NTNU – Trondheim
Norwegian University of
Science and Technology

Estimating Time-Continuous Gene Expression Profiles Using the Linear Mixed Effects Framework

Christian Magnus Page

Master of Science in Statistics
Submission date: July 2012
Supervisor: Mette Langaas, MATH
Co-supervisor: Torunn Bruland, IKM

Norwegian University of Science and Technology
Department of Mathematical Sciences

PREFACE

This thesis concludes my master's degree (M.Sc.) in statistics at Norwegian University of Science and Technology (NTNU) at Department of Mathematical Science (IMF), under supervision of Mette Langaas.

The research topic was presented by collaborators at Department of Cancer Research and Molecular Medicine (IKM), The Faculty of Medicine, NTNU. The problems were related to experimental design and interpretation of genome-wide time series analysis in mammalian cells.

I would like to thank the collaborators Torunn Bruland DPhil. at IKM and Arnar Flatberg (The Norwegian Microarray Consortium), for making the data set available and giving helpful comments. I would also like to thank friends and family, among them; Ole Fredrik Brevig for and mathematical advice and help with the LaTeX typesetting, and Kim Page and Kari Krizak Halle for proofreading and helpful comments, and everyone else that has somehow helped me during this year.

Especially, I would like to give my warmest thanks to my supervisor Associate Prof. Mette Langaas for motivation and excellent guidance.

Christian Page
Trondheim, June 2012

ABSTRACT

We are drowning in information and starving for knowledge.

Rutherford D. Roger

With the first generation of microarray experiments there were discussions and important arguments on how the samples should be treated. This included what kind of transformation and normalization procedures that the data should be subjected to before the actual analysis. With the new generation of microarray experiments, genome wide association studies, and more complicated experiments, new questions and standards arises. An important question is the appropriate use of controls, and what effect these controls have on the assessed behaviour of the genes.

The main objective in this thesis was to analyse a data set and experimental procedure from an experiment done at IKM (NTNU, 2009), where the effect of a gastrin treatment was measured on a set of genes along with an unstimulated control sample over a time interval. The experiment was replicated once, giving two independent experiments. A natural extension of this is then; what is the gain in precision by adding additional replicas to the experiment? And what additional information is given in an unstimulated control sample?

Our approach was to use the Linear Mixed Effects (LME) framework to fit a regression curve to each gene, for both the treated time series itself, and the treated adjusted for by the unstimulated control sample. Each replication was assumed to have a random offset from the common mean. The mean was modelled using basis expansion with the Legendre polynomials, thus allowing it to vary as a smooth function over the time interval.

A computer simulation showed that an increase in the number of independent time series sampled would decrease the error in the estimated expression profile. Even when this causes the number of time points (measurements) within the time series to decrease.

The analysis of the data showed that not using an unstimulated control gave many false positive results, however, always using such a control will also cause an increase in both false negative and false positive results, due to increase in stochastisity. However, having an unstimulated control sample will give the researcher an increased control when assessing the effect of the treatment.

SAMMENDRAG

Etter endring i studieforskiften §34 (Målform ved skriftlig vurdering) ved NTNU, skal alle oppgaver skrevet på et ikke-skandinavisk språk, som leveres ved fakultetet for Informatikk, Matematikk og Elektronikk (IME fakultetet) ved NTNU, leveres med sammendrag på et skandinavisk språk. Instituttet for Matematiske fag (IMF) ingår under dette fakultet. Bestemmelsen gjøres gjeldende fra og med 01.01.2012 etter vetak fra Dekanus.

Denne oppgaven tar for seg experimentelt oppset ved bruk av mikromatriser for beregning av tidskontinuerlige genprofiler. Vi så spesielt på to spørsmål; (i) hvordan antallet av biologiske replikater påvirker pressisjonen av den estimerte genprofilen, og (ii) hvordan valg av referansepunk for endring i genutrykkningen påvirker signifikansnivået til den estimerte genprofilen.

Analysene er gjort på et dataset fra Institutt for Kreftforskning og Moleylær Medisin (IKM), hvor forskerne målte genutrykket til to celleprøver ved 12 ulike tidspunkt spredt over 14 timer. Den ene celleprøven var behandlet med pattedyrhormonet gastrin, mens den andre celleprøven var en ustimulert kontrol. Begge prøvene ble målt ved samme tidspunkt, slik at alle eksterne faktorer som tempratur, luftrykk og luftfuktighet skulle påvrike celleprøvene like mye. Forsøket ble så replikert en gang, noe som gav to uavhengige forsøk (to biologiske replikater).

Vår tilnærming var å bruke en LME (Linear Mixed Effects; Lineære Kombinerte Effekter) model til å tilpasse hvert gen, med en *tilfeldig effekt* for hver biologiske replikat. I modellen tar vi i bruk basis ekspansjon for å modellere den samlede "gjennomsnittelige" effekten over tid, basis ekspansjonen var basert på Legendre polynomene slik at den esitmerte geneprofilen er et glatt polynom an en gitt orden. De tilfeldige effektene er i oppgaven modelert som stokastisk variabel, som beskriver et konstant skift fra den samlede effekten.

