Eberhard Karls Universität Tübingen
Fakultät für Informations- und Kognitionswissenschaften
Wilhelm-Schickard-Institut für Informatik

Bachelor Thesis Bioinformatics

Detecting and modeling time shifts in microarray time series data applying Gaussian processes

Max Zwießele

30.08.2010

Reviewer

Kay Nieselt
University of Tübingen

Supervisor

Karsten Borgwardt, Oliver Stegle
Max Planck Institute Tuebingen

Name, first name: Zwießele, Max

Detecting and modeling time shifts in microarray time series data applying Gaussian processes

Bachelor Thesis Bioinformatics

Eberhard Karls Universität Tübingen

Period: 01.05.2010-30.08.2010

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Bachelorarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum Unterschrift

Abstract

Microarray time series experiments enable us to analyse vast amounts of genes' expression levels. Thus, they are useful to study the regulatory networks controlling the genes of an organism. Assessing if a time series resembles another is difficult, as for instance, noise in data has to be considered. Another reason for variability in data is a simple time shift between time series, which occurs due to technical and biological problems microarray researchers still struggle with. In this thesis, we attend to model this time shift explicitly. This results in an improvement in differentially gene expression detection, as well as in the ability to model correlations between time shifts and transcription factors.

First, we develop a method to model such a time shift in time series by the means of Gaussian process frameworks (called GPTimeShift). Second, we present two applications of our Gaussian process based model on microarray data. For the data set containing exposition studies of *Arabidopsis thaliana* to a fungal pathogen (i.e., *Botrytis cinerea*) we could show a slight improvement as compared to a standard Gaussian process based model without time shift. For the data containing *knock-out* experiments of yeast populations we were able to investigate correlations between transcription factors and time shifts.

The applications introduced here, are just two examples of the enormous capability of our model GPTimeShift. We can imagine a wide field of GPTimeShift, such as clustering time series, modeling cell-cycle behavior, or modeling flowering time experiments. In summary, our model GPTimeShift sheds a new light on (microarray) time series data and future studies will investigate more advantages.

Acknowledgements

First of all, I would like to thank Karsten Borgwardt and Oliver Stegle for giving me the opportunity to work on this thesis and for their indispensable feedback. Furthermore, I would like to thank Oliver Stegle for his innovative ideas, excellent motivation and virtuous patience.

I also would like to thank the members of my work group - Christoph Lippert, Nino Shervashidze, Theofanis Karaletsos, Till Helge Helwig, Karin Klotzbücher, Barbara Rakitsch – as well as Christopher Burger, Yoshinobu Kawahara and Christian Schuler for useful, inspiring discussion and proper proofreading.

Finally, I would like to thank my parents Sibylle and Frieder for their patience and assistance in helping me finishing this thesis.

Contents

List of Figures

List of Tables

Thesis Outline and Motivation

Newest approaches in microarray experiments allow us to analyse vast amounts of gene expression time series. In Chapter one, we will introduce the biological and technical background for microarray experiments. The technical background, indeed, includes the preprocessing and normalization issues microarray developers still compete with. Many studies about microarray time series measurements where done to reveal differentially expressed genes. For examples, refer to the papers of [SDW+09, CC03, NNSA04]. Almost all existing approaches detecting differentially expressed genes implicitly assume no time shifts between time series. We presume a time shift in such time lines can explain technical and biological problems, microarray measurements prone to (see Chapter one). Therefore, our novel approach is to include a time shift into Gaussian process regression of microarray time series data. The first section of Chapter two will review the mathematical background of how to use Gaussian processes [RW05] to model and analyse microarray time series experiments. In the following sections of Chapter two, we will reveal and explain our novel approach, in which we include time shifts in time lines of microarray time series measurements, applying Gaussian processes. In Chapter three, we will present the results of the application of our novel approach, called *GPTimeShift* (see Appendix B.2 for the implementation). The first application is to detect differentially expressed genes in the stress response of *Arabidopsis thaliana*, exposed to a fungal pathogen (i.e., *Botrytis cinerea*). Second, we will present the application of GPTimeShift to the stress response of yeast populations on *knocked-out* transcription factors (TFs). We want to detect TF pairs that are influenced by the knock-out of *mei4*. Using the fact that *mei4* does not influence the targets after it is knocked-out, we determine correlations between the time shift and TF pairs. We will discuss the results of both applications in Chapter three and state our conclusions about the results in Chapter four. Additionally, we will also elucidate the deficits of our approach and how future applications can possibly overcome this deficits in Chapter four.

Chapter 1

Biological and Technical Background for Differential Gene Expression Detection

In this thesis, we will study time shifts in full genome expression profiles. Our ultimate goal is to elucidate the influence of time shifts on regulatory networks of biological organisms. Therefore, we will analyse the stress response of *Arabidopsis thaliana* and *yeast* populations to the exposure of stress factors and the changes such an exposure coerce. Several macromolecular processes arouse these changes in regulatory networks, such as: *DNA structure*, *gene expression* and *differential expression*. In Section 1.1, we will review the biological background that is essential to understand these macromolecular processes.

To analyse gene expression, we have to denote the expression level of genes. A widely used technique is the analysis of *microarray experiments*. *DNA microarrays* are gene chips which measure the expression level of thousands of genes at a certain time point. The data produced by microarrays is prone to technical problems, namely, *noise* in measurement, different *experimental protocols*, and slight *variance* in experimental conditions. Additionally, the measurements of microarrays subject to biological problems (i.e., *noise*, dependency to *temperature*, and slightly differences in conditions of *replicates*). We will elucidate the existing approaches handling these problems in Section 1.2. We assume, the interaction of these technical and biological problems to cause time shifts between measurements. To prevail these time shifts, we model them explicitly, using Gaussian processes (see Chapter two).

In Section 1.1.4, we will explain the technology of microarray experiments in more detail. Two, or more microarray experiments, considering the same organism in two or more conditions and time series, provide differential gene expression detection. There are many approaches to detect differentially expressed genes [SDW+09, LRM+06, NNSA04, CC03]. We will review some key methods of how to

determine differentially expressed genes in Section 1.2.2.

1.1 Introduction to DNA, Genes and Differential Expression

Most heritable information of each cell in the organism is coded in *deoxyribonucleic acid (DNA)*. One unit of heritable information is called a *gene*. For understanding an organism, we are to understand DNA. DNA contains all genes, which will be transcribed and translated into proteins. A gene is called *expressed* if it is transcribed into mRNA. One gene can be translated into one or several proteins. The process of transcription and translation will be explained in Section 1.1.1. To understand the role of a certain protein, we analyse the expression level of the corresponding gene. We assume the transcript (i.e., protein) of a gene to be activated in the cell if the corresponding gene is expressed. Supplementary processes control and eventually deactivate the transcript of a gene after a certain lifetime. Genes are influenced in their activity by several factors, such as: the environment, other genes, splicing activities, proteins, cellular states and stress factors. To restrict these amounts of data we concentrate on the study of stress reactions. Exposing an organism to a stress factor causes the organism to react. One reaction among others is to change the gene expression level of the exposed cell structure. The specific character of such a change critically depends on the stress factor the organism is exposed to. Stress factors can be fungal, viral, or bacterial infections.

A detailed view on genetics is far beyond the scope of this thesis. We would like to refer interested readers to the books of [Hen98, SB03, ZSTV04].

1.1.1 DNA: History and Structure

Before we look at the functionality of DNA, it is certainly important to know about the discovery of DNA. The history of DNA begins with Gregor Mendel. Mendel discovered in 1865 through breeding experiments with peas that the different phenotypes of the peas were inherited based on specific laws [Men65]. These laws were later called "Mendelian laws". In 1869 Friedrich Miescher first isolated DNA, but at this time he failed to realize the importance of his discovery. In 1909 Wilhelm Johannsen the first time used the word *gene* to describe a *unit of heredity*. In the years from 1949 until 1953 the structure of DNA was revealed by several studies. DNA consists of four bases, connected by a sugar-phosphate *backbone*. These bases, called *nucleotides*, build up two complementary linearly composed strands, which form a *double helix*. Two nucleotides respectively form hydrogen bonds, which stabilize the double helix. The four nucleotides are **A**denine, **T**hymine, **G**uanine and

Cytosine. **A** and **T** can bind to each other by forming two hydrogen bonds. Therefore, **A** and **T** are said to be *complementary*. **G** and **C** are also complementary: they form three hydrogen bonds. This complementary strand is called *complementary DNA (cDNA)*. When two fitting cDNA strands bind to each other they build up the double helix, mentioned above. The binding of two complementary cDNA strands is called *hybridization*. Inside the cell, DNA is always present in double helix form. Figure 1.1 depicts the double helix form with its backbone and nucleotides.

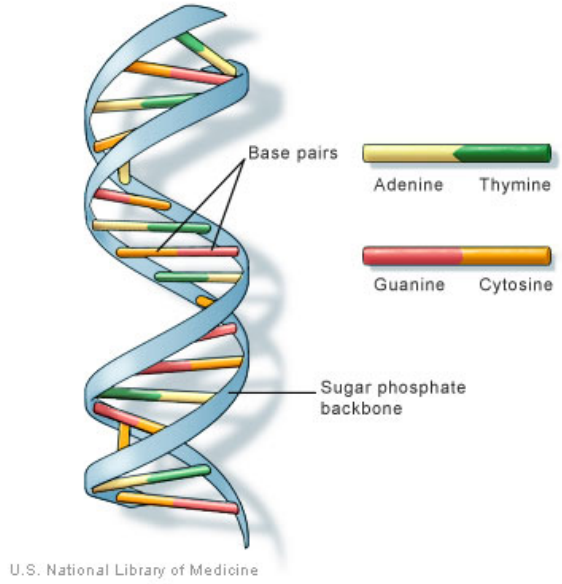


Figure 1.1: Double helix structure of the DNA. The double helix is built up by linearly linked nucleotides - connected by a sugar-phosphate backbone. Hydrogen-bonds form only between **A** and **T** and between **G** and **C**.

In 1957, Francis Crick stated that the DNA is transcribed into *RNA* and then translated into a protein. In 1977, the first methods to *sequence DNA* were introduced by Frederick Sanger, Allan Maxam and Walter Gilbert. This was a big step towards understanding the *genetics* of an organism. The next years revealed many different properties of DNA. In the 1980's, the first *microarray technologies* were developed. By 2001, the human genome was sequenced as a whole and published in Nature [LLB+01]. In the years since 1865 many more discoveries about DNA and its properties were made, but to describe all of them would go beyond the scope of this thesis.

1.1.2 Properties and Peculiarities of Genes

For a better understanding of genes, let us have a detailed look at what a gene is.

