the conditional distribution of Equation 3.16 is:

$$\boldsymbol{\mu}_f = \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}^{\mathbf{T}} \left[\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right]^{-1} \mathbf{y} \tag{3.17}$$

and the covariance of the conditional distribution of Equation 3.16 given by:

$$\mathbf{C}_f = \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star} - \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}^{\mathbf{T}} \left[\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right]^{-1} \mathbf{K}_{\mathbf{x},\mathbf{x}_\star} \tag{3.18}$$

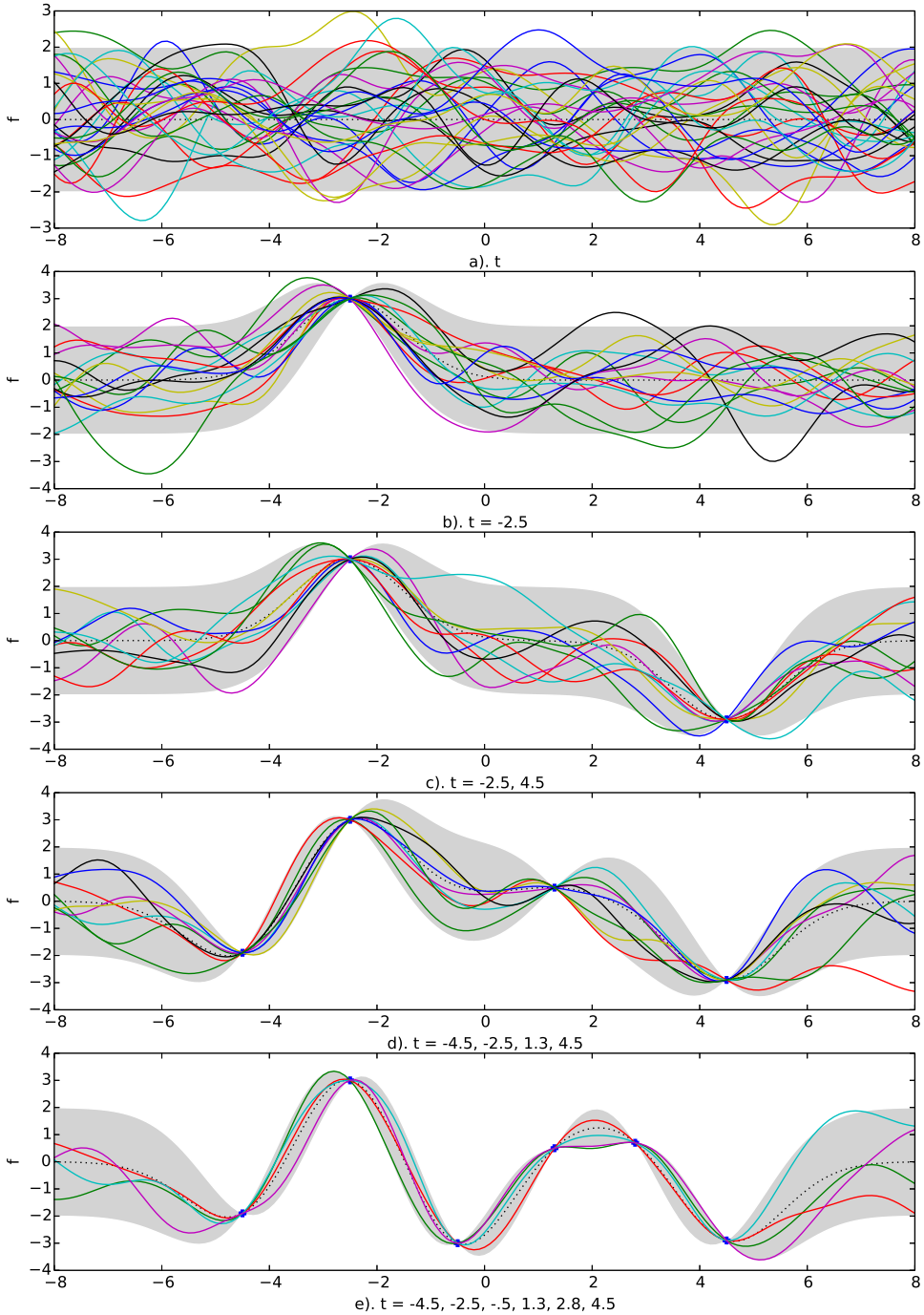

Fig. 3.7 Simple example of regression using Gaussian process

These results can be calculated using block matrix inverse rules. The derivation can found in appendix section (Appendix **??**). Figure 3.7 shows a simple example of regression using Gaussian process.

3.5.2 Hyperparameter Learning

To construct the covariance function still we need to consider the hyperparameters and optimize those. The most efficient and commonly used optimization technique for hyperparameters can be done using maximum likelihood. If we consider all the hyperparameters α, σ^2 and l in to a vector $\boldsymbol{\theta}$, then we can use gradient methods to optimize $p(\mathbf{y}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. The Log likelihood is given by:

$$p(\mathbf{y}|\boldsymbol{\theta}) = -\frac{D}{2} log 2\pi - \frac{1}{2} \times log \left|\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2 \mathbf{I}\right| - \frac{1}{2}\mathbf{y}^T \left[\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2 \mathbf{I}\right]^{-1} \mathbf{y} \tag{3.19}$$

We can have the Log maximum likelihood by:

$$\boldsymbol{\theta}_{max} = argmax\left(p(\mathbf{y}|\boldsymbol{\theta})\right) \tag{3.20}$$

3.6 Toward the GP model of TFA

Simo Särkkä indicated an analogical pathway [2] to construct a kernel function for Gaussian process from Markovian assumption based probabilistic approach of Sanguinetti et al. (2006). In the earlier probabilistic approach gene specific TFAs was obtained from-

$$\mathbf{b}_{n(t+1)} \sim \mathcal{N}(\gamma \mathbf{b}_{nt} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}) \tag{3.21}$$

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

$$\mathbf{b}_{n(k+1)} \sim \mathcal{N}\left(\gamma \mathbf{b}_{nk} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}\right), \tag{3.22}$$

and

$$\mathbf{b}_{n_1} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{3.23}$$

[2] Not published yet, came through some discussions

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

$$u_{k+1} \sim \mathcal{N}\left(\gamma u_k + (1-\gamma)\,\mu, (1-\gamma^2)s\right), \tag{3.24}$$

where μ and s are scalar.