Gjennom en data simulering fant vi at antallet med uavhengige forsøk (biologiske replikater) hadde en possitiv innvirkning på presisjonen til den beregnede geneprofilen. Dette selv om antallet tidspunkter (avhengige forsøk) måtte reduseres, om det totale antalletet forsøk skulle holdes konstant. Vi fant også ut at ved å neglisere den ustimulerte kontrollprøven ville vi få mange falske positive resultater. Men en ustimulert kontrol prøve kan også ødelegge for estimeringen, ved at signalet fra den gastrin stimulerte celleprøven drukner i støy fra den ustimulerte kontrollen. Utkritisk bruk av ustimulert kontroll kan derfor ødellege like mye som å være til fordel, men med en ustimulert kontrollprøve tiljengelig gir dette forskeren en mye større grad av kontroll og fleksibilitet for å avgjøre hvilken effekt gastrin har på genene i den aktuell cellelinjen.

CONTENTS

LIST OF FIGURES

List of Tables

1

INTRODUCTION

Gene expression is the result of activity of the genotype in an organism, and gene expression analysis is the study of this activity. With new advances in biotechnology, measuring DNA and (m)RNA activity is becoming increasingly cheaper and opens the door for more complex experiments. These type of experiments strives to give an insight into how the genes act and their relationship to the surrounding environment.

One new area of research is time continuous expression profiling, on genomic data sets. Here the gene expression is assumed to be a smooth function in time, and one wishes to reconstruct this function for each gene separately, based on observations from microarrays. The estimated underlying curve can then be used to draw conclusions about the genetic behaviour under different conditions.

1.1 Biological Problem and Experimental Design

Understanding gene regulation and cellular decision-making is fundamental to understanding diseases. Genome-wide gene expression time series experiments, are an important approach to studying a cellular response to a given stimulus. This since they enable capturing of the dynamics of the response. The gastric hormone gastrin is a versatile inducer of transcriptional responses and a central regulator of function and growth of the gastric mucosa. Gastrin may also play a role in carcinogenesis of gastrointestinal and stem cell-derived tumours. In order to study molecular mechanisms involved in the gastrin response, the researcher at IKM(NTNU), used the pancreatic derived adenocarcinoma AR42J cell line, which expresses gastrin receptors endogenously and is therefore frequently used as a model system for gastrin responses.

1.1.1 Gene Expression Time Profiling

Adenocarcinoma AR42J cells were grown in 6-well plates (3×105 cells/well in 2 ml DMEM containing 15% FBS) for 72 hours. Then the growth medium was replenished with 2 ml serum-free DMEM, and the cells serum starved for 20-24 hours before adding gastrin (10 nM). Treated and untreated cells were grown in parallel and harvested (pool of

3 technical replicates) at several time points, given in Table 8.1. In order to be able to control for changes in gene expression levels in untreated cells during the 14 hour observation period, the researchers also sampled untreated control cells at all time points. Thus, the Illumina data is a time course reference design, with a non-stimulated reference measured at each sampled response. This design gives information concerning appropriateness of controls in microarray time series experiments: Using on the starting point t_0 as a correction or unstimulated controls at all the different time points. This allows the researcher to infer the level of change the treatment has in the treated cell culture, relative to the normal behaviour of the cell (the unstimulated cell culture).

Most literature on microarray experiment analyses published up till today, concern end points in different cell types, tissues or states. These analyses are well suited to detect the activity status of genes at certain time points under given conditions, but do not provide much information about the regulatory mechanisms in the transitions between different states or processes. Current methodology used for comparison of distinct states, focuses on hypothesis testing methods to detect differentially expressed genes. Application of such methods to time series data with more than a few time points, provides large collections of differentially expressed genes, and further interpretation is often cumbersome. There has also been surprisingly little discussion about these control mechanisms, and almost no published articles about this subject. This analytical shortcoming provide a major motivation for doing this project on microarray time expression profiling.

1.1.2 The Dataset

The data set used in this thesis comes from an experiment at IKM (NTNU), conducted by Torunn Bruland and collaborators in 2009. The data were then preprocessed by Arnar Flatberg (The Norwegian Microarray Consortium).

The data form a long time series[1] (longitudinal data) for each probe (gene) in the microarray, with close to 9000 probes. Measurements were taken over a time frame of 14 hours with two biological replicas for each stimulated treatment and unstimulated control. This gives a total of four time series per gene.

1.2 Thesis Outline

In this thesis we will use the LME (Linear Mixed Effects) framework to answer two important questions: (i) What effect does the numbers of biological replicates have on the precision of the estimated expression profile? (ii) What is the effect of using an external

[1] In microarray context, all time series consisting of more then 5 time points is considered as a 'long' time series.

control sample as compared to using only the starting point as a control adjustment?