A gene of an organism is a segment of one of the organism's *chromosomes*. It holds the *hereditary information* for one or more specific proteins of the organism. Its *transcription and translation* is regulated by other genes or proteins of the genome of the organism. One chromosome contains - depending on its organism - up to *several thousand* genes. An *eukaryotic* gene may contain *exons* and *introns*. Additionally, there are gene expression *regulatory regions* at the beginning (*5'-end / promoter*) and the end (*3'-end*). [ZSTV04]

The complementary transcription of introns and exons of DNA forms the pre-transcript. The pre-transcript consists of pre-messenger RNA (pre-mRNA), as it is not spliced yet. Cutting out the introns of pre-mRNA is called *splicing*, and the abundance of translated mRNA is called the *expression level* of the gene. *RNA* is quite similar to DNA, but usually consists of only single strands, and the nucleotide **U**racil replaces **T**hymine. After splicing, mRNA is translated into a protein by the ribosomes. The ribosomes connect amino acids (AAs) to each other, one AA for every triplet of mRNA nucleotides. A triplet of mRNA, coding for one AA, is called *codon*. Which AA is used is determined by the codon of mRNA nucleotides. In summary, each protein of the organism is coded in DNA. Genes are transcribed into mRNA, and translated into the protein by ribosomes. The flow of information from DNA to the proteins is illustrated in Figure 1.2.

The *genome* of an organism comprises the set of all genes. These genes exert a variety of biological functions. One of the key steps towards understanding the function of a gene is to determine which influences the gene reacts on and which influences it might have on other genes. These discoveries can be made by detecting differential expression of genes between certain conditions.

1.1.3 Differential Gene Expression

A gene of an organism is transcribed and translated depending on the state the organism is in. The state of an organism changes, due to environmental influences it is exposed to. Changing states of an organism affects the expression level of genes to differ. Namely, these differences in expression levels arouse due to *stress factors*, other proteins, cell-cycle activity and more factors. If the expression of one gene varies between two, or more, conditions it is called *differentially expressed* between these conditions. These conditions represent different cell-cycle stages or exposition to stress versus no exposition. To get a more detailed and realistic picture of gene expression, one studies time series of gene expression levels. One now can not just distinguish between the 'yes' and 'no', but also can decide 'when' the gene

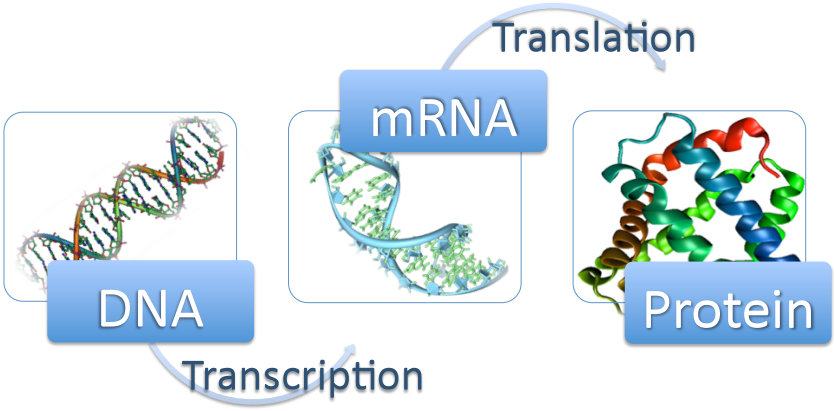


Figure 1.2: The translation and transcription of a gene into a protein. First, the DNA[1] is transcribed into the mRNA[2], where one mRNA nucleotide is the transcript of one DNA nucleotide. Second, the mRNA is translated into the corresponding protein[3], where one AA of the protein is coded by one triplet of mRNA nucleotides.

is differentially expressed. This gives the opportunity to make assumptions on the co-regulation of genes. Genes that are differentially expressed in the same state of an experiment are assumed to be co-regulated by the same *transcription factor(s) (TFs)*. A TF is a protein that influences the expression level of one or more genes by binding to the binding sites of the gene's promoter region.

1.1.4 Technology and Design of Microarrays

A key step in determining the function of a gene is to measure gene activity in terms of its expression level. Therefore, *DNA microarrays* were developed. DNA microarrays are gene chips, consisting of a membrane or a glass slide on which *probes* of genes are applied. The technology is based on the *hybridization* of DNA. The applied probes of known genes are hybridization points for *samples* of the organism. This is why the genes of the probes have to be pre-known and specific for a subsequence of the samples. There are two different ways to apply the probes onto

[1]DNA: `http://wiki-commons.genealogy.net/images/f/f0/DNA_Overview.png`
[2]mRNA:`http://upload.wikimedia.org/wikipedia/commons/4/4a/Pre-mRNA-1ysv.png-tubes.png`
[3]Protein: `http://commons.wikimedia.org/wiki/File:Myoglobin-1mba.png`

the surface of the chip. The first one is to *spot* the probes of cDNA onto the surface of the array by a *print tip*, and the second is, to synthesize the probes directly onto the chip *in-situ*. These synthesized probes are *oligonucleotides*, often called *oligos*.

For a full review on microarrays, please refer to [BH02, ZSTV04].

1.2 Microarray Gene Expression Time Series

Comparing two, or more stress mediated conditions promises to provide deep insights into the regulatory networks of an organism. The vast amount of genes one can measure by microarray experiments provide a substantial step towards understanding the molecular mechanisms mediating the response to an exposure to stress. Furthermore, capturing the state of genes over time reveals the time-dynamic changes due to this stress exposure. We aim at revealing these time-dynamic changes by analyzing time shifts in the time line of microarray data.

After explaining low-level microarray processing [BCS+04], namely, preprocessing and normalization of microarray raw data, we will discuss existing high level microarray analyses in Section 1.2.2. Additionally, we motivate considering a time shift in microarray data.

1.2.1 Low-Level Analysis of Microarray Data

A couple of facts are essential to the proper analysis of microarray data. First, *background noise* occur, due to nonspecific hybridization of the samples. For instance, the mean background is estimated and subtracted. Therefore, the resulting expression levels may be negative, although true expression levels cannot be negative. Second, *normalization* should be applied, to compensate varying efficiencies of the intermediary steps of the measurement, and varying amounts of oligos per cell.

Preprocessing of Microarray Raw Image Data. For each microarray experiment, one has to process the raw image data, namely, the signal intensities for each spot. The raw image data is adjusted and corrected in four steps. First, *detect* the oligo spots. Second, eliminate *background noise*. Third, determine the *quantity of signal and noise*. And, eventually *qualify* the data by, for instance *signal-to-noise-ratio*. These four steps of correction are called *preprocessing* of the raw image data [BCS+04]. After preprocessing the microarray data is mostly represented as a real vector or matrix of expression levels. The expression vectors or matrices have to be *normalized* within one and between several experiments, as one wants to compare these

experiments in terms of differentially expression detection.

Microarray Normalization. After image-analysis of microarray data, the preprocessed raw expression data has to be normalized to ensure data set comparability and to compensate experimental variances. There were several approaches addressed to these problems. Quackenbush *et al.* [Qua02] reviewed and explained a couple of normalization methods, such as: *lowess normalization*, *intensity based filtering*, or *average over replicates*.

1.2.2 High-Level Analysis of Microarray Data

After low-level processing of microarray data, we can finally go into determining differentially expressed genes. There exist many different approaches for detecting differentially expressed genes. One distinguishes between the following three big groups of methods. **Clustering methods, [MDA04, Chapter 3,4]** addressing the visual comparison of gene expression shapes. **Statistical methods, [MDA04, Chapter 5]** for detecting statistical significant differentially expressed genes (for instance, the well known t-test). And **Classification methods, [MDA04, Chapter 7]** which provide preferably unsupervised classification of expression profiles into different classes, namely, machine learning. For instance, one application of machine learning is the Support Vector Machine. [SS02, SC08]) Most of these methods do not address microarray time series data, but experiments holding just two conditions.

As microarray time series data prone to terms of biological and technical problems (1.2), there are always little time shifts in data preventing existing clustering, identification and classification methods from a clean differential gene expression detection (i.e., they produce false positives), namely, due to a time shift in time line, even the shapes of two similar samples differ. These FPs can be avoided by allowing the algorithm to shift the data inputs against each other in time (see results in Section 3.1).

The classification of differentially gene expression is categorized into four statistics: *true positives (TP)*, *false positives (FP)*, *true negatives (TN)* and *false negatives (FN)*. The four statistics are used to determine the quality of the prediction, the classification algorithm provided. In Table 1.1, we explain the meaning of the four statistics in detail.

In the following, we will review the papers our novel method (using GPs to model time shifts, 2.3) is based on:

		Prediction	
		P	**N**
Truth	**P**	*TP*	*FN*
	N	*FP*	*TN*

Table 1.1: The four statements about classification, contrasting the biological **Truth** with the algorithmic **Prediction**. Where P=Positive, N=Negative, T=True and F=False

Detecting Intervals of Differential Expression with Robustness to Outliers. Stegle *et al.* [SDW+09] developed an approach to detect differentially expressed genes in microarray time series data. They used Gaussian processes to manage outliers (i.e., extremely noisy observations) in data and - the most significant step - the intervals in which genes are differentially expressed. To model the intervals in which the genes are differentially expressed, they used an extra GP framework. Furthermore, they considered the noise of each sample not to be overall the same, but to have a probability to be an outlier (to manage noisy observations).

Modeling Transcription Factor Target Detection by Gaussian Processes Models. Honkela et al. [HGG+10] used GPs on microarray time series data in March 2010 to model expression profiles of two TFs, namely, Twist and Mef2. The inputs of the GP are TFs and the outputs are the according genes to be classified as targets or not. Their assumption was to determine target genes of TFs, using simple linear ordinary differential equations to model the translation and transcription activation.

Modeling Time Shifts in Microarray Time Series data. In this thesis, we re-used the Gaussian process implementation ("gpr.py" B.2) of Stegle *et al.* [SDW+09] to learn hyperparameters and regress the data (i.e., we changed regression to that effect of normal distributed priors, see 2.3.2). We newly developed a covariance function ("secfTimeParameter.py" B.2) for modeling time shifts in data (2.3.1).

We took up the idea from Honkela *et al.* [HGG+10] to study TF pairs. Our intention was not to use linear correlations between TFs and their targets, but to analyse time shifts occurring due to a TF deletion. We analysed the correlation between TFs and time shifts, while knocking-out a certain TF of the investigated organism (i.e., *yeast*). See Section 3.2 for the results.

Chapter 2

Gaussian Process Regression for Modeling Microarray Time Series Experiments

In the first section of this chapter, we will present the formal mathematical definitions of microarray time series data that are essential in the following chapters. Second, we will review Gaussian processes in Section 2.2, with respect to applicability. Finally, we will introduce our novel approach modeling time shifts by Gaussian processes in the trailing sections of this chapter.

2.1 Mathematical Definitions for Microarray Time Series

A microarray time series experiment is a $n \times p$ matrix, with n genes $g \in \{1, 2, \ldots, n\}$ and p experiments $r \in \{1, 2, \ldots, p\}$. We assume each condition to have multiple *replicates*. Each replicate is represented as one microarray experiment r and is described over time. Namely, it provides N time points $t \in \{1, 2, \ldots, N\}$. The specific *value* of a time point t in a replicate r of a gene g is $x^g_{r,t}$. Analogously, the *expression level* of the same time point is $y^g_{r,t}$. In the following, we will refer to the *time line* of a gene g as *input* $\mathbf{x}^g_r = (x^g_{r,t} : t = 1, 2, \ldots, N)$ and to its *expression profile* as *output* $\mathbf{y}^g_r = (y^g_{r,t} : t = 1, 2, \ldots, N)$. The whole time series of a replicate r in a gene g is then represented by input/output *pairs*

$$(\mathbf{x}, \mathbf{y})^g_r = \left((x, y)^g_{r,t} : \forall t, t \in \{1, 2, \ldots, N\}\right) . \tag{2.1}$$

In Table 2.1, we illustrate the mathematical definition of microarray time series data.