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

$$t_k = kDt. \tag{3.25}$$

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

$$du = -\lambda\,(u-\mu)\,dt + q^{1/2}dB, \tag{3.26}$$

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

$$u(t_k) = au(t_{k-1}) + b\mu + w_{k-1}, \tag{3.27}$$

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,

$$u_{k+1} \sim \mathcal{N}\left(au_k + b\mu, c\right). \tag{3.28}$$

We can now match the coefficients:

$$a = \exp(-\lambda Dt) = \gamma \tag{3.29}$$

$$b = 1-\exp(-\lambda Dt) = 1-\gamma \tag{3.30}$$

$$c = (1 - \gamma^2)s = [q/(2\lambda)][1 - \exp(-2\lambda Dt)] \tag{3.31}$$

Equation 3.29 quite luckily has a nice solution $\gamma = \exp(-\lambda Dt)$ and from Equation 3.31 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 $\lambda = -\log\gamma$
$q = -2s\log\gamma$.

We can now recall that 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 start. 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-

$$\mathbf{K}_b(t, t') = \mathbf{\Sigma}\boldsymbol{\gamma}^{|t-t'|} \tag{3.32}$$

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

$$\mathbf{db} = -\lambda(\mathbf{b} - \boldsymbol{\mu})\mathbf{dt} + Q^{1/2}\mathbf{dB}. \tag{3.33}$$

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.

4.2 Relation to Gene Expressions

We now assume that the jth 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 ith time point, the log of the jth 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 σ^2 on $\hat{\mathbf{y}}_u$ alone using L_u. Once this is done, fix the value of σ^2 in L_q and 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 5.8 shows activities of different transcription factors while the model was developed considering Exponentiated Quadratic kernel with White kernel in additive form.

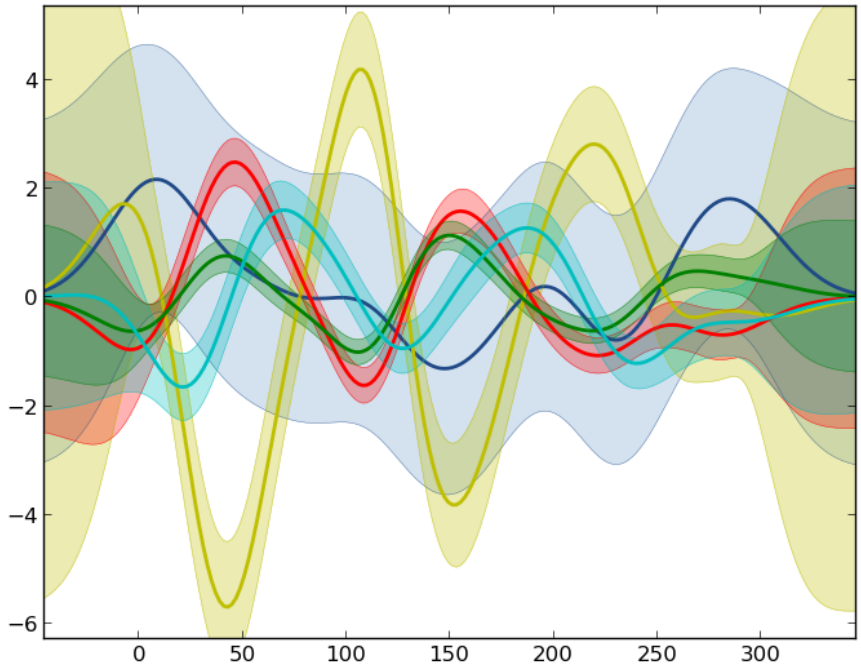

Fig. 4.1 Variation of activities of Transcription factors RBF+white kernels

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

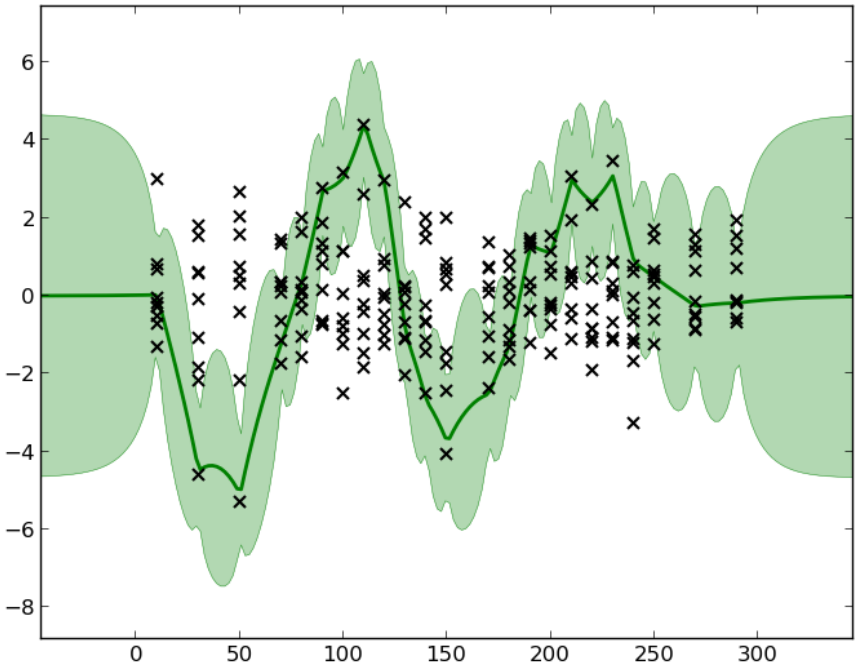

Fig. 4.2 Transcription factor activity of ACE2

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:

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

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

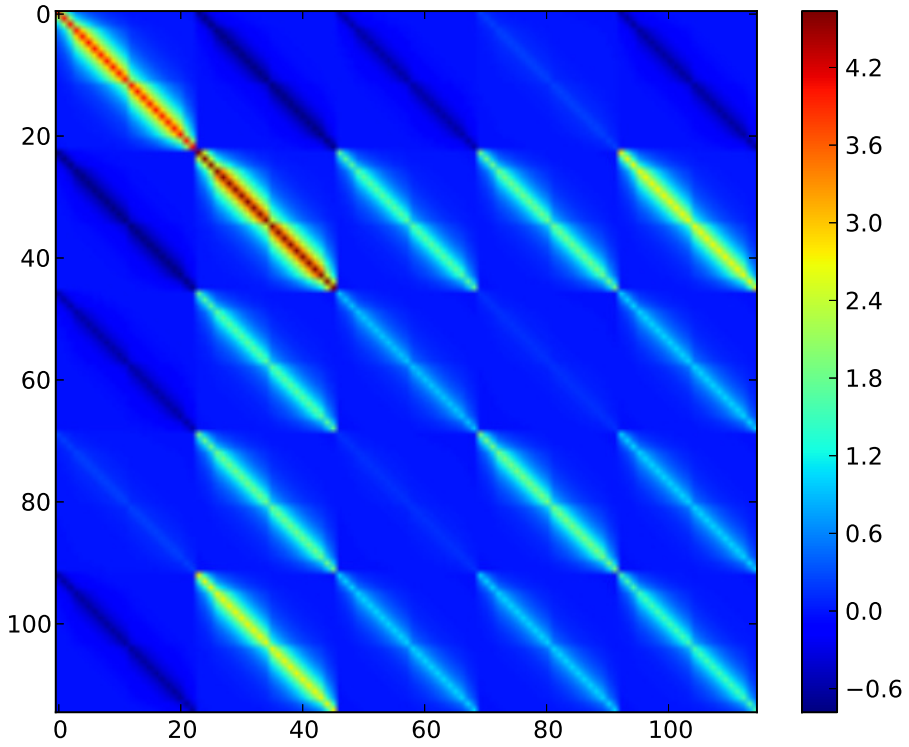

Fig. 4.3 Kernel of Intrinsic Coregionalization model $\mathbf{K}_f$ considering 6 Transcription factors where covariance matrix $\mathbf{\Sigma}$ of (*Equation*4.5) was constructed using Ornstein-Uhlenbeck kernel and White kernel in additive form

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:

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

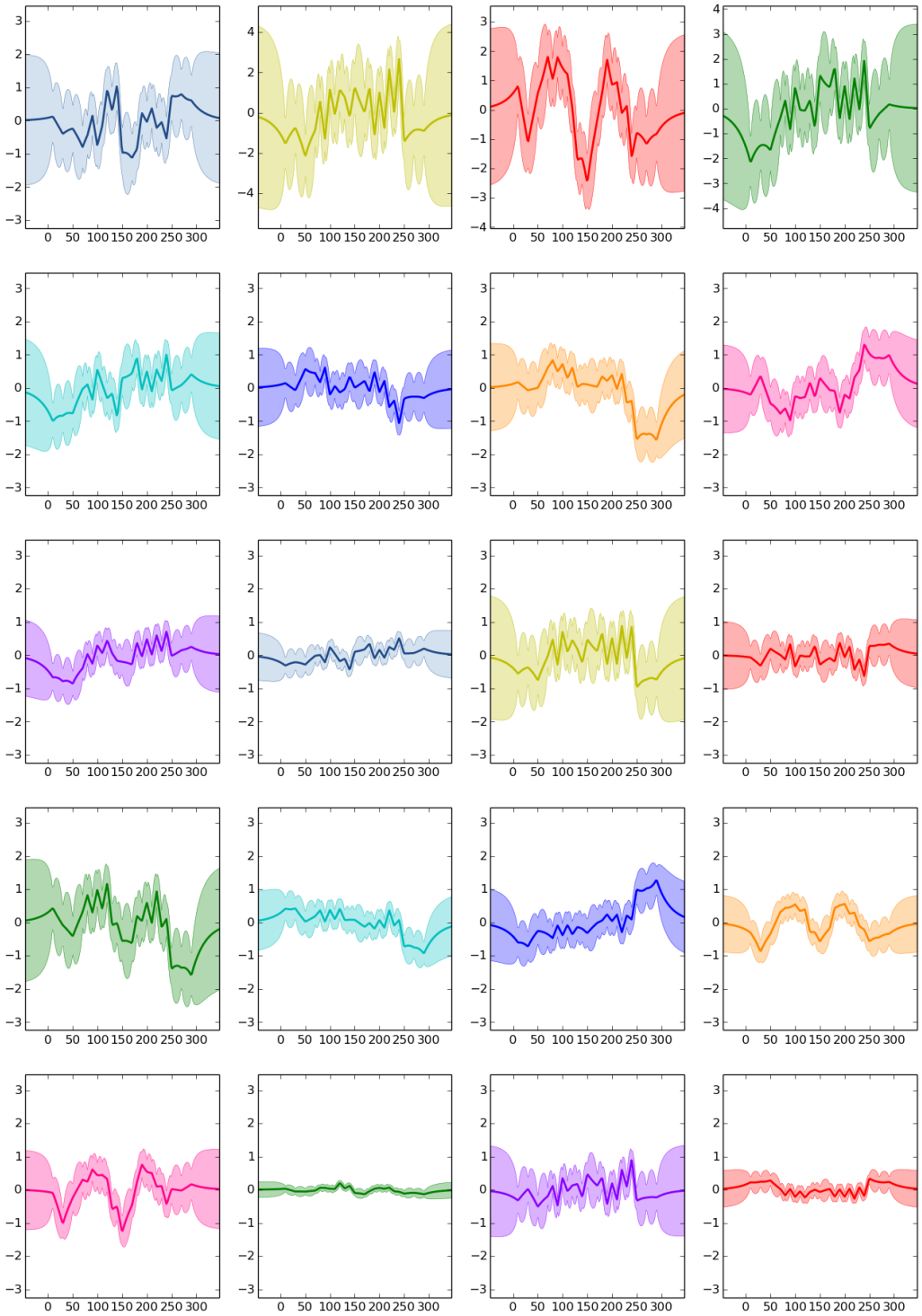

Fig. 4.4 Transcription factor activity of different TF using Ornstein-Uhlenbeck kernel and White kernel in additive form