In Chapter 2 we introduce the LME framework and give an introduction to the theory and application of LME models. In Chapter 3 we use basis expansion to explain any trend in time and introduce the Legendre polynomials as our chosen basis functions. In Chapter 4 we review the existing literature on gene expression and discuss the methods and approaches used. In Chapter 5 we discuss some criteria for evaluating an estimator and introduce the mean squared error (MSE) criterion. We outline an approximation using Monte Carlo integration when the formulation can not be obtained in closed form, or is to difficult to calculate analytically. In Chapter 6 we present and discuss the suggested model built to analyse the data set. In Chapter 7 we use a Monte Carlo simulation to analyse the effect of adding and removing biological replicas, and to see if there is an optimal way of combining the number of time points and the number of biological replicas, this using the MSE criterion to compare different settings, this chapter strongly links to Chapter 5. In Chapter 8 we investigate the influence of the frame of reference (using t_0 or an unstimulated control), in combination with additional control mechanisms, and model complexity. This using the models presented in Chapter 6. The thesis ends with a discussion and concluding remarks in Chapter 9.

1.2.1 Prerequisites

This thesis discusses problems in biostatistics, and the reader is assumed to have knowledge of some techniques and terms used in modern biotechnology. Knowledge in statistics and specifically regression analysis is assumed throughout out the text. Basic mathematics concepts are assumed known (from matrix algebra and basic analysis).

1.3 Bibliographic Note

For an investigation into the effect of gastrin on mammalian cells, see Watson et al. (2006) and Burkitt et al. (2009). For an introduction to time series experiments in microarray, see Androulakis et al. (2007) and Nueda et al. (2009). However, Storey et al. (2005) is the only article we found that mentions something similar to a frame of reference, but does not go into detail, or discusses what effect this has.

The genome-wide gene expression analysis was performed by Illumina ExpressionBeadChips `http://www.illumina.com/documents/products/datasheets/datasheet_gene_exp_analysis.pdf`. RNA amplifications and hybridization were performed at the NTNU Genomics Core Facilely (GCF).The microarray data were prepared according to minimum information about a microarray experiment (MIAME) recommendations (A. et al., 2001) and deposited in the Array Express (Brazma et al., 2003). Detailed informa-

tion about the microarray design (platform) and raw data files from the experiments are accessible by use of accession number: GSE32869.

2

LINEAR MIXED EFFECTS MODELS

In this chapter we give an introduction to the framework of the Linear Mixed Effects (LME) model. First we give a theoretical presentation of LME models and covariance structures. The estimation of parameters in the model is based on the use of Maximum Likelihood (ML) and Restricted Maximum Likelihood (REML) estimation, which is presented afterwards. We then show how to make inference about the model parameters, using ANOVA and confidence interval. The chapter ends with a discussion on the diagnostics of an estimated model.

The LME framework is an extension of the ordinary linear models. Where all data are independent and identically distributed (iid) in linear models, LME models allow for correlations in the observations, such that, different sources can generate the data, which will be known as clusters, and have internal correlation. As an example, multiple persons can each give many data points. All the data from one person will then be correlated, but the data from one person will be uncorrelated with data from another person. The collection of data from multiple persons will constitute a collection of clusters, that can itself be a new cluster, such that all the persons will together constitute a new cluster. One can think of the observations from one person as the first level, and each person as the second level in a hierarchical model. An LME model can have arbitrary many levels, but we will only consider two level models in this master's thesis.

2.1 Introduction to Linear Mixed Effects Models

An LME model is a composite of a fixed effect μ that is common for all data, and a random effect η which varies between the clusters. The observations are then assumed to be a linear combination of those effects, giving

$$y = \mu + \eta + \varepsilon. \tag{2.1}$$

Here μ is a function common to all data, η is some function that is different for each clusters, and ε is the error, with normal assumptions. The function η is called a random effect, and is modelled as a random variable. The functions μ and η have to be built using a finite linear combination of some basis functions.

The advantage of including the random effect is that, within each cluster the observations can be correlated. So the correlation in y is modelled using η, and the error ε is then, in most cases, assumed to be uncorrelated for all observations. But the clusters have to be independent of each other, e.g. two independent persons, or observations from different geographical locations.