In most cases of this thesis, we do not refer to a certain gene g. So we will only label a certain gene if necessary and, otherwise, disregard the labeling.

	$r=1$				⋯	$r=p$		
	$t=1$	$t=2$	⋯	$t=N$		$t=1$	⋯	$t=N$
$g=1$	$(x,y)^1_{1,1}$	$(x,y)^1_{1,2}$	⋯	$(x,y)^1_{1,N}$	⋯	$(x,y)^1_{p,1}$	⋯	$(x,y)^1_{p,N}$
$g=2$	$(x,y)^2_{1,1}$	$(x,y)^2_{1,2}$	⋯	$(x,y)^2_{1,N}$	⋯	$(x,y)^2_{p,1}$	⋯	$(x,y)^2_{p,N}$
⋮	⋮	⋮		⋮		⋮		⋮
$g=n$	$(x,y)^n_{1,1}$	$(x,y)^n_{1,2}$	⋯	$(x,y)^n_{1,N}$	⋯	$(x,y)^n_{p,1}$	⋯	$(x,y)^n_{p,N}$

Table 2.1: Illustrates the definition of mathematics of microarray time series data. Each gene g is represented by input/output pairs $(\mathbf{x},\mathbf{y})^g_r = \left((x,y)^g_{r,t} : \forall t, t \in \{1,2,\ldots,N\}\right)$, where $r \in \{1,2,\ldots,p\}$ represents the replicate and t represents the time point the pair refers to.

With respect to differential expression, we always compare two, or more conditions against each other. Each condition $\mathbf{C_k}$ of a microarray experiment is represented by the set of replicates $(\mathbf{x},\mathbf{y})_\mathbf{r}$, the condition refers to. The $\mathbf{k}$-th condition is defined as

$$\mathbf{C_k} = \left\{(\mathbf{x},\mathbf{y})_r : \forall r : r \in \mathbf{k}\right\} \ . \tag{2.2}$$

Note, that the replicates are labeled consecutive, and therefore, the conditions are labeled by the set of replicate indices they refer to. In the following we will refer to the special case of comparing two conditions $\mathbf{C_k}, \mathbf{C_{k'}}$ against each other.

2.2 Gaussian Process (GP) Regression

A gaussian process (GP) is a distribution over functions $f : \mathbf{x} \mapsto \mathbf{y}$. It governs probabilities for all functions $f : \mathbf{x} \mapsto \mathbf{y} = f(\mathbf{x})$ mapping an input $\mathbf{x}$ to the respective output $\mathbf{y}$. In microarray time series data $\mathbf{x}$ refers to the time line and $\mathbf{y}$ refers to the (log) expression level of a gene.

We assume that such a model is applied to individual genes, i.e. genes are assumed to be independent. Additionally, in contrast to many other regression methods the GP method is non-parametric. Non-parametric methods avoid making assumptions on a particular form of f, such as assuming f is linear.

A Gaussian process designates a prior for each function f, assigning higher probabilities to functions favored by the prior. Which functions are favored is implied by the covariance function $k(x,x'|\theta)$, the so-called *kernel* [SS02]. The covariance function allows beliefs, such as smoothness of f, or length scales to be incorporated. Since the GP is a stochastic process it denotes the properties of functions

(assigning the prior). In a regression setting, these properties over functions are then constraint by the observed training dataset $\mathbf{D} = \{(\mathbf{x}, \mathbf{y})_r : r \in \{1, \ldots, p\}$, choosing only functions that are in accordance with these data. The covariance function and hence the GP prior is influenced by hyperparameters θ. Their meaning will be explained in Section 2.2.1.

For the purpose of this thesis, we mainly take advantage of two central results. First, given a Gaussian process prior with a covariance function and observed training data $\mathbf{D}$, we can make predictions at previously unseen inputs $\mathbf{x}^*$. The GP not only provides point estimates for the most likely outputs $\mathbf{y}^*$, but also yields uncertainties. With Gaussian noise of variance σ^2 the posterior probability of $\mathbf{y}^*$ is Gaussian distributed,

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{D}, \theta) = \mathcal{N}\big(\mathbf{y}^*|\underline{\mathbf{y}}^*, \mathrm{cov}(\mathbf{y}^*)\big), \quad \text{where} \tag{2.3}$$

$$\underline{\mathbf{y}}^* = K_{\mathbf{x}^*,\mathbf{x}}[K_{\mathbf{x}} + \sigma^2\mathbf{E}]^{-1}\mathbf{y} \ , \tag{2.4}$$

$$\mathrm{cov}(\mathbf{y}^*) = K_{\mathbf{x}^*} - K_{\mathbf{x}^*,\mathbf{x}}[K_{\mathbf{x}} + \sigma^2\mathbf{E}]^{-1}K_{\mathbf{x},\mathbf{x}^*} \ , \tag{2.5}$$

and $\mathbf{E}$ denotes the identity matrix.

Equations taken from [RW05, Equations (2.22),(2.23),(2.24)].

Second, a Gaussian process prior allows alternative models to be compared. For example, we can access the likelihood of a particular covariance function k or particular hyperparameter settings. The quantity

$$\log p(\mathbf{y}|\mathbf{x}, \theta) = -\frac{1}{2}\mathbf{y}^\top\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log 2\pi, \quad \text{where} \tag{2.6}$$

$$\mathbf{K} = K_{\mathbf{x}} + \sigma^2 \ , \tag{2.7}$$

Equation taken from [RW05, Equation (2.30)].

assigning the probability of the observed data $\mathbf{x}, \mathbf{y}$, given the covariance function k and parameters θ is of key interest to calculate Bayes factors, which will be used in the following. Note that the matrix $\mathbf{K}$ is given by evaluating $k(x, x')$ for all pairs of x and x' ($x, x' \in \mathbf{x}, \mathbf{x}*$), giving rise to a $N \times N$ matrix.

For more detailed information about GPs please refer to the book of [RW05].

2.2.1 Covariance function (CF)

The covariance function (CF) $k(x, x'|\theta)$ of a GP defines how much two outputs y, y' covary, in dependence of its $s = |\theta|$ hyperparameters $\theta \in \mathbb{R}^s$. This covariance also

depends on the distance between the inputs x, x' and valid covariance functions span a large space. There is only one constraint for a function over two inputs x, x' to be a covariance function. The covariance matrix $K_{\mathbf{x}}$ created from the function values $k(x, x'|\theta) : \forall x, x' \in \mathbf{x}$ has to be positive definite for any input $\mathbf{x}$ and any set of hyperparameters θ. For more covariance functions please consider the books of [RW05, Nea96] or the thesis [Gib97]. Here, we will focus on the squared exponential (SE) covariance function, because it is commonly used and can be intuitively understood. The SE perhaps is the most widely used CF. Our new covariance function, which accounts for time shifts, will also be based on the SE.

Squared Exponential Covariance Function

The squared exponential covariance function (SE)

$$k\big(x, x'|\theta = (A, L, \sigma)\big) = A^2 \cdot \exp\left(-\frac{\|x - x'\|^2}{2L^2}\right) + \sigma^2 \quad , \tag{2.8}$$

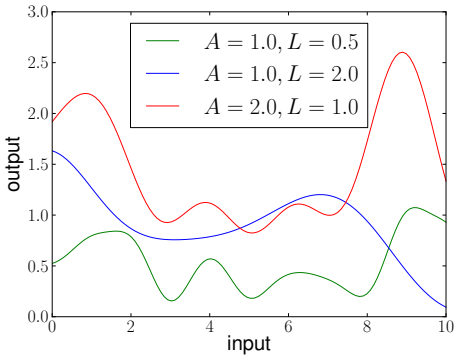
defines how much two outputs y, y' covary with respect to the length scale parameter L, the amplitude parameter A and a noise level σ. These hyperparameters $\theta_{SE} = (A, L, \sigma)$ influence regression behavior and can be flexibly adjusted. For the SE the three hyperparameters have the following intuitive meaning:

- A the output amplitude variance. It defines the amplitude of the correlation between y and y'. Therefore, it influences the point-wise standard deviation of the output functions f. A directly correlates with the amplitude (i.e., a large A implies large amplitude).
- L the output longitude variance. It defines the typical length scale of inputs x and x' such that the outputs show covariation. It's value directly correlates with the smoothness of the regressed function.
- σ observation noise. Large noise values define a large allowed error for functions fitted to training data.

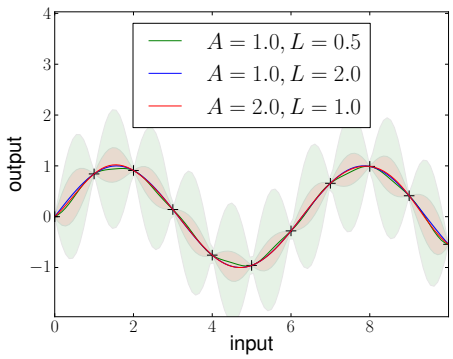
It is important to note that hyperparameters have not to be chosen arbitrary, and instead, can be learned from data. In Figure 2.1, we illustrate the influence of the hyperparameters on the regressed functions.

2.2.2 Learning Hyperparameters

For GP regression one has to determine the most probable set of hyperparameters $\theta \in \mathbb{R}^s$, to match the training data set $\mathbf{D} = (\mathbf{x}, \mathbf{y})$. The log marginal likelihood



(a) Three random examples drawn from SE covariance function with varying hyperparamters $\theta = (A, L)$, without noise σ.



(b) Sparse samples drawn from sinus function and regressed using GP with different hyperparameters $\theta = (A, L)$, without noise.

Figure 2.1: Three examples drawn from a GP with different hyperparameters for SE covariance function without noise. The shaded area shows for each direction two times the point-wise squared standart deviation $2 \cdot sd^2$ of the predicted function values $\mathbf{y}^*$ (which results in a total of $2(\cdot 2 \cdot sd^2)$). This area equates the 95% confidence interval for the joint distribution over the functions f of the GP [RW05].

$\log p(\mathbf{y}|\mathbf{x}, \theta)$, given the inputs $\mathbf{x}$ and the current hyperparameters θ can be written as:

$$\log p(\mathbf{y}|\mathbf{x}, \theta) = -\frac{1}{2}\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log 2\pi \quad , (2.6) \tag{2.9}$$

where $\mathbf{K} = \mathbf{K_x} + \sigma\mathbf{E}$ is the covariance matrix with noisy observations. We learn the most probable set of hyperparameters by denoting

$$\hat{\theta} = \arg\max_{\theta}\{\log p(\mathbf{y}|\mathbf{x}, \theta) + \log p(\theta)\} \quad , \tag{2.10}$$

where $p(\theta)$ is the hyperprior probability for the hyperparameters θ (see Section 2.2.3).