Chapter 5

Clustering Gene Expression data

5.1 Introduction

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 our paper is to specify the significantly different genes that may effect the speed of ALS progression by building a new model. We used Gaussian process 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 programming language *python* based tool GPy[1], to develop our model. Later

[1]http://sheffieldml.github.io/GPy/

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

We investigated the clusters obtained from our model. We have calculated the enrichment scores Huang et al. (2009a) for every cluster using $DAVID$[2] (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 on some clusters with high enrichment score and demonstrated some interesting

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

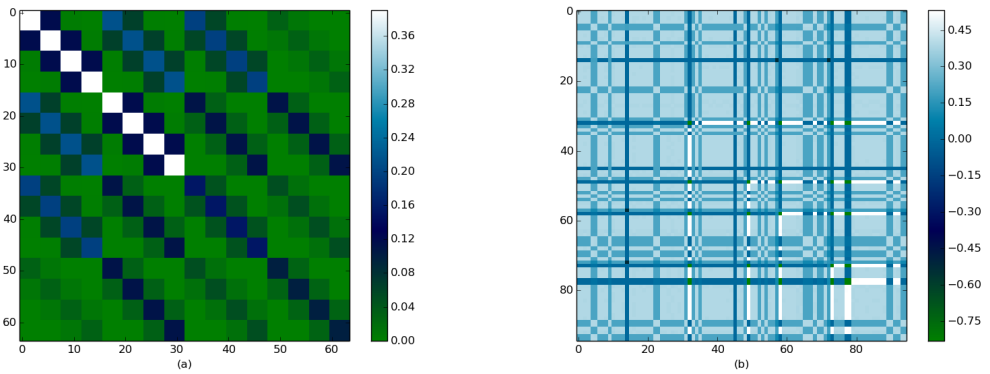

Fig. 5.1 Simple representation of kernel- (a). *Coregionalization* kernel in the input space with 64×64 dimensions; (4 strains $(129Sv - SOD1, 129Sv - Ntg, C57 - Ntg$ and $C57 - SOD1) \times 4$ replicates $\times$ 4 time points or stages of the disease). (b). kernel after optimization considering only 100 genes .

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 some functionality for the speed of propagation of ALS particularly with reference to this specific type of mouse model.

5.2 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 Honkelaa 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. (2013). Whilst Gaussian process also overcome the requirement of evenly spaced time points for time expression data.

Here we constructed a hierarchical Gaussian process Hensman et al. (2013) 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 replications (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.

5.3 Methodology

5.3.1 Hierarchical Gaussian Process

Our gene expression time series came from four different strain and there are four biological replicates. So for every individual gene we can incorporate these in an hierarchical fashion. Let $\mathbf{y}_{nr}$ denotes gene expression of n^{th} gene in the r^{th} biological replicates and i^{th} biological strain. Measurements were made at four different times and collected into a vector $\mathbf{x}_{nir}$. The data for i^{th} strain's n^{th} gene is denoted by $\mathbf{Y}_n = \{\mathbf{y}_{nr}\}_{r=1}^{N_n}$ and $\mathbf{X}_n = \{\mathbf{x}_{nr}\}_{r=1}^{N_n}$.