Written in matrix notation and assuming that μ and η can be constructed linear in β and u, the model in Equation (2.1) can be written as

$$\mathbf{y}_k = \mathbf{X}_k \boldsymbol{\beta} + \mathbf{Z}_k \mathbf{u}_k + \boldsymbol{\varepsilon}_k, \tag{2.2}$$

where $k = 1, \dots K$ is used as a grouping factor for the clusters. The fixed effect design matrix $\mathbf{X}_k$ is a $n_k \times (p+1)$ matrix, where n_k is the total number of observations in cluster k, and $p+1$ is the number of covariates used in the model. The fixed effects vector β is then a $(p+1) \times 1$ vector,

$$\mathbf{X}_k \boldsymbol{\beta} = \begin{bmatrix} 1 & x_{11}^k & \dots & x_{p1}^k \\ 1 & x_{12}^k & \dots & x_{p2}^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n_k}^k & \dots & x_{pn_k}^k \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}.$$

The fixed effects vector β is the same for all clusters, but the design matrix may be different for the different clusters. The design matrix for the random effects is $\mathbf{Z}_k$ matrix, that has dimension $n_k \times (q+1)$

$$\mathbf{Z}_k \mathbf{u}_k = \begin{bmatrix} 1 & z_{11}^k & \dots & z_{q1}^k \\ 1 & z_{12}^k & \dots & z_{q2}^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & z_{1n_k}^k & \dots & z_{qn_k}^k \end{bmatrix} \begin{bmatrix} u_0^k \\ u_1^k \\ \vdots \\ u_q^k \end{bmatrix}.$$

Here $\mathbf{u}_k$ is the vector of random effects for cluster k.

The vector of the observations is denoted by $\mathbf{y}_k$, and is a column vector of length n_k, and the error vector $\boldsymbol{\varepsilon}_k$ is a vectors of length n_k. We assume that the random effects and the errors have the following distributions,

$$\mathbf{u}_k \overset{\mathrm{d}}{=} \mathcal{N}_{q+1}(\mathbf{0}, \mathbf{D}) \qquad k = 1, \dots, K, \tag{2.3a}$$

$$\boldsymbol{\varepsilon}_k \overset{\mathrm{d}}{=} \mathcal{N}_{n_k}(\mathbf{0}, \mathbf{R}_k) \qquad k = 1, \dots, K. \tag{2.3b}$$

Here $\mathbf{D}$ is the variance-covariance matrix for the random effects, and $\mathbf{R}_k$ is the variance-covariance matrix in the error term, for a cluster k. If all dependency between the observations have been explain by the random effect, the $\mathbf{R}_k$ matrix becomes a diagonal $(\sigma_R^2 \mathbf{I}_n)$ variance matrix. The structure of the covariance matrices will be further discussed in Section 2.2.

2.1.1 The Implied Marginal Model

The error term can be redefined by adding the error and the random effect together, this gives the equation

$$\boldsymbol{\varepsilon}_k^\star = \mathbf{Z}_k\,\mathbf{u}_k + \boldsymbol{\varepsilon}_k. \tag{2.4}$$

Since linear combinations of independent multivariate normally distributed variables are multivariate normally distributed, the new defined error $(\varepsilon^\star)$, will also be multivariate normally distributed. The expected value of $\boldsymbol{\varepsilon}^\star$ is then

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\varepsilon}_k^\star] &= \mathbb{E}[\mathbf{Z}_k\,\mathbf{u}_k] + \mathbb{E}[\boldsymbol{\varepsilon}_k] \\
&= \mathbf{Z}_k\,\mathbb{E}[\mathbf{u}_k] + \mathbb{E}[\boldsymbol{\varepsilon}_k] \\
&= \mathbf{Z}_k\,\mathbf{0} + \mathbf{0} && \text{((2.3a) and (2.3b))} \\
&= \mathbf{0},
\end{aligned}$$

and the covariance matrix of $\boldsymbol{\varepsilon}_k$ is

$$\begin{aligned}
\mathrm{Cov}(\boldsymbol{\varepsilon}_k^\star) &= \mathrm{Cov}(\mathbf{Z}_k\,\mathbf{u}_k) + \mathrm{Cov}(\boldsymbol{\varepsilon}_k) \\
&= \mathbf{Z}_k \mathrm{Cov}(\mathbf{u}_k)\mathbf{Z}_k^T + \mathrm{Cov}(\boldsymbol{\varepsilon}_k) \\
&= \mathbf{Z}_k\,\mathbf{D}\,\mathbf{Z}_k^T + \mathbf{R}_k \\
&= \mathbf{V}_k.
\end{aligned} \tag{2.5}$$

Then the redefined error term follows a multivariate normal distribution, with mean $\mathbf{0}$ and variance-covariance matrix given in Equation (2.5), such that

$$\boldsymbol{\varepsilon}_k^\star \overset{\mathrm{d}}{=} \mathcal{N}_n(\mathbf{0}, \mathbf{V}_k). \tag{2.6}$$

Then the regression model can now be written as

$$\mathbf{y}_k = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}_k^\star. \tag{2.7}$$

This is called the implied marginal model, and has distribution

$$\mathbf{y}_k \overset{\mathrm{d}}{=} \mathcal{N}_{n_k}(\mathbf{X}_k\,\boldsymbol{\beta}, \mathbf{V}_k). \tag{2.8}$$

These calculations substantially reduces the estimation problem, as the only two quantities to be estimated are the fixed effects and the covariance matrix $\mathbf{V}_k$, especially if $\mathbf{V}_k$ has few parameters, and known structure.