2.2.3 Hyperpriors

To control hyperparameters, hyperpriors have been introduced, providing probabilities $p(h)$ for each hyperparameter $h \in \theta$. The impact of hyperparameters θ on the predicted function values $\mathbf{y}^*$ can be seen at figure 2.1(a). The hyperpriors are appropriate to restrict the most probable set of hyperparameters $\hat{\theta}$ on preferable

regions. The most probable set of hyperparameters is defined as

$$\hat{\theta} = \arg\max_{\theta}\{p(\theta|\mathbf{D})\} \tag{2.11}$$

$$= \arg\max_{\theta}\{p(\mathbf{y}|\mathbf{x},\theta)\cdot p(\theta)\} \tag{2.12}$$

$$= \arg\max_{\theta}\{\log\ p(\mathbf{y}|\mathbf{x},\theta) + \log\ p(\theta)\}\ , \tag{2.13}$$

(compare 2.10)

where the hyperparameters $h \in \theta$ maximize the probability for the function values $\mathbf{y}^*$ to describe the given outputs $\mathbf{y}$ considering the inputs $\mathbf{x}$. To restrict the hyperparameters, the hyperprior probability

$$\log p(\theta) = \log \prod_{h\in\theta} p(h) = \sum_{h\in\theta} \log p(h) \tag{2.14}$$

is added to the logarithmic probability of the predicted function values $\mathbf{y}^*$, as defined in Section 2.2.2.

2.3 Time Shift Sensitive GP Regression: GPTimeShift

To regress microarray time series data assuming time shifts between replicates, it is desired to model these time shifts explicitly. In the parlance of regression, instead of searching functions $f : \mathbf{x} \mapsto \mathbf{y} = f(\mathbf{x})$, we now search functions $f : \mathbf{x} \mapsto \mathbf{y} = f(\mathbf{x}-\delta T)$, mapping the time line to the expression level of a gene, considering time shifts δT for each replicate in data. Within the GP model, we represent the time shift δT as additional hyperparameters

$$\mathbf{T} = (T_r : \forall r, r \in \{1, 2, \ldots, p\})\ , \tag{2.15}$$

where $T_1, T_2, \ldots, T_p$ represent one time shift hyperparameter per replicate $(\mathbf{x},\mathbf{y})_{\mathbf{r}}$ $(\mathbf{2.1})$.

After learning the individual time shift hyperparameters, one per replicate within and between each condition, it is useful to create a summary time shift δT. To interpret the effective shift between two conditions $\mathbf{C_k}$ and $\mathbf{C_{k'}}$, we use

$$\delta T_{\mathbf{k},\mathbf{k'}} = \sum_{r\in\mathbf{k'}} \frac{T_r}{|\mathbf{C_{k'}}|} - \sum_{r\in\mathbf{k}} \frac{T_r}{|\mathbf{C_k}|} \tag{2.16}$$

throughout this thesis.

Additionally, we require a novel covariance function to take the time shift into

account explicitly. In other words the covariance function is able to deal with time shift hyperparameters $\mathbf{T}$, subtracting the time shift parameters $T_r \in \mathbf{T}$ from their respective replicates $(\mathbf{x}, \mathbf{y})_\mathbf{r}$.

2.3.1 Squared Exponential Covariance Function with Time Shift Parameter

We provide for each replicate $(\mathbf{x}, \mathbf{y})_\mathbf{r}$ one time shift hyperparameter $T_r \in \mathbf{T}$. These time shifts allow the GP regression to vary the time line $\mathbf{x}_r$ of the respective replicates $r \in \{1, \ldots, p\}$. As already denoted by its name, the squared exponential covariance function with time shift parameter (SETP) is based on the squared exponential covariance function, which is described in Section 2.2.1. Our new set of hyperparameters $\theta_{SETP} = (A, L, \mathbf{T}, \sigma)$, now contains the new hyperparameters $\mathbf{T} = (T_1, T_2, \ldots, T_p)$. The SETP is defined as:

$$k(x_r, x'_{r'}) = A^2 \cdot \exp\left(-\frac{d^2}{2L^2}\right) + \sigma \quad , \tag{2.17}$$

$$d = \left|\left|(x - T_r) - (x' - T_{r'})\right|\right| \quad , \tag{2.18}$$

where T_r denotes the time shift of the replicate x_r, and similarly $T_{r'}$ denotes the time shift of the replicate $x'_{r'}$.

2.3.2 Implementation details

Learning Hyperparameters. To determine the most probable set of hyperparameters $\hat{\theta}$, we use a gradient optimizer. These efficient optimization methods require the function to be minimized, here $-\log p(\mathbf{y}|\mathbf{x}, \theta)(2.6)$, as well as it's gradients $\partial/\partial h \; k(x, x')$ with respect to the hyperparameters $h \in \theta$. Hyperparameters that are constrained to be positive, such as the amplitude A, the length scale L and the noise level σ, are optimized in the log space. For these the hyperparameter inputs of the optimizer are defined on the real line, i.e. $\theta^O = \exp(\theta)$; where the optimization takes place over θ. As the time shift hyperparameters $\mathbf{T}$ can be positive or negative, these hyperparameter values are optimized directly. The optimization of this log marginal likelihood is now additive, rather than multiplicative. The full set of analytic derivatives of the kernel function is given in Appendix A.2.

Hyperprior for Time Shifts. In Section 2.2.2, we explained how to learn hyperparameters θ from data. The definition of the probability $p(\theta|\mathbf{D})$ for the hyperparameters θ to explain the data set $\mathbf{D}$ does not change due to our new set of hyper-

parameters θ_{SETP}. In this thesis, we use the $\Gamma(\mathbf{x}, k, t)$ probability density function for the probabilities of the hyperpriors A, L and σ, and the Gaussian probability density function for time shifts $\mathbf{T}$, as we assume time shifts to be both and negative. Therefore, it is crucial that the optimizer does not take the logarithm of the hyperparamters $\mathbf{T}$.

$\Gamma(\mathbf{x}, k, t)$ probability density function:
The $\Gamma(\mathbf{x}, k, t)$ probability density function is defined as:

$$\Gamma(\mathbf{x}, k, t) = \frac{t^{-k} \cdot \mathbf{x}^{k-1}}{\Gamma(k)} \cdot \exp\left(\frac{-\mathbf{x}}{t}\right) \quad , \tag{2.19}$$

where $\Gamma(k) = k!$ is the Gamma function [SG02]. Three probability density functions drawn from $\Gamma(\mathbf{x}, k, t)$ with different scale and shape parameter are shown in Figure 2.2.

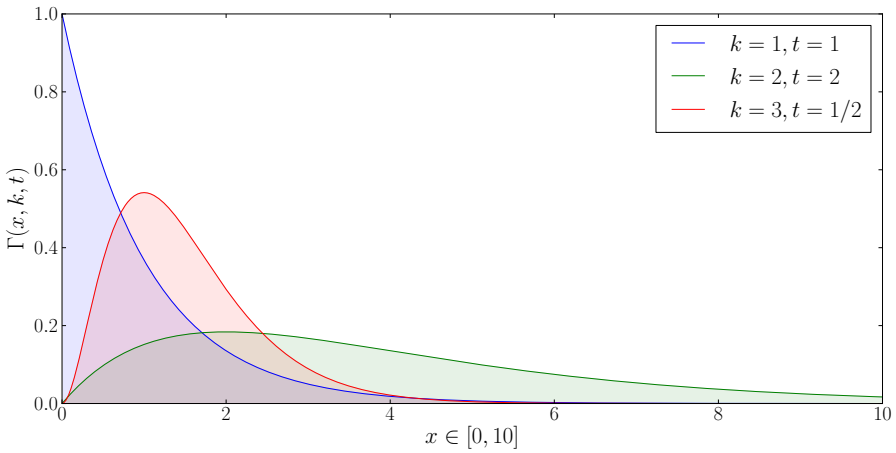


Figure 2.2: Probability density function $\Gamma(\mathbf{x}, k, t)$ for different scale and shape parameter.

Gaussian probability density function:
The Gaussian probability density function is defined as:

$$\mathcal{N}(\mu, \sigma) = \frac{\exp(-\frac{(x-\mu)^2}{2\sigma^2})}{\sqrt{2\pi}\sigma} \quad , \tag{2.20}$$

with mean μ and standard deviance σ.

2.3.3 Example Application of GPTimeShift

To shift the data, each time shift hyperparameter T_r is subtracted from the respective replicate $(\mathbf{x}, \mathbf{y})_r$ it corresponds to. The effect of such a time shift on a simulated data set, is shown in Figure 2.3. Note, that the simulated data set provides optimal time shift behavior.

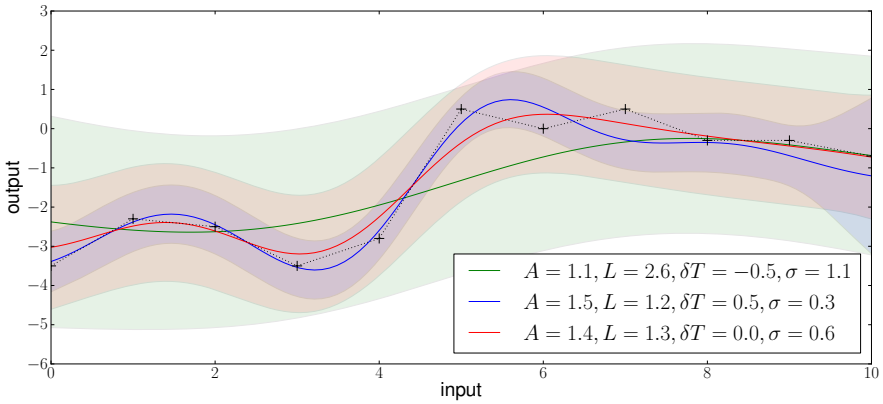


Figure 2.3: The effect of a time shift in the input $\mathbf{x}$ of a data set $\mathbf{A}$ on the regression of the corresponding outputs $\mathbf{y}$. In this particular example, we adjusted the time shift δT manually and let the regression learn all other hyperparamters A, L and σ. As you can see the data fits best with a time shift of $\delta T = 0.5$.

2.4 Bayes Factors for Model Comparison

The main application of the time shift invariant regression approach is the comparison of alternative hypotheses. In the setting of differential gene expression, we provide Bayes factors, comparing the probabilities of two different hypotheses:

The **first hypothesis** is that the expression profiles of a gene in two, or more, conditions are more likely described by individual GP models $\mathcal{H}^{\mathcal{I}}_{\delta T}$, one for each condition $C_{\mathbf{k}}$. The probability for this first hypothesis, in the special case of two conditions $\mathbf{C_k}$ and $\mathbf{C_{k'}}$ is

$$p(\mathbf{C_k}|\mathcal{H}^{\mathcal{I}}_{\delta T}, \hat{\theta}^{\mathbf{T}}_{\mathcal{I}}) \cdot p(\mathbf{C_{k'}}|\mathcal{H}^{\mathcal{I}}_{\delta T}, \hat{\theta}^{\mathbf{T}}_{\mathcal{I}}) \ , \tag{2.21}$$

where $\hat{\theta}^{\mathbf{T}}_{\mathcal{I}} = (A, L, \mathbf{T}, \sigma)$ are the most probable hyperparameters for expression levels from condition $\mathbf{C_k}$ and $\mathbf{C_{k'}}$ respectively (see Section 2.2.2).