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

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

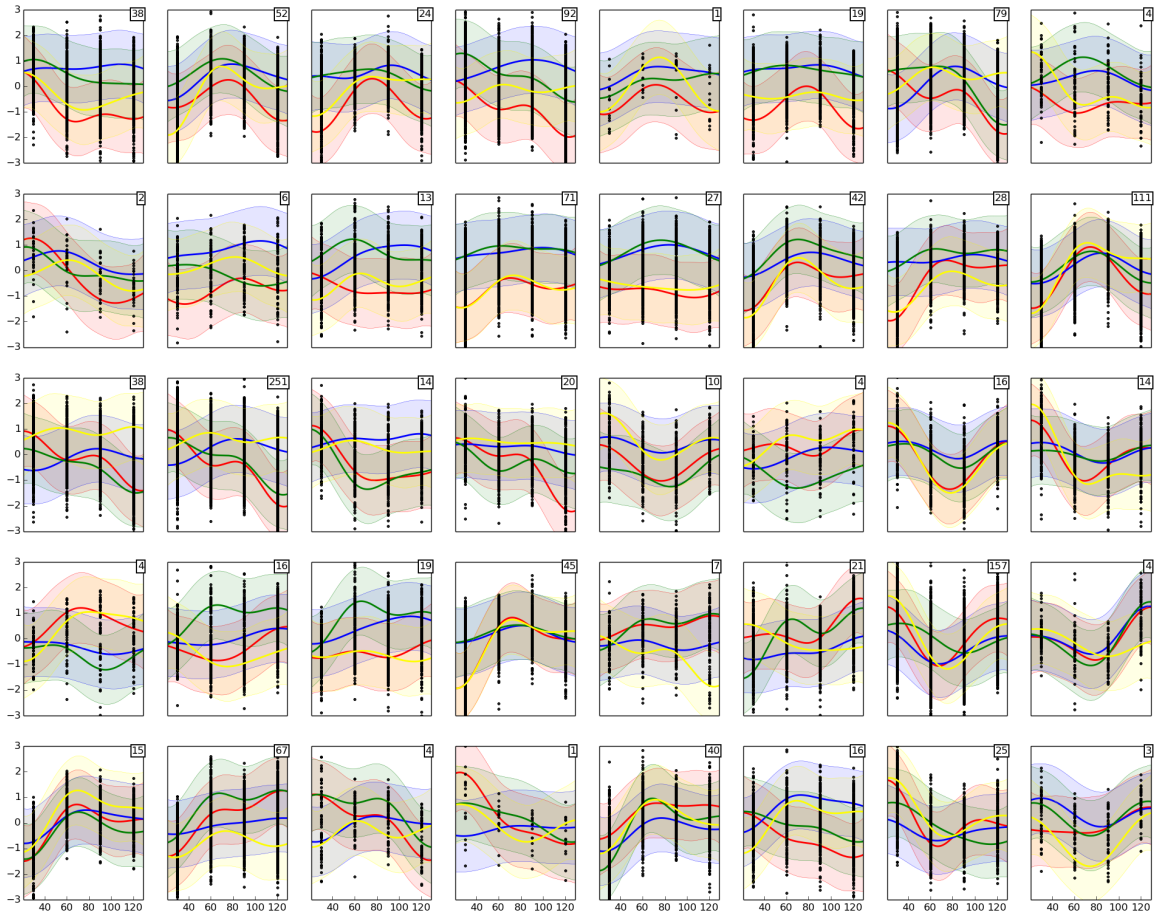

Fig. 5.2 Clustering genes expressions using hierarchy of Gaussian processes. Some representative clusters from the 203 clusters generated (top to bottom, left to right: cluster 136 to 143, 151 to 158, 166 to 173, 181 to 188 and 196 to 203). Along x-axis the four time stages are pre-symptom, onset, symptom 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.

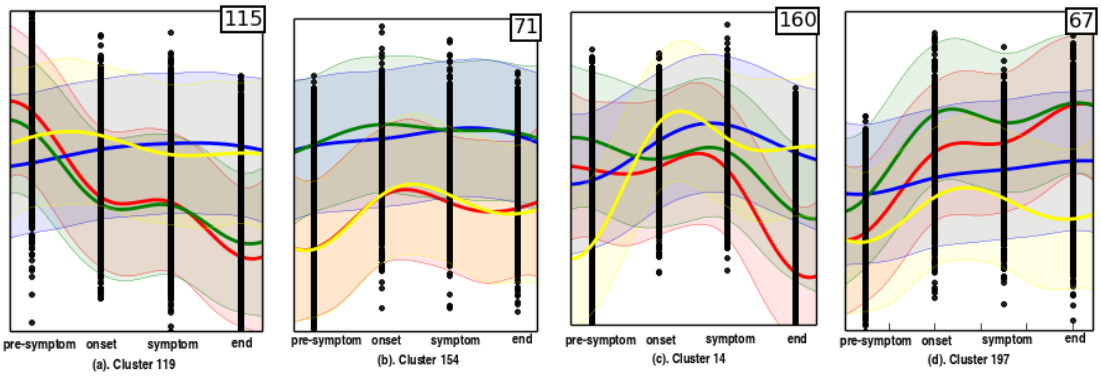

Fig. 5.3 Along x-axis of each individual figure four time stages are pre-symptom, onset, symptom and end-stage. Four different colours yellow, red, green and blue are representing four mouse strains $129Sv-SOD1, 129Sv-Ntg, C57-Ntg$ and $C57-SOD1$ respectively. Examples of clusters where genes from different 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.

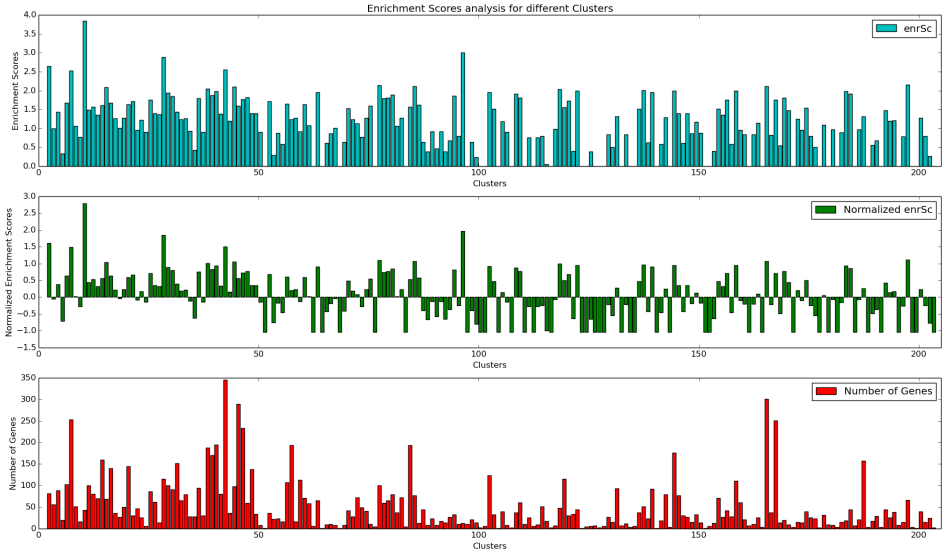

Fig. 5.4 Enrichment scores analysis. Figure in the top shows the enrichment score for different clusters where $x - axis$ is the cluster number and $y - axis$ shows the enrichment score of that cluster. Figure located at the middle shows the normalized score. While the bottom figure shows the number of genes belongs to any specific cluster.

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

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

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

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

While designing different kernels k we have used *coregionalization* model.

5.3.2 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 the replicates and strains. In machine learning language this approach is sometimes known as 'multi-task learning' 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 $a_{d,q}$ 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

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

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. 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. Here 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 5.1 shows a simple representation of the *coregionalization* kernel in the input space and the representation of an optimized kernel where we considered only 100 genes.

5.3.3 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 sub groups through a mixture of Gaussian process models: each component is a function over time, genetic background and condition.

Partitioning genes into clusters can be done by some using our Gaussian process prior over the functions and a Dirichlet process prior for the mixing coefficients. This can be achieved through Gibbs sampling Dunson (2010), but this can be slow in practice. A potentially improved model was proposed by Hensman et al. (2013), where they consider the structure of covariance across the gene and separately across replicates. The use 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 expectation maximization algorithm is used with new covariance function[3].