2.1.2 The Full Model

To construct the full model, the design matrices for the fixed effect clusters are stacked upon each other, while the design matrix for the random effects are placed on the diagonal

of a new matrix, giving $\mathbf{X}$ and $\mathbf{Z}$ as

$$\mathbf{X} = [\mathbf{X}_1 \ \mathbf{X}_2 \ \ldots \ \mathbf{X}_K]^T \qquad \mathbf{Z} = \text{block}(\mathbf{Z}_k). \tag{2.9}$$

The fixed effect parameter β is unchanged, but the different random effects $\mathbf{u}_k$'s are not the same and are stacked in a column vectors of length $N = \sum n_k$, as are all the $\mathbf{y}_k$'s. The full model can then be written as

$$\mathbf{Y} = \mathbf{X}\,\boldsymbol{\beta} + \mathbf{Z}\,\mathbf{u} + \boldsymbol{\varepsilon}. \tag{2.10}$$

Here $\mathbf{Y}$ and $\boldsymbol{\varepsilon}$ are now $N \times 1$ vectors, $\mathbf{X}$ is a $N \times (p+1)$ matrix, and $\mathbf{Z}$ is a $N \times (q+1)K$ block diagonal matrix, the random effects vector is augmented along with $\mathbf{Z}$, giving it dimension $K(q+1) \times 1$. Since the number of fixed effects are common to all clusters by construction, the column length has to be the same for all $\mathbf{X}_k$. The same arguments goes for the $\mathbf{Z}_k$.

2.2 Covariance Structures

The complexity of LME models is determined by the number of parameters (both fixed and random), and the covariance structure in the model. Here we show how to extend the LME model with correlation structure, starting with the necessary definitions.

Definition 2.1. *The covariance between two random variables X_1 and X_2 is defined as*

$$\text{Cov}(X_1, X_2) = \mathbb{E}\left[(X_1 - \mathbb{E}[X_1])(X_2 - \mathbb{E}[X_2])\right].$$

Provided that $\mathbb{E}[|X_1 X_2|] < \infty$, $\mathbb{E}\left[|X_1|^2\right] < \infty$ *and* $\mathbb{E}\left[|X_2|^2\right] < \infty$

A normalized version of the covariance is the correlation, which is bounded by the interval $[-1, 1]$.

Definition 2.2. *The correlation between two random variables X_1 and X_2 is defined as*

$$\text{Corr}(X_1, X_2) = \frac{\text{Cov}(X_1, X_2)}{\sqrt{\text{Var}(X_1)\text{Var}(X_2)}}.$$

Provided that the covariance exists, and that $\text{Var}(X_1) > 0$ *and* $\text{Var}(X_2) > 0$.

For a vector of random variables, all the different covariances can be presented in a variance-covariance matrix. The variance-covariance matrix is then a matrix with the variances on the diagonal and the covariances on the off-diagonals. Since the covariance is symmetric, this matrix will also be symmetric. It can also be shown, that a variance-covariance matrix is always positive semi-definite.

The variance-covariance matrix for the random effects is denoted $\mathbf{D}$, and for the error it is denoted $\mathbf{R}_k$,

$$\mathbf{D} = \mathrm{Cov}(\mathbf{u}_k) \qquad \mathbf{R}_k = \mathrm{Cov}(\boldsymbol{\varepsilon}_k) \qquad \mathbf{R} = \mathrm{block}(\mathbf{R}_k). \tag{2.11}$$

The variance-covariance matrix of the implied marginal model is given from Equation (2.5), is called $\mathbf{V}_k$, and given as

$$\mathbf{V}_k = \mathbf{Z}_k \, \mathbf{D} \, \mathbf{Z}_k^T + \mathbf{R}_k. \tag{2.12}$$

If the errors $\boldsymbol{\varepsilon}_k$ are uncorrelated, then the covariance matrix $\mathbf{R}$ is diagonal ($\sigma_R^2 \mathbf{I}_N$). If however, the errors are correlated within each cluster, then $\mathbf{R}$ becomes a block diagonal matrix. The covariance matrix ($\mathbf{V}_k$) is generally dense by construction, since $\mathbf{Z}_k \, \mathbf{D} \, \mathbf{Z}_k^T$ is dense, but usually with few parameters.

2.2.1 Within Cluster Correlation

The intraclass correlation (ICC) gives a measure of the correlation between two observations from the same cluster. It can be interpreted as the amount of the variation that can be attributed to the variation between the clusters, compared to the total variation.

Definition 2.3. *The intraclass correlation for the LME model with only one random effect (intercept) is defined as the variance between the clusters, divided on the total variance*

$$\rho = \frac{\sigma_D^2}{\sigma_D^2 + \sigma_R^2} \geqslant 0$$

where $\sigma_D^2 = \mathbf{D} = \mathrm{Var}(\mathbf{u})$ *and* $\sigma_R^2 = \mathrm{Var}(\boldsymbol{\varepsilon})$.