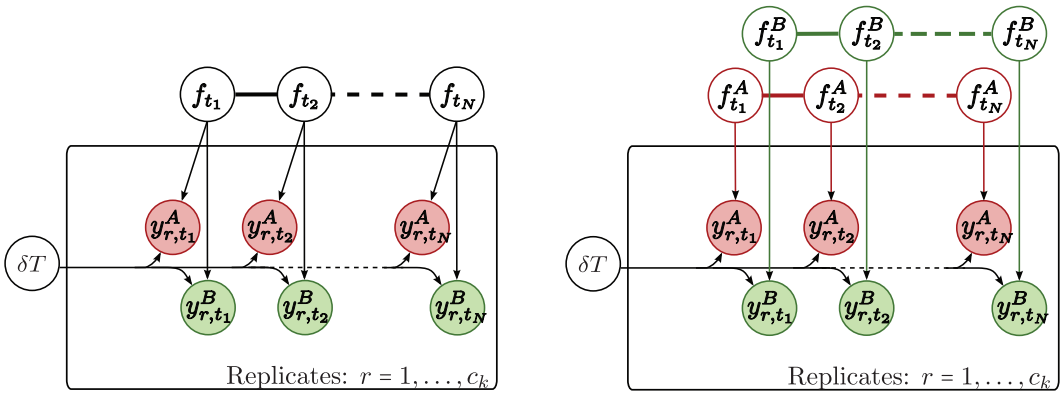
The **second hypothesis** is that the expression profiles of genes are more likely described by one GP model $\mathcal{H}^{\mathcal{S}}_{\delta T}$. Intuitively, this can be thought of as using a single

regressor for both conditions, assigning the probability

$$p(\mathbf{C_k} \cup \mathbf{C_{k'}} | \mathcal{H}_{\delta T}^{\mathcal{S}}, \hat{\theta}_{\mathcal{S}}^{\mathbf{T}}) \tag{2.22}$$

to observed data. Here, $\hat{\theta}_{\mathcal{S}}^{\mathbf{T}}$ are the most probable hyperparameters $\theta_{\mathcal{S}}^{\mathbf{T}} = (A, L, \mathbf{T}, \sigma)$ to regress the both conditions $\mathbf{C_k}$ and $\mathbf{C_{k'}}$ in one shared common GP model.

An illustration of these two hypotheses is given at Figure 2.4.



(a) *shared* model (b) *individual* model

Figure 2.4: The two models of GP regression. a) The shared model regresses the data outputs $\mathbf{y}^{\mathbf{A},\mathbf{B}} = (y_{r,t_1}^{\mathbf{A},\mathbf{B}}, \ldots, y_{r,t_N}^{\mathbf{A},\mathbf{B}})$ by one GP framework $\mathbf{f} = (f_{t_1}, f_{t_2}, \ldots f_{t_N})$ and corresponding inputs $\mathbf{x} = (t_1, \ldots, t_N)$. b) To individually regress each data set, instead, we provide two separate GP frameworks $\mathbf{f}^{\mathbf{A},\mathbf{B}} = (f_{t_1}^{\mathbf{A},\mathbf{B}}, f_{t_2}^{\mathbf{A},\mathbf{B}}, \ldots f_{t_N}^{\mathbf{A},\mathbf{B}})$ for each output $\mathbf{y}^{\mathbf{A},\mathbf{B}} = (y_{r,t_1}^{\mathbf{A},\mathbf{B}}, \ldots, y_{r,t_N}^{\mathbf{A},\mathbf{B}})$ and corresponding input $\mathbf{x} = (t_1, \ldots, t_N)$. Figure taken from [SDW+09, Figure 2]

With regard to detecting time shifts, we provide three comparing Bayes factors, scoring in the quality of the time shift in addition to differential expression. First, we provide the Bayes factor $\mathcal{S}_{\delta T}^{\mathcal{I}}$, which compares the individual hypothesis with the shared hypothesis, both including the time shift hyperparameters $\mathbf{T}$.

$$\mathcal{S}_{\delta T}^{\mathcal{I}} := \log \frac{p(\mathbf{C_k} | \mathcal{H}_{\delta T}^{\mathcal{I}}, \hat{\theta}_{\mathcal{I}}^{\mathbf{T}}) \cdot p(\mathbf{C_{k'}} | \mathcal{H}_{\delta T}^{\mathcal{I}}, \hat{\theta}_{\mathcal{I}}^{\mathbf{T}})}{p(\mathbf{C_k} \cup \mathbf{C_{k'}} | \mathcal{H}_{\delta T}^{\mathcal{S}}, \hat{\theta}_{\mathcal{S}}^{\mathbf{T}})} \quad . \tag{2.23}$$

Second, we define the Bayes factor $\mathcal{S}_0^{\mathcal{I}}$ that compares the same hypotheses as the first, with the difference that in the shared model all time shift hyperparameters $T_r \in \mathbf{T}$ are set to zero, instead of learning them from data. This corresponds to the assumption that times shifts are not present.

$$\mathcal{S}_0^{\mathcal{I}} := \log \frac{p(\mathbf{C_k} | \mathcal{H}_{\delta T}^{\mathcal{I}}, \hat{\theta}_{\mathcal{I}}^{\mathbf{T}}) \cdot p(\mathbf{C_{k'}} | \mathcal{H}_{\delta T}^{\mathcal{I}}, \hat{\theta}_{\mathcal{I}}^{\mathbf{T}})}{p(\mathbf{C_k} \cup \mathbf{C_{k'}} | \mathcal{H}_0^{\mathcal{I}}, \hat{\theta}_{\mathcal{S}}^{(0,\ldots,0)})} \quad . \tag{2.24}$$

Finally, the Bayes factor $\mathcal{S}^{\mathcal{S}}$ comparing the two shared hypotheses, one with and the other without time shift (i.e., set all time shift hyperparameters $T_r \in \mathbf{T}$ to zero) is given by:

$$\mathcal{S}^{\mathcal{S}} := \log \frac{p(\mathbf{C_k} \cup \mathbf{C_{k'}} | \mathcal{H}^{\mathcal{I}}_{\delta T}, \hat{\theta}^{\mathbf{T}}_{\mathcal{S}})}{p(\mathbf{C_k} \cup \mathbf{C_{k'}} | \mathcal{H}^{\mathcal{I}}_{0}, \hat{\theta}^{(0,\ldots,0)}_{\mathcal{S}})} \tag{2.25}$$

Note that for the sake of comparability between the Bayes factors, all individual hypotheses models are provided with time shifts between the replicates.

Comparing the two hypotheses with the three Bayes factors defined above provides following intuitionally interpretation. The hypothesis stated in the numerator of the Bayes factor is more probable for the Bayes factor being greater than zero. Similarly, the denominator hypothesis is more probable, if the Bayes factor is less than zero. Namely, $\mathcal{S}^{\mathcal{S}}$ being greater than zero indicates the hypothesis with time shift hyperparameters $\mathbf{T}$ to be more probable than the hypothesis without $\mathbf{T}$.

Chapter 3

Applications

We applied GPTimeShift in two ways. We will first present the results applying GPTimeShift to study the improvements a time shift can provide, including it into differential gene expression detection. Second, we will present the results, using GPTimeShift to analyse time shifts of pairs of transcription factors (TFs).

3.1 GPTimeShift for Differential Gene Expression Detection

Plant stress response involve significant transcriptional changes in the regulatory networks of the plant. We studied the stress response of *Arabidopsis thaliana*, infected with fungal pathogen *Botrytis cinerea*. Detached leaves of *Arabidopsis thaliana* where inoculated with *Botrytis cinerea* spores, harvested every 2h up to 48 h (i.e. a total of 24 time points). *Botrytis cinerea* penetrates the leaves and causes expanding necrotic regions. Full genome expression profiles were generated from four plants in both conditions (i.e., *control*, not inoculated and *decease*, infected by the fungal pathogen). We applied GPTimeShift for each probe to the time courses of both conditions and all four plants.

Our goal is to elucidate the changes in the time line of the regulatory networks of *Arabidopsis thaliana*. Exposing the plant to the pathogen *Botrytis cinerea* causes the plant to react. We assume this reaction to involve time shifts in the expression levels of genes, maybe without changing the shape of the profiles (i.e., the shape of the expression profile may not change, but there is a time shift between the conditions). Our assumption is based on the organism to react more slowly, because it has to deal with the pathogen first, and then continue normal living activities. Therefore, we applied GPTimeShift to prevent regression from falsely detect those time shifted genes as differentially expressed.

We chose following hyperpriors restricting GP from learning unpleasant hyperparameters:

$$p(A) := \Gamma(A, 6, 0.3) \tag{3.1}$$

$$p(L) := \Gamma(L, 30, 0.1) \tag{3.2}$$

$$p(T) := \mathcal{N}(0, 0.25) \ \forall T \in \mathbf{T} \tag{3.3}$$

$$p(\sigma) := \Gamma(\sigma, 1, 0.3) \ . \tag{3.4}$$

In Figure 3.1, we illustrate these hyperpriors. Applying GPTimeShift with this constellation of hyperpriors to the full genome profiles of *Arabidopsis thaliana* provided the following results.

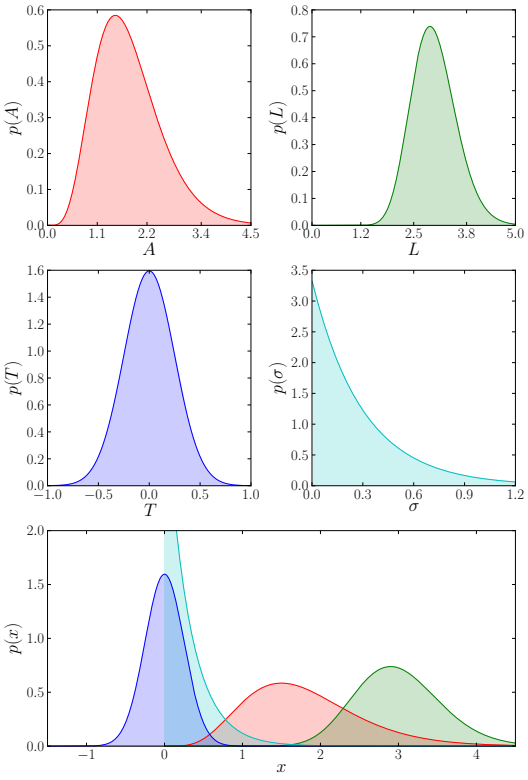


Figure 3.1: Hyperprior probability densities for hyperparameters $A, L, \mathbf{T}$ and σ. The four distributions are plotted in one plot respectively (compare upper four subplots). For a better comparability, the lower subplot shows the hyperprior distributions; all together in one plot.