5.4 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

[3]The idea is implemented in a tool named named *GPClust*, available at https://github.com/jameshensman/GPclust Hensman et al. (2013).

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 using Gaussian process Kalaitzis and Lawrence (2011) can be used to analyse the time series gene expression and filter the quiet or inactive genes from the differentially expressed ones. In addition identifying genes that have a good signal-to-noise ratio (SNR) is also used to filter down the total number of genes that need further analysis. We can rank the genes by the ratio of the mean replicate-wise variance to the variance of the replicate-wise means. In our analysis we used a combination of both of the approach. 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 for further analysis. Before the filtering the gene expression levels of each replicates were normalized to zero-mean over all the samples.

Cluster analysis: In the previous selection stage we derived 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 5.2 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 variety of number of genes. Many of the clusters indicated different dynamic behaviour of the gene set. Many of the clusters were attractive for further analysis but that is beyond the scope our this study. We included some examples in the Figure 5.3. 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.

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

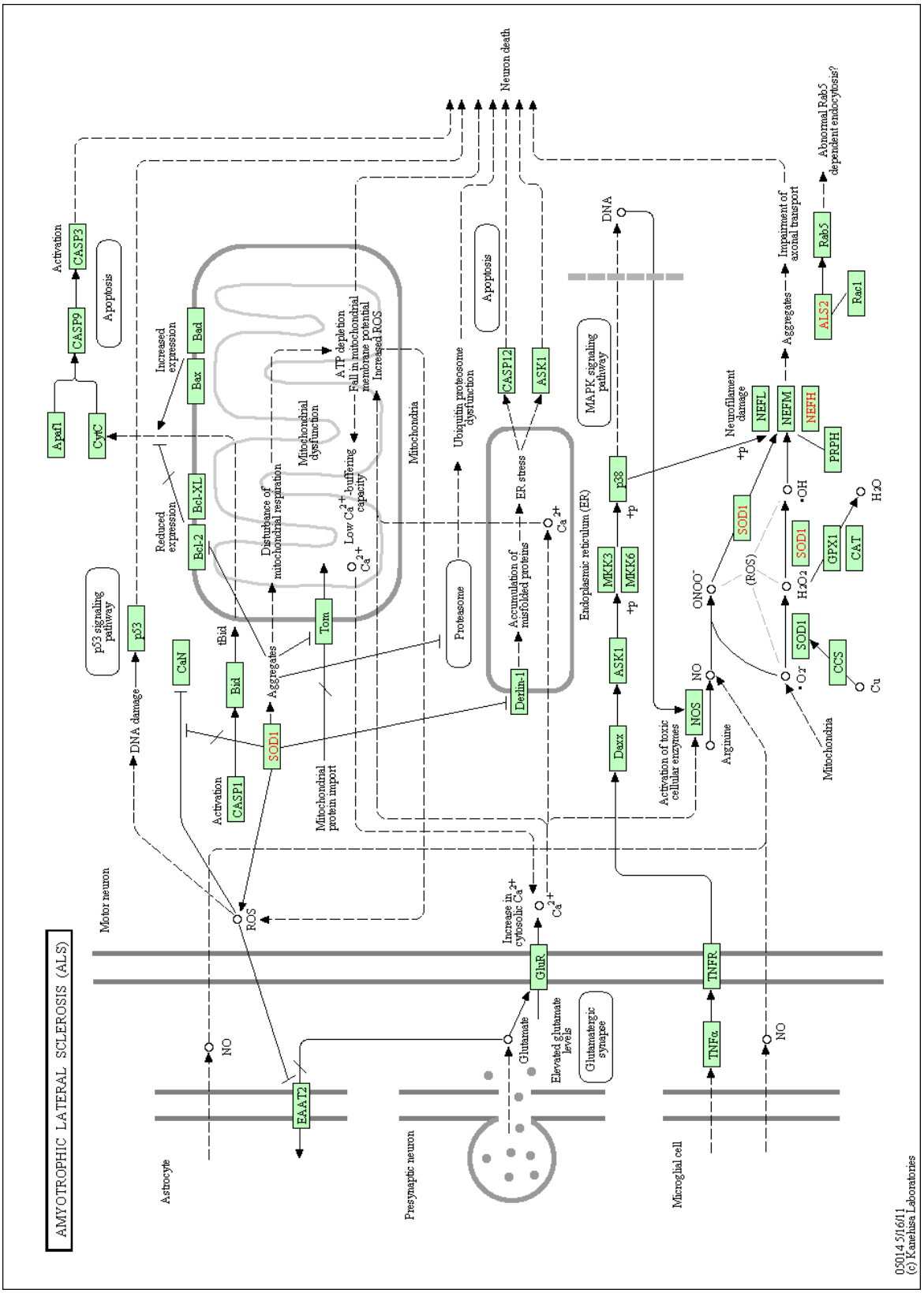

Fig. 5.5 Pathway analysis: KEGG_Pathway.

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) 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 5.4 shows the result. Whilst in an analysis a group of genes with an enrichment score of 1.3 can be considered as a threshold value to decide whether this list of genes are enriched or not, here for our 203 clusters we have found at least 15 clusters have an enrichment score of ≥ 2.

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 earlier stages) and performed gene ontology enrichment analysis (one example is given at Table **??**) and pathway analysis on individual clusters. We identified one of our cluster (cluster197; Figure 5.3) 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, $Derlin-1$, can accumulate with other misfolded proteins and cause the neuron death and belongs to our chosen cluster. Figure 5.5 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 neuron death and related with some other neural disorder like Parkinson.

5.5 Conclusions

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 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 will be useful to investigate other biological activity where clustering is required.