From this definition it is clear that the correlation is always positive, and thus not representing the whole spectrum of possible correlations.

With multiple random effects, this definition has to be extended and then gives a correlation matrix. If $\mathbf{R}_k$ is the same for all clusters, then the correlation matrix is the same for all clusters, since $\mathbf{D}$ is only dependent on the different random effects $u_0^k, \cdots, u_q^k$, and not on the different clusters. The subscript k in the correlation matrix is therefore induced by $\mathbf{R}_k$ and not through the random effects.

Definition 2.4. *We define the intraclass correlation for a cluster* k*, by*

$$\boldsymbol{\rho}_k = \left(\Lambda_k^{-1/2}\right) \mathbf{V}_k \left(\Lambda_k^{-1/2}\right)^T,$$

where Λ_k *is defined as a* $n_k \times n_k$ *diagonal matrix, with the diagonal elements from* $\mathbf{V}_k$*, so that* $\Lambda_k^{1/2} = \sqrt{\mathrm{diag}(\mathbf{V}_k)}$*, by spectral value decomposition.*

The ρ_k is the correlation between the observations within a cluster k. The ρ_k matrix has dimension $(n_k \times n_k)$, if there are no missing observations.

2.2.2 Construction of Variance-Covariance Matrices

Two variance-covariance matrices needs to be specified, for the random effects (from Equation (2.3a)), and the covariance matrix of the errors (Equation (2.3b)). The covariance matrix $\mathbf{D}$, is how the random effects are correlated. The matrix $\mathbf{R}$ gives the covariance structure in the errors, and indicates if the errors are correlated within a cluster or if they are independent.

The variance-covariance error matrix $\mathbf{R}_k$ is often assumed to be diagonal, on the form $\sigma_R^2 \mathbf{I}_n$. In a time series setting, such as longitudinal data analysis, the observations within the clusters can be assumed correlated with each other, often modelled as an AR(p) structure. The precision matrix ($\mathbf{R}_k^{-1}$) then becomes spares, as a $2p+1$ bond diagonal matrix. Assuming that the errors are dependent on one time step back, as an AR(1), will give a sparse (tridiagonal) precision matrix. However, the inverse of a sparse matrix is not itself sparse, and may become dense and difficult to work with computationally. This itself is usually not a problem, since $\mathbf{V}_k$ is dense by construction. Even with only one random effect, the matrix $\mathbf{Z}_k \, \sigma_D^2 \, \mathbf{Z}_k^T$ is dense, with $\sigma_D^2 + \sigma_R^2$ on the diagonals, and σ_D^2 on the off-diagonals.

The complete covariance matrix $\mathbf{V}$ is constructed by diagonally stacking $\mathbf{V}_k$, which then will give a new positive definite covariance matrix, corresponding to Equation (2.10).

2.3 Parameter Estimation

In all aspects of statistics the data reduction plays an important part, and in this thesis it revolves around the estimation of the parameters in Equation (2.1). There are a wide range of methods available to estimate parameters, the most notable among them is the maximum likelihood (ML) method.

2.3.1 Maximum Likelihood Estimate

Using the implied marginal model from Equation (2.8), assuming that all the observations follow a multivariate normal distribution, the joint probability distribution then becomes

$$
\begin{aligned}
f(\mathbf{Y}|\boldsymbol{\beta},\, \mathbf{V}) &= \prod_{k=1}^{K} f(\mathbf{y}_k|\boldsymbol{\beta},\, \mathbf{V}_k) \\
&= \prod_{\forall k} \frac{1}{(2\pi)^{\frac{n_k}{2}} \det(\mathbf{V}_k)^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{y}_k - \mathbf{X}_k\boldsymbol{\beta})^T \mathbf{V}_k^{-1} (\mathbf{y}_k - \mathbf{X}_k\boldsymbol{\beta}) \right\},
\end{aligned}
$$

hence, given the data, the log-likelihood becomes

$$l_{\mathrm{ML}}(\boldsymbol{\beta},\, \mathbf{V}) = \frac{-n\log(2\pi)}{2} - \frac{1}{2}\sum_{\forall k}\log(\det(\mathbf{V}_k)) \\ - \frac{1}{2}\sum_{\forall k}(\mathbf{y}_k - \mathbf{X}_k\boldsymbol{\beta})^T\mathbf{V}_k^{-1}(\mathbf{y}_k - \mathbf{X}_k\boldsymbol{\beta}). \tag{2.13}$$

If the variance-covariance matrix $\mathbf{V}_k$ is assumed known, then the log-likelihood becomes a function of $\boldsymbol{\beta}$ only. This can be maximized on closed form, giving

$$\widehat{\boldsymbol{\beta}}^{\mathrm{ML}} = \left(\sum_{\forall k}\mathbf{X}_k^T\mathbf{V}_k^{-1}\mathbf{X}_k\right)^{-1}\sum_{\forall k}\mathbf{X}_k^T\mathbf{V}_k^{-1}\mathbf{y}_k. \tag{2.14}$$

The log-likelihood for the covariance matrix $\mathbf{V}_k$ cannot be maximized by a closed form expression, and a numerical approximation algorithm has to be used. To maximize the likelihood with respect to $\mathbf{V}_k$, the expectation maximization (EM) procedure can be used. The EM algorithm is an iterative process that seeks to maximize the expected value of the log-likelihood for an estimated parameter.