Stegle *et al.* provided us with a subset of 2000 randomly chosen genes of measured time courses. They asked a human expert to manually label each probe as either 'differentially expressed', 'not differentially expressed', or 'dubious case'. Excluding the dubios cases, we used the 1890 labeled probes as biological **Truth** to compare our results to. We applied GPTimeShift and a GP standard model, which not includes a time shift in regression to this labeled set of genes. Figure 3.2 shows the resulting area under the ROC (AUC) curve of both, GPTimeShift and the GP standard model. GPTimeShift provides an AUC of 0.986, and shows little improvement to the AUC of the standard model (GPStandard, AUC=0.985). We will clarify this insignificant improvement in our conclusions 4.1.

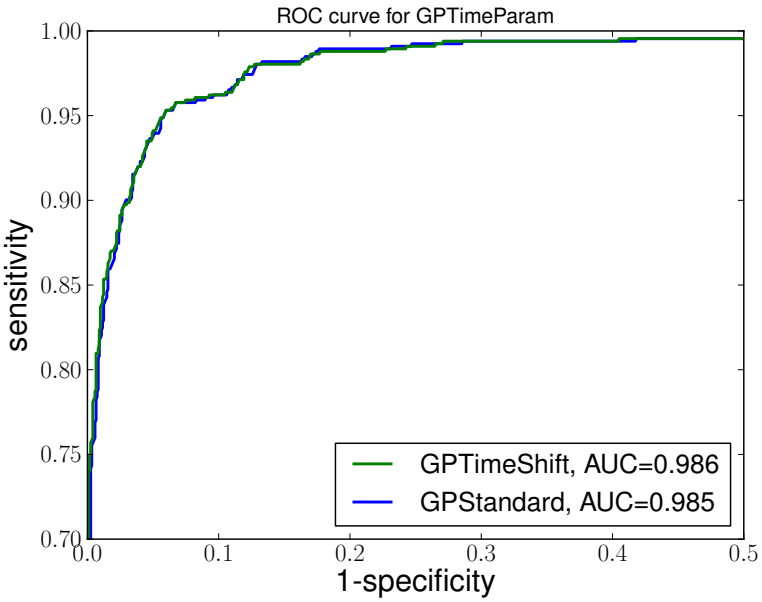


Figure 3.2: AUC analysis of GPTimeShift against GP standart

3.2 GPTimeShift for TF Pairs and TF Target Detection

We applied GPTimeShift on data produced by knock-out experiments of yeast populations. Mata *et al.* knocked-out four different genes and generated full genome expression profiles for both conditions (i.e., *WT* and *mutant*) and all four knock-out experiments every 1h up to 8h (i.e., a total of 8 time points). The four knocked-out genes are *rep1, mei4, atf21* and *atf31*, which encode for known transcription factors (TFs) controlling sexual differentiation [MWB07]. We focused on analyzing the experiment that shows the expression profiles of knocking out *mei4*. In the following we refer to the knock-out treatment as *mutant* and to the wild-type as *WT*. In addition to this we used a composition of pairs of known TFs. We used this composition as biological truth, to compare our results against.

3.2.1 Hyperprior selection

Given sufficient data, (i.e., somewhat more than three replicates) hyperparameters can perfectly be learnt from this data without hyperpriors. These are necessary if the data is sparse. The time series of the replicates of this particular data set are unfortunately already averaged, so there is only one replicate per condition to regress over. Therefore, we chose strict hyperpriors to prevent regression from learning wrong hyperparameters. We used visually illustrations (i.e., plots of the expression profiles) of some specimen of data to determine the right hyperpriors. We discovered the noise of the specimen to be strict within the open interval $]0.2, 0.5[$. Therefore, we used a strict prior for the noise σ and let the GP learn the others (i.e., hyperpriors for A, L and $\mathbf{T}$ are rather loose). The hyperpriors for GPTimeShift in this

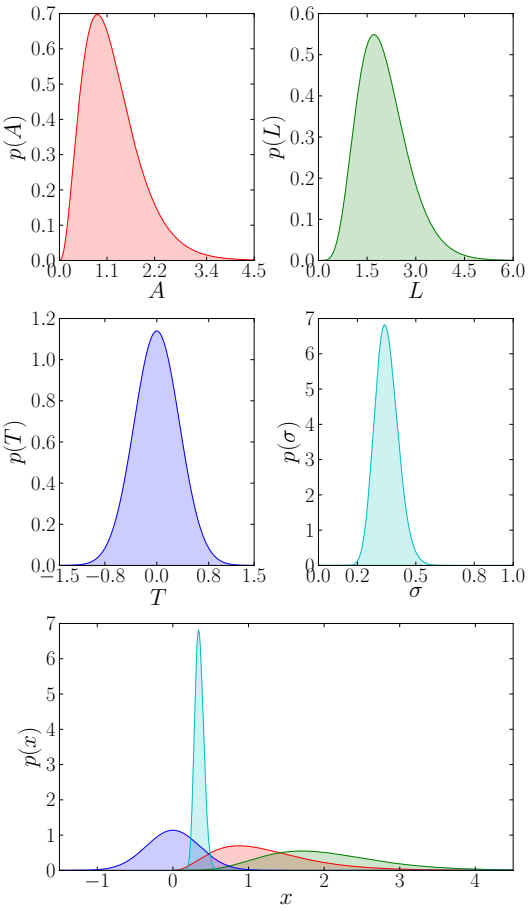


Figure 3.3: Hyperprior probability densities for hyperparameters $A, L, \mathbf{T}$ and σ. The four distributions are plotted in one plot respectively (compare upper four subplots). For a better comparability, the lower subplot shows the hyperprior distributions; all together in one plot.

particular applications are chosen as:

$$p(A) := \Gamma(A, 3.5, 0.35) \tag{3.5}$$

$$p(L) := \Gamma(L, 6.7, 0.3) \tag{3.6}$$

$$p(T) := \mathcal{N}(0, 0.35) \ \forall T \in \mathbf{T} \tag{3.7}$$

$$p(\sigma) := \Gamma(\sigma, 35, 0.01) \ . \tag{3.8}$$

Figure 3.3 illustrates the impact of the different hyperpriors. We have plotted the distributions, respectively (i.e., the first four subplots), and all together in one plot, for the purpose of comparability (i.e., the lowest subplot). The hyperprior distribution gives the probability $p(h)$ for a hyperparameter h (compare 2.2.3).

3.2.2 Detecting TF pairs by Time Shift Analyses

We used the dataset for *mei4*. It contains the mean values over the *WT's* and *mutant's* time series. The *mutant* is *mei4p*. We focus at variation of time shifts of *mei4* in the course of knockout reaction, rather than sexual differentiation.

Synchronizing *yeast* cells in their cell-cycle activity is difficult, because each cell is an individual organism (i.e., the tempo of each cell's cell-cycle activity slightly differ from each other). Additionally, the fact, that in one population one TF (*mei4*) is knocked-out implies a time shift in data (compare 3.1). So, in all probability a time shift occurs between the conditions $\mathbf{C_k}, k = 1, 2$ of data. To compensate these time shifts we allow the GP model to learn the hyperparameters $\mathbf{T}$ in data for the individual model (see Section 2.4).

To understand the following results, please regard Figure 3.4.

As we search for time shifts between the time lines in *WT* and *mutant* of pairs of TFs, the interpretation of the Bayes factors (2.4) slightly change. We provide two sets of Bayes factors (i.e., $\mathcal{S}^{\mathcal{I}}_{\delta T}, \mathcal{S}^{\mathcal{I}}_{0}$ and $\mathcal{S}^{\mathcal{S}}$) for each comparison of a pair of TFs. One set for the comparison of the *WT* time lines and the other for the comparison of the *mutant* time lines. As we compair different genes against each other, we have to make sure that they are at least in one condition (*WT* or *mutant*) correlated. Now, with respect to this comparisons an interesting outcome of a gene pair is that both (*WT* and *mutant*) conditions are strongly correlated and in one condition a time shift is involved, while non in the other. So, one of the pairs gives reaction on the perturbation of the other in the form of a time shift, or, analogously, the time shift is gone, due to perturbation.

To give a more intuitive explanation of how we compair the TF pairs, we illus-

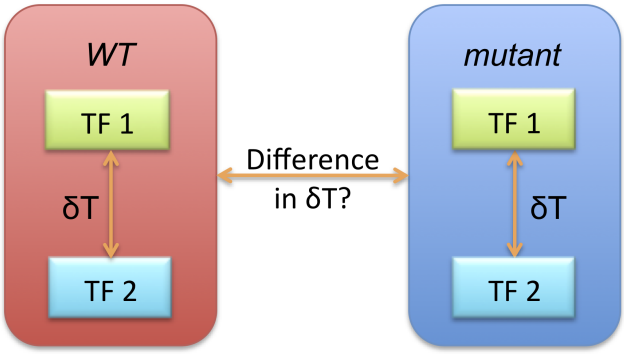
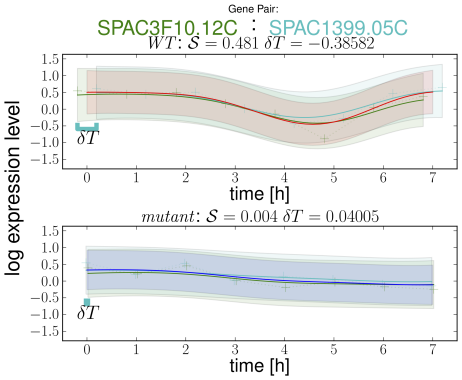


Figure 3.4: Illustration of how to compare TF pairs in *WT* and *mutant*, respectively. We regress the time series (of both TFs) in *WT* and *mutant* against each other. Thereafter, we can compare the time shifts δT of both conditions against each other. A difference between two time shifts declares for example that the two TFs differ in their time course after being perturbated. Although it is not clear, which one.

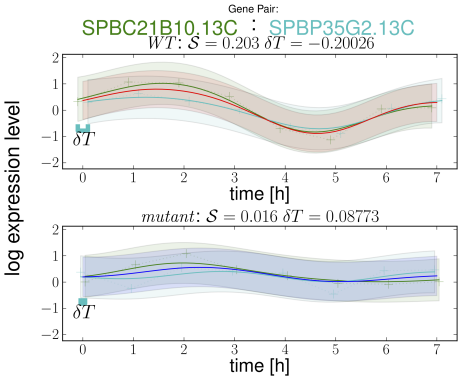
trate in Figure 3.5 the structure of some sample TFs that are correlated in *WT* and *mutant* with a difference in their δT. Obviously, the *WT* and *mutant* pairs correlate to each other. The inputs $\mathbf{x}_r \in \mathbf{C_k}, k = 1, 2$ of the pairs are shifted against each other in time.