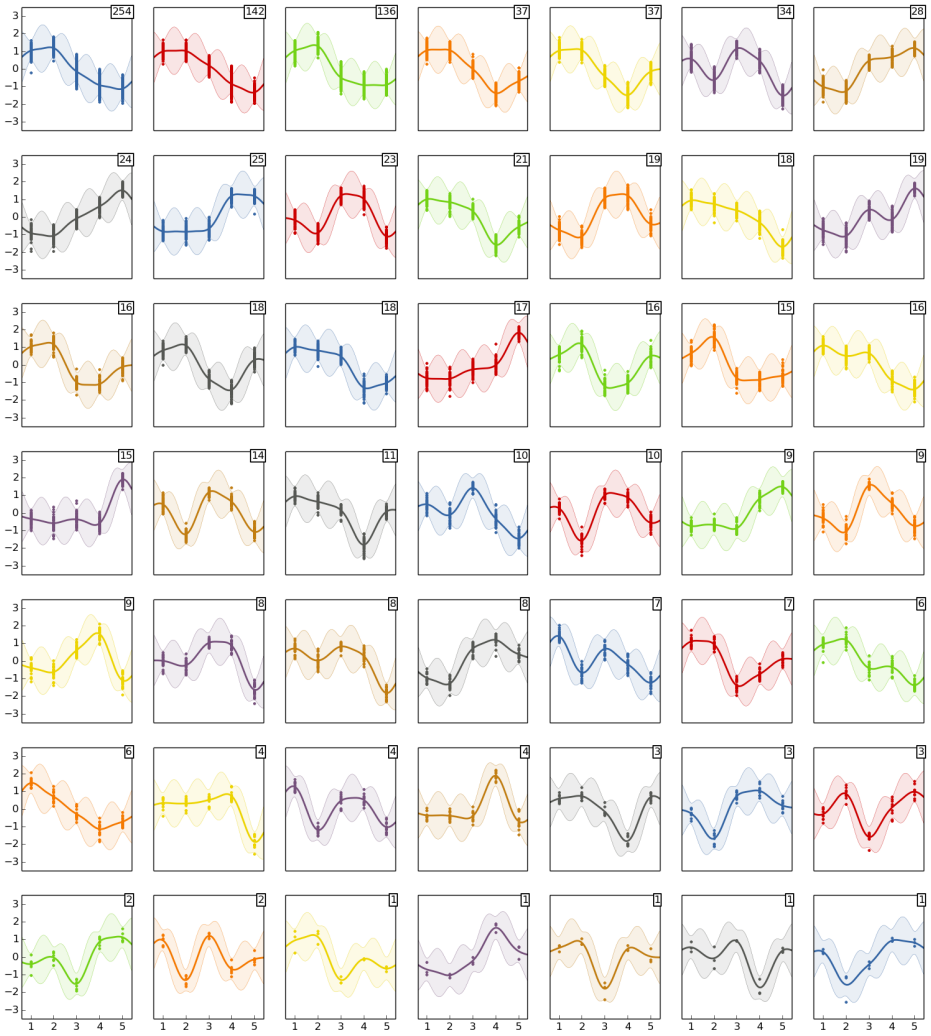

Fig. 5.6 Clustering Gene Expression data

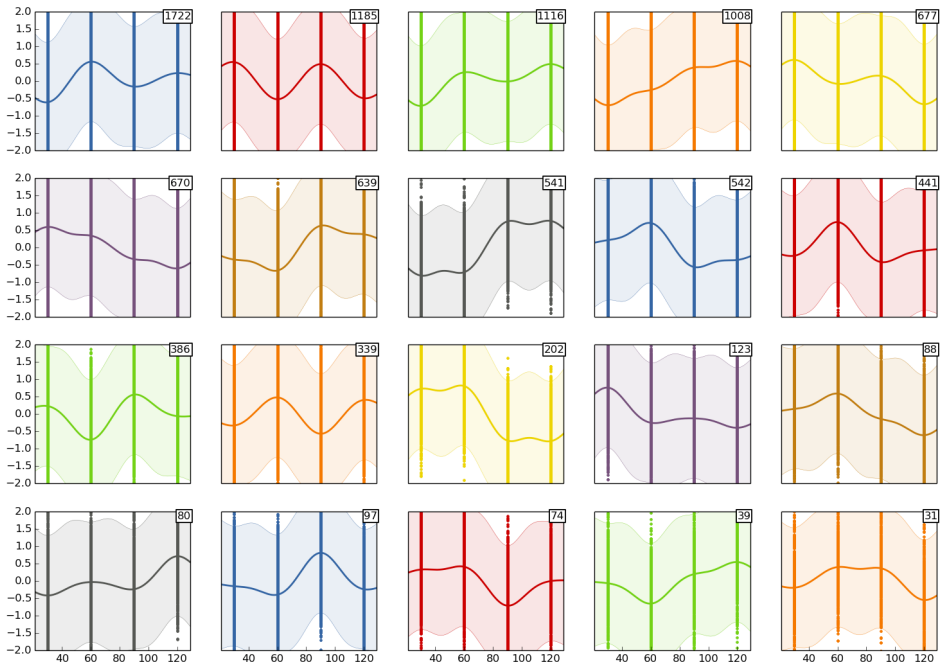

Fig. 5.7 Clustering Gene Expression data no coregionalization

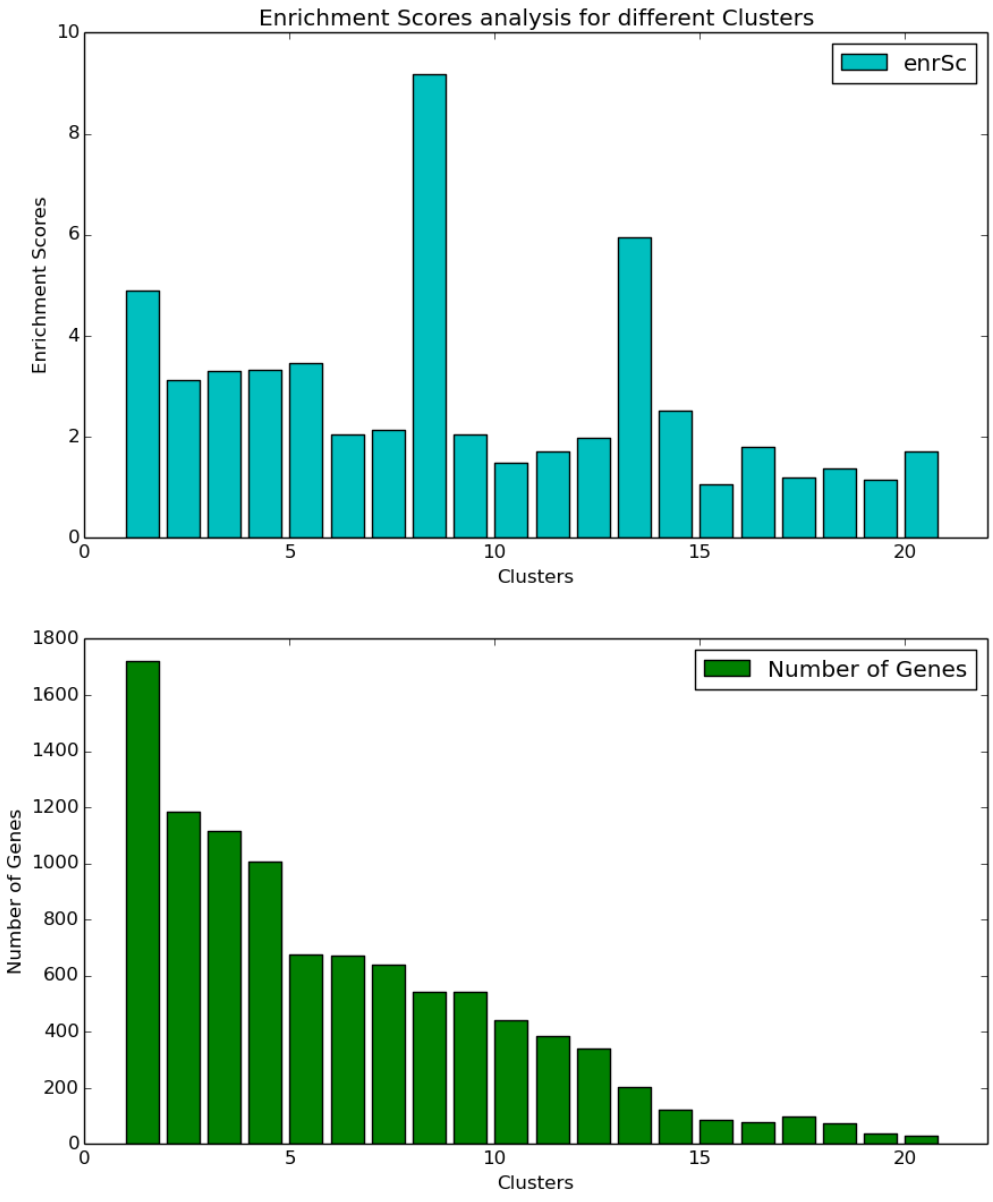

Fig. 5.8 Clustering Gene Expression data bar Graph

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. (2013) 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. (2013) 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 43)

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 22)

Alvarez, M. A. and Lawrence, N. D. (2011). Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, (12):1459–1500. (pages 70 and 71)

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 54 and 71)

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 65)

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 64)

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press. (page 37)

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 38)

Boulesteix, A.-L. and Strimmer, K. (2005). Predicting transcription factor activities from combined analysis of microarray and chip data: a partial least squares approach. *Theoretical Biology and Medical Modelling*, 2(23). (page 22)

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

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, 64, and 75)

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 64)

Castaneda, L. B., Arunachalam, V., and Dharmaraja, S. (2012). *Introduction to Probability and Stochastic Processes with Applications*. Wiley. (page 37)

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 32, 34, and 83)

Dunson, D. D. (2010). Nonparametric bayes applications to biostatistics. *Bayesian Nonparametrics, Cambridge University press*. (page 72)

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 32)

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 64)

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 22)

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman and Hall / CRC. (page 65)

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 64)

Hensman, J., Lawrence, N. D., and Rattray, M. (2013). Hierarchical bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. *BMC Bioinformatics*, 14(252). (pages 66, 72, and 83)

Honkelaa, 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 65)

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 64 and 75)

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 64 and 75)

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 26 and 29)

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 32, 34, 65, 73, and 82)

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 37)

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 27)

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 8 and 53)