2.3.2 Estimation of Restricted Maximum Likelihood

Restricted Maximum Likelihood (REML) estimation is an alternative to ML estimation, in that it have the advantage of giving unbiased estimates of the covariance parameters, and thus is often preferred when estimating the covariance.

$$l_{\mathrm{REML}}(\mathbf{V}_k) = -\frac{1}{2}(n-(p+1))\log(2\pi) - \frac{1}{2}\sum\log(\det(\mathbf{V}_k)) \\ -\frac{1}{2}\sum(\mathbf{r}_k^T\mathbf{V}_k^{-1}\mathbf{r}_k) - \frac{1}{2}\sum\log(\det(\mathbf{X}_k^T\mathbf{V}_k^{-1})), \tag{2.15}$$

where p is the number of parameter estimated in the fixed effect, and $\mathbf{r}_k$ is the observed error, defined in Equation (2.16) as

$$\mathbf{r}_k = \mathbf{y}_k - \mathbf{X}_k\,\widehat{\boldsymbol{\beta}} = \mathbf{y}_k - \mathbf{X}_k\left(\sum_{\forall k}\mathbf{X}_k^T\mathbf{V}_k^{-1}\mathbf{X}_k\right)^{-1}\sum_{\forall k}\mathbf{X}_k^T\mathbf{V}_k^{-1}\mathbf{y}_k. \tag{2.16}$$

The REML estimation takes into account the loss of degrees of freedom from estimating $\boldsymbol{\beta}$, and thus gives an unbiased estimator. When $\mathbf{V}_k$ is estimated with $\widehat{\mathbf{V}}_k^{\mathrm{REML}}$, the β's can be estimated using Equation (2.14) to obtain the REML estimate of $\boldsymbol{\beta}$. The estimates

will differ since the variance-covariance matrix is estimated differently, and affects the estimated $\widehat{\boldsymbol{\beta}}$'s.

Equations 2.13 and 2.15 have strong similarities, but where the ML estimator uses n, the REML uses $(n - (p + 1))$, and it subtracts a "penalty", seen as the last sum in Equation (2.15). As long as the determinant of the design matrix $\mathbf{X}_k$ is not close to zero, the penalty does not diverge. To maximize the REML equation, the same approach as for the ML can be used by iterating with the EM algorithm. It has been proven that the REML estimator is consistent, under fairly weak conditions.

2.3.3 Best Linear Estimator

A best estimator is often considered to be the estimator that has the least variance possible variance, of all considered estimators. For an unbiased estimator, the minimal variance is the Cramér-Rao lower bound, and an estimator which obtain this bound is called the most efficient estimator. The unbiased linear estimator with least variance, is called a Best Linear Unbiased Estimator (BLUE).

Definition 2.5. *A linear unbiased estimator $\hat{\theta}$ is **BLUE**, if and only if, for any other unbiased linear estimator $\tilde{\theta}$,*

$$\mathrm{Var}(\hat{\theta}) \leqslant \mathrm{Var}(\tilde{\theta}).$$

A well known theorem, called the Gauss-Markov theorem, says that in ordinary linear models, this estimator can be obtained.

Theorem 2.6. ***Gauss-Markov Theorem**; Under normal conditions, the linear estimator $\hat{\theta}$ obtained by least squares, has the least variance among all possible linear unbiased estimators.*

This implies that the least squares estimate is a BLUE. Under normal conditions without random effects, the MLE is the same as the LS estimate, which then implies that the ML estimate is also a BLUE. With random effects included, it becomes more difficult, however, it has been proven that under fairly general conditions, the Gauss-Markov theorem can be extended to include the estimation of random effects.

The restriction to only unbiased estimators, may not be the wisest choice. By allowing for some bias, one might reduce the variance considerably. This is discussed in detail in Chapter 5, where bias and variance is combined to form an assessment criterion of an estimator.

2.4 Statistical Inference

Our aim is now to draw inference about the model parameters. Typically, one wishes to test if at least some parameters differ from zero (or any other value). To test if a linear combination of the parameters are different from zero a contrast matrix can be used.

2.4.1 Fixed Effects Parameters

To construct a hypothesis test for the fixed effect parameters, two approaches can be used. One is the standard t-test for one parameter, and the other test is the F-test for contrasts of the parameters. Both tests depend on the estimated variance-covariance matrix, and are thus often referred to as conditional tests.