We now can distinguish between genes incorporated into the perturbation difference and genes that are not. In biological words we can denote the organism's reactions on the missing TF. As we will see the overall reaction is as predicted: most TFs are extinguished, due to perturbation. By denoting pairs that differ in *WT* and *mutant*, we highly probable denote genes that are involved in the stress reaction of perturbation of *mei4*. The genes are reacting with time shifts, possibly due to a failing TF. Thus, the TF pairs which are correlated in both conditions (both negative scores $-\mathcal{S}^{\mathcal{I}}_{\delta T}$ are greater than a threshold of -2.5) are of interest. The threshold is chosen as the maximum of the distribution of $-\mathcal{S}^{\mathcal{I}}_{\delta T}$ scores. See Figure 3.6(a) for these TFs. After denoting the TFs, that are correlated (green dashed rectangle) we have to denote the right TF pairs. We assume, that there is a certain noise in the time shifts, so only the tails of the distribution of the δT differences are of interest. See figure 3.6(b) for the distribution of the differences in δT of the TF pairs. The genes involved in the stress reaction (tails of the distribution of differences in δT) are listed in the csv file "yeastPairsdTpositive.csv", providet for download on my website `http://people.kyb.tuebingen.mpg.de/maxz/`. Note that all genes are correlated and not differentially expressed, as we still compare different genes to each other. See Section 4.1 for our conclusions about the detected TF pairs.

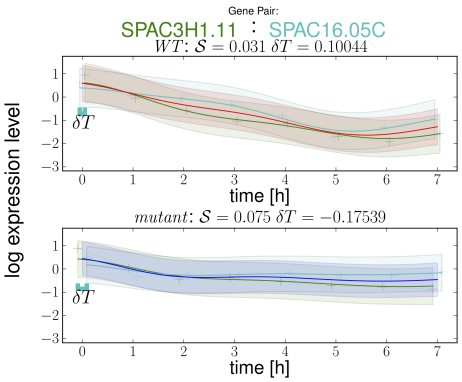
We predict the perturbation of *mei4* leads other genes to be extinguished. If



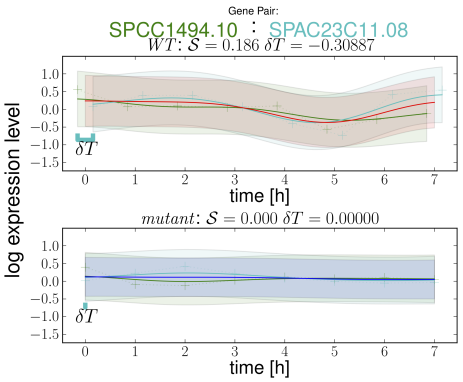
(a) SPAC1399.05C is regulated later in *WT*. In *mutant*, there is no difference in expression level of the two genes.



(b) Both genes are down-regulated at the end in *WT* but there is no difference in their δT. These genes are obviously genes that are deactivated due to perturbation but we do not filter these.

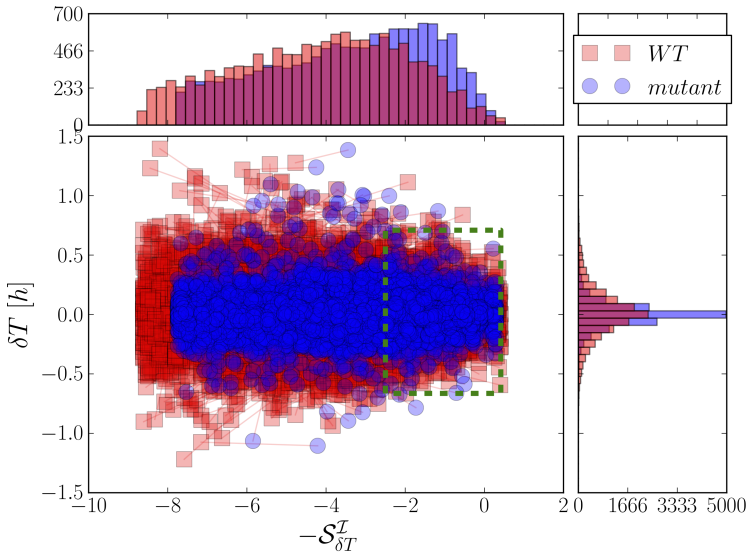


(c) Perturbation sparsely influences the gene's expression level. These genes are less probable introduced in the stress reaction of the *yeast* cells.

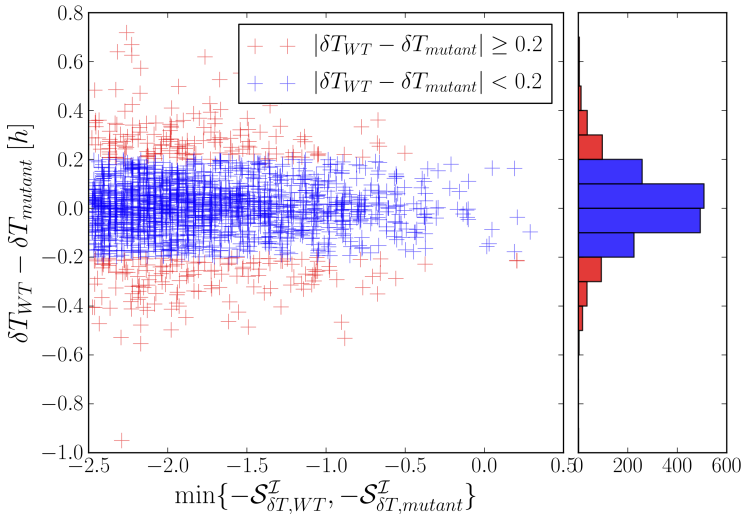


(d) The time shift is gone due to perturbation.

Figure 3.5: Some examples showing the correlated TF pairs in their *WT*(red) and *mutant*(blue) replicates and their respective δT*s*. The green regression line shows the first TF and the cyan the second.



(a) δT against $-\mathcal{S}^{\mathcal{I}}_{\delta T}$. The TF pairs are shown as red squares (*WT*) and blue dots (*mutant*) connected by a red line. The pairs of interest are shown in the green rectangle.



(b) Distance of δT between the pairs of the TFs versus the minimum $\min\{-\mathcal{S}^{\mathcal{I}}_{\delta T}(WT), -\mathcal{S}^{\mathcal{I}}_{\delta T}(mutant)\}$ of the scores. If the minimum of the scores is greater than the threshold -2.5 WT and $mutant$ are both correlated. The absolute distance of their δT has to be greater then the threshold shown in the figure's legend.

Figure 3.6: Data analysis of TF pairs.

the genes are extinguished we can highly probable say that the genes will no more show a time shift against other (maybe also extinguished) genes at all. In summary the time shift δT of all genes will drastically decrease in *mutant*. We discovered this fact in the data. Please have a look at Figure 3.7 for the resulting and explaining plot. There, we can see the distributions of both time shifts δT_{WT} and δT_{mutant} in *WT* and *mutant*. The fact, that the distribution of δT_{mtuant} is more likely to be zero, indicates that a set of genes is extinguished.

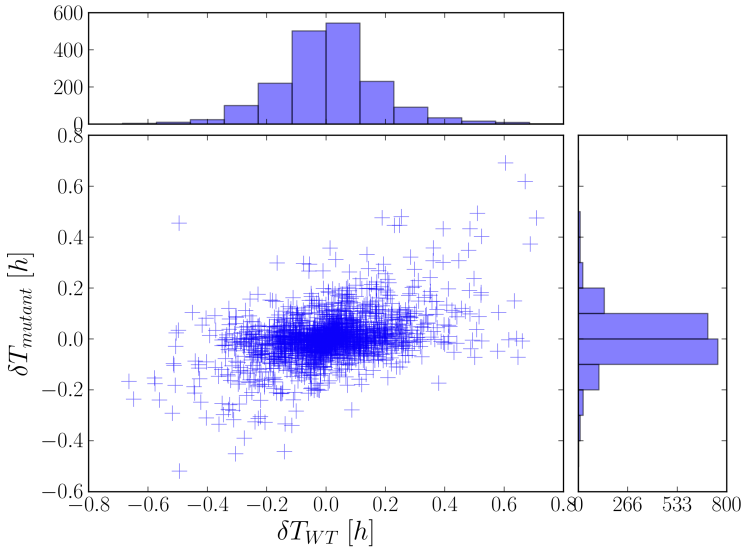


Figure 3.7: All gene's time shifts δT_{WT} and δT_{mutant} in *WT* and *mutant* conditions. The distribution of δT_{mutant} indicates the extinguished genes due to perturbation of *mei4*. There is much less activity in time shifts δT_{mutant}

Chapter 4

Conclusions

The method of this thesis - GPTimeShift - provides an effective improvement on GP regression applied to microarray time series data. Additionally, GPTimeShift provides new perspectives on time series data (i.e., we are prepared to analyse time series in terms of time shifts). Uncloaking time shifts in gene expression shows us (since now) concealed time courses of regulatory networks. This is a good step to our ultimate goal to elucidate such regulatory networks.

4.1 GP Model with Time Shift

Applying GPTimeShift to detect differentially expressed genes provided little amelioration to a GP based standard model without time shift. This leads to the conclusion that the full genome profiles of *Arabidopsis thaliana* prone little to time shifts in data. The Experimentators proceeding the experiment were working exact and precise and the technical problems of microarrays (i.e., noise, temperature, sample preparation and more, see 1.2) were principally extinguished by preprocessing (1.2.1) and normalization (1.2.1). Despite the insignificant improvement, we proved our assumption that allowing time shifts in regression improves the accuracy of detecting differentially gene expression, and even more pleasant, we proved GPTimeShift to not corrupt prediction.

Analyzing TF pairs by applying GPTimeshift, which change their synchronicity due to perturbation admitted us to elucidate key correlations between TF pairs and time shifts. The TF pairs we detected showed a difference in their *WT's* and *mutant's* δT. Comparing the detected pairs to the list of known TF pairs unfortunately indicates that the method has to be adjusted to eventually detect TF pairs. Although the detected pairs do not specify TF pairs, we proved our method, due

to the time shifts δT in the time series of *mutants* annihilate over the experiment. This behavior was expected by a biological expert, who encouraged this result by biological means, namely, deleting a TF causes the perturbated organism to reduce its regulatory network, because the TF does no more control its targets.

In summary the method seems to work and there are many more applications we can think of. We will manifest some of these future applications in Section 4.2.

4.2 Future Implementation

Including time shifts into detecting differentially expressed genes and TF pair correlations in fact show improvements. Despite these improvements, we will outline the future applications and improvements we can additionally think of GPTimeShift.

Separating Individual and Shared Time Shift. To enhance the power of our approach we could change the behavior of the GP models which are compared to each other. The two GP models we compare are $\mathcal{H}_0$ and $\mathcal{H}_{\delta T}$. Both models maintain time shifts for each replicate in individual and shared hypothesis. Here, the enhancement could be to divide the time shift δT into two seperate time shifts $\delta T^{\mathcal{I}}$ and $\delta T^{\mathcal{S}}$. One time shift for the individual model $\delta T^{\mathcal{I}}$ and one for the shared $\delta T^{\mathcal{S}}$. Thus only one time shift would be possible between condition groups it would enhance the computational time for the optimizer to find the optimal set of hyperparameters $\hat{\theta}$. The time shift δT than would be the sum of the individual and shared time shift $\delta T := \delta T_{\mathcal{I}} + \delta T_{\mathcal{S}}$.