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 22 and 27)

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 66)

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. (page 38)

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 65)

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 64)

Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in Applied Probability*, 5:439–468. (page 37)

Medvedovic, M., Yeung, K., and Bumgarner, R. (2004). Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8). (page 66)

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 22)

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 72 and 82)

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics No. 118, New York: Springer-Verlag. (page 37)

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 66)

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 37)

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 37)

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 25 and 73)

Pe'er, D. (2003). *From gene expression to molecular pathways*. PhD thesis, The Hebrew University. (page 32)

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, 64, and 82)

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 64)

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 37, 39, 40, and 45)

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)

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 64)

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, 22, 23, 24, 25, 49, 53, 57, 65, 81, and 82)

Shaye, D. D. and Greenwald, I. (2011). Ortholist: A compendium of c. elegans genes with human orthologs. *PLoS ONE*, 9(1). (pages 9 and 81)

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 25, 53, and 65)

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 56)

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 65)

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 64)

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Phys. Rev.*, 36:823–841. (pages 44 and 50)

Wackernagel, H. (2003). *Multivariate Geostatistics An Introduction with Applications*. Springer Science and Business Media. (pages 54, 70, and 71)

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 37)

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 Rasmussen, C. E. (1996). Gaussian processes for regression. *Advances in Neural Information Processing Systems 8, MIT Press*, pages 1–7. (page 37)

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

WormNet (2014). Wormnet. http://www.functionalnet.org/wormnet/about.html [Online; accessed 30-Nov-2014]. (pages 27 and 29)

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 27)

Appendix A

Apendix 1

Windows OS

TeXLive package - full version

Appendix B

Apendix 2