Negative correlated gene expression profiles. In some cases the gene expression profiles are negatively correlated. This correlation leads the time shift to be wrong. The GP model with time shift assumes the two expression profiles, which are being compared, to have a time shift or not. If there is no time shift in data and the correlation gets negative this assumptions leads to wrong results. Refer to Figure 4.1, for such a negative correlated gene with a wrong time shift. One possible solution for this problem is to mirror the one gene expression profile at its mean and regress the two profiles, thereafter. Comparing these two separate regressions and choosing the one with the better fit will lead to better results, especially to those, which are negative correlated.

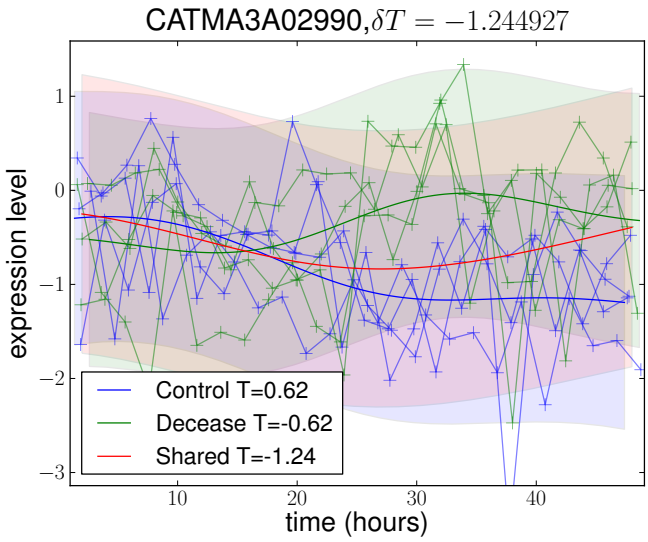


Figure 4.1: Negative correlated genes cause the GP model to learn wrong hyperparameters. This is because the GP model provides a fit for two samples, which cannot be fitted allowing a time shift, but mirroring one of the two samples at its mean.

Gene expression profiles without any variance. Most genes do not vary in their expression profile, considering noise in data. It would be an improvement in computational means to filter out these genes in advance. Additionally the GP model can hardly fit genes, which got no information content, due to scarcely variance. There are several filter methods dealing with such questions and we hope to find an appropriate for including it into Gaussian processes.

Appendix A

Mathematical Appendix

This appendix shows more detailed views on the most significant mathematical steps I got through during the work on the thesis.

A.1 Mathematical Notation

A.1.1 Sets, Matrices, Variables etc.

x	Scalar variable
$\mathbf{x} = (x_1, \ldots, x_n)$	Vector or ordered set of length n
$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$	Set or matrix of length n
$K_{\mathbf{X}} = [k(x, x') : \forall x, x' \in \mathbf{X}]$	Covariance matrix for all pairs of $x, x' \in \mathbf{X}$
$K_{\mathbf{X}^*,\mathbf{X}} = [k(x, x') : \forall x, x' \in \mathbf{X}, \mathbf{X}^*]$	Covariance matrix for all pairs of $x, x' \in \mathbf{X}, \mathbf{X}^*$

A.1.2 Functions, Models etc.

$f(\mathbf{x})$	Function over $\mathbf{x}$
$k(x, x')$	Covariance function over x and x'

A.2 Derivatives of SETP 2.3.1

Since we used an optimizer, that uses the gradient of the CF $\partial/\partial p$ for each $p \in \theta$, this derivatives has to be calculated. Notice, that the derivatives are mathematically computed with $(l_A, l_L, \mathbf{T}, l_\sigma) = \log(A, L, \exp(\mathbf{T}), \sigma)$, for the optimization to be

additive, rather than multiplicative. So with

$$d = \left\| \left(x - \sum_{i=1}^{p} \delta_{i,r} T_r \right) - \left(x' - \sum_{i=1}^{p} \delta_{i,r'} T_{r'} \right) \right\| \tag{A.1}$$

the partial derivatives of $k(x, x')$ are

$$k(x_r, x'_{r'}) = \exp(l_A)^2 \cdot \exp\left(-\frac{d^2}{2\exp(l_L)^2} \right) + l_\sigma^2 \tag{A.2}$$

$$\frac{\partial}{\partial l_A} k(x_r, x'_{r'}) = 2\exp(2l_A) \cdot \exp\left(-\frac{d^2}{2L^2} \right) \tag{A.3}$$

$$\frac{\partial}{\partial l_L} k(x_r, x'_{r'}) = k(x_r, x'_{r'}) \cdot \left(\frac{d^2}{\exp(2l_L)} \right) \tag{A.4}$$

$$\frac{\partial}{\partial T_r} k(x_r, x'_{r'}) = k(x_r, x'_{r'}) \cdot \left(-\frac{d}{L^2} \cdot \left(\sum_{i=1}^{p} \delta_{i,r} - \delta_{i,r'} \right) \right) \tag{A.5}$$

A.3 Kronecker Symbol

The Kronecker symbol is defined as:

$$\delta_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases} \tag{A.6}$$

Appendix B

Software Appendix

B.1 Python

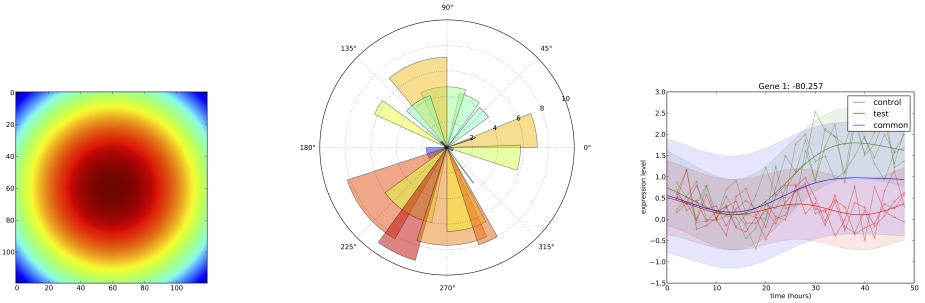
This thesis was implemented using the python script language. For employing the methods one has to make sure the *python2.6* package is installed on the corresponding machine. A quick how-to can be found on the python homepage `http://python.org/`, as well as the documentation, some libraries and more.
To use python appropriate to mathematical and plotting issues, we want to recommend the following libraries for the python language, as they are essential for using the methods introduced in this thesis.

B.1.1 SciPy

"SciPy (pronounced "Sigh Pie") is open-source software for mathematics, science, and engineering. It is also the name of a very popular conference on scientific programming with Python. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The SciPy library is built to work with NumPy arrays, and provides many user-friendly and efficient numerical routines such as routines for numerical integration and optimization. Together, they run on all popular operating systems, are quick to install, and are free of charge. NumPy and SciPy are easy to use, but powerful enough to be depended upon by some of the world's leading scientists and engineers. If you need to manipulate numbers on a computer and display or publish the results, give SciPy a try!"
For more information I would like to recommend their homepage `http://www.scipy.org/`.

B.1.2 Matplotlib

"matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and ipython shell (ala MATLAB® or Mathematica®†), web application servers, and six graphical user interface toolkits.*
matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code. For a sampling, see the screenshots, thumbnail gallery, and examples directory"
Some example visualizations of the plotting library matplotlib are shown in figure B.1. See the whole example library, documentation, downloads and more on their



(a) Colormap, showing a gaussian distribution

(b) Polar map example

(c) Example of drawing a time series

Figure B.1: Some examples drawn from matplotlib library

homepage `http://matplotlib.sourceforge.net/`.

B.2 Implementation

You can download the implementation of GPTimeShift from my website `http://people.kyb.tuebingen.mpg.de/maxz/`. We provide the classes for GP regression (`gpr_nonlog.py` and `secfTime Parameter.py`) and one example application `example.py`. Additionally, we have added a README for proper applicability of the code.

References

[BCS+04] BM Bolstad, F. Collin, KM Simpson, RA Irizarry, and TP Speed. Experimental design and low-level analysis of microarray data. *International Review of Neurobiology*, 60:25–58, 2004.

[BH02] P. Baldi and G.W. Hatfield. *DNA microarrays and gene expression: from experiments to data analysis and modeling*. Cambridge Univ Pr, 2002.

[CC03] X. Cui and G.A. Churchill. Statistical tests for differential expression in cdna microarray experiments. *Genome Biol*, 4(4):210, 2003.

[Gib97] M.N. Gibbs. Bayesian gaussian processes for regression and classification. *Unpublished doctoral dissertation, University of Cambridge*, 1997.

[Hen98] W. Hennig. *Genetik*. Springer Berlin, 1998.

[HGG+10] A. Honkela, C. Girardot, E.H. Gustafson, Y.H. Liu, E.E.M. Furlong, N.D. Lawrence, and M. Rattray. Model-based method for transcription factor target identification with limited data. *Proceedings of the National Academy of Sciences*, 107(17):7793, 2010.

[LLB+01] E.S. Lander, L.M. Linton, B. Birren, C. Nusbaum, M.C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[LRM+06] Alex Lewin, Sylvia Richardson, Clare Marshall, Anne Glazier, and Tim Aitman. Bayesian modeling of differential gene expression. *Biometrics*, 62(1):1–9, 2006.

[MDA04] G.J. McLachlan, K.A. Do, and C. Ambroise. *Analyzing microarray gene expression data*. Wiley-IEEE, 2004.

[Men65] G. Mendel. *Versuche über Pflanzen-Hybriden (1865)*. Arkana-Verl., 1865.

[MWB07] J. Mata, A. Wilbrey, and J. B
"ahler. Transcriptional regulatory network for sexual differentiation in fission yeast. *Genome Biology*, 8(10):R217, 2007.

[Nea96] R.M. Neal. *Bayesian learning for neural networks*. Springer Verlag, 1996.

[NNSA04] Michael Newton, Amine Noueiry, Deepayan Sarkar, and Paul Ahlquist. Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostat*, 5(2):155–176, 2004.

[Qua02] J. Quackenbush. Microarray data normalization and transformation. *nature genetics*, 32:496–501, 2002.

[RW05] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, December 2005.

[SB03] W. Seyffert and R. Balling. *Lehrbuch der Genetik*. Spektrum Akademischer Verlag, 2003.

[SC08] I. Steinwart and A. Christmann. *Support vector machines*. Springer Verlag, 2008.

[SDW+09] O. Stegle, K. Denby, D.L. Wild, Z. Ghahramani, and K.M. Borgwardt. A robust bayesian two-sample test for detecting intervals of differential gene expression in microarray time series. In *Research in Computational Molecular Biology: 13th Annual International Conference, Recomb 2009, Tucson, Arizona, USA, May 18-21, 2009, Proceedings*, page 201. Springer, 2009.

[SG02] P. Sebah and X. Gourdon. Introduction to the gamma function. *numbers. computation. free. fr/Constants/constants. html*, 2002.

[SS02] B. Sch
"olkopf and A.J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. the MIT Press, 2002.

[ZSTV04] A. Zien, B. Schoelkopf, K. Tsuda, and JP Vert. A primer on molecular biology. *Kernel methods in computational biology*, page 3, 2004.