The hypothesis for only one parameter, can be stated as follows

$$\begin{aligned} &\mathrm{H}_0 : \beta_j = 0 \\ &\mathrm{H}_1 : \beta_j \neq 0, \end{aligned} \tag{2.17}$$

for some j. We base the inference on the t-statistic, calculated by

$$t_{df} = \frac{\widehat{\beta}_j}{\widehat{\mathrm{se}}(\widehat{\beta}_j)},$$

which approximately follows the t-distribution, with the degrees of freedom depending on the cluster that β is associated with.

To test if a set of linear combinations of the parameters are zero,

$$\begin{aligned} &\mathrm{H}_0 : \mathbf{C}\boldsymbol{\beta} = \mathbf{0} \\ &\mathrm{H}_1 : \mathbf{C}\boldsymbol{\beta} \neq \mathbf{0}, \end{aligned} \tag{2.18}$$

for some contrast matrix $\mathbf{C}$. The contrast matrix $\mathbf{C}$ must have column dimension equal to the dimension of β, which is $p + 1$. The number of rows in the contrast matrix $\mathbf{C}$ are the numbers of linear combinations of the parameter that is tested. The test statistic for this hypothesis is constructed as in Equation (2.19). Under H_0, the F-statistic

$$F = \frac{\widehat{\boldsymbol{\beta}}^T \mathbf{C}^T \left(\mathbf{C} \left(\sum_{k=1}^{K} \mathbf{X}_k^T \widehat{\mathbf{V}}_k \mathbf{X}_k \right) \mathbf{C}^T \right) \mathbf{C} \widehat{\boldsymbol{\beta}}}{\mathrm{rank}(\mathbf{C})} \tag{2.19}$$

is approximately F_{df_1, df_2}-distributed, with degrees of freedom equal to the rank of $\mathbf{C}$ in the numerator, and depending on the grouping level in the denominator.

If the contrast matrix $\mathbf{C}$ is a diagonal matrix ($\mathbf{I}_{p+1}$), then this is the same as testing the hypothesis

$$\begin{aligned} &\mathrm{H}_0 : \beta_0 = \beta_1 = \cdots = \beta_p = 0 \\ &\mathrm{H}_1 : \text{At least one } \beta \text{ differs from zero.} \end{aligned} \tag{2.20}$$

Setting the first column to zero, one can then test if there is a change from a constant level.

2.4.2 Confidence Interval

To construct a confidence interval around the curve f, in a point x_0, using the estimated curve $\hat{f}$ and assuming that the $\hat{\beta}$'s are approximately t-distributed, we get

$$\mathrm{CI} : \hat{f}(x_0) \pm t_{df,\alpha/2} \sqrt{\mathrm{diag}\left(\mathrm{Cov}\left(\hat{f}(x_0)\right)\right)},$$

this since the estimated function $\hat{f}$ is assumed to be a linear combination in $\widehat{\boldsymbol{\beta}}$, as $\hat{f} = \mathbf{X}\,\hat{\beta}$. Then the confidence interval around $f(x_0) = \mathbf{x}_0^T\,\boldsymbol{\beta}$ can be calculated as

$$\mathrm{CI} : \mathbf{x}_0^T\,\widehat{\boldsymbol{\beta}} \pm t_{df,\alpha/2} \sqrt{\mathrm{diag}\left(\mathbf{x}_0\,\widehat{\mathrm{Cov}}\left(\widehat{\boldsymbol{\beta}}\right)\,\mathbf{x}_0^T\right)}.$$

This will give confidence interval on for f, in each point $\mathbf{x}_0$. If the confidence interval that does not contain zero for all points, will suggest that the function f is not zero in a least one point. The fundamental theorem of algebra dictates that a polynomial of degree $p+1$ with non-zero coefficients, can have at most $p+1$ roots. This means that testing if the function f is zero in a set of points larger then $p+1$, is the same as testing if all the coefficients are equal zero.

In order to describe the precision in the estimate while not using a confidence interval, it is possible to use the Mean Squared Error (MSE) point wise. The MSE is a composite of the variance and bias, and will be further developed in Chapter 5.

2.5 Diagnostics for MLE Models

Once a model is constructed, it is then important to check if the underlying assumptions hold. For a linear regression problem there exists an array of diagnostic techniques, however, due to the increased complexity in the LME model, the number of checks become more restricted.

A number of key assumptions have been made in the modelling process; (i) independence in the error, (ii) approximately normally distributed error, and (iii) no heteroscedasticity in the error. All these issues have to be assessed and if the model fail to meet these

criteria, an update may be required. Removing influential observations, transformation of input, or basis expansion and regularization are examples of such updates. The model diagnostics are based on multiple quantities, the most important of which are residual checks. Additional validation assessment is to look for influential observations, and to analyse the behaviour of the random effects. However, using the *raw residuals* has been noted to brake some of the assumptions, and should therefore be handled accordingly. They most common way to handle this, is to use some form of standardization.

Using the raw residuals r^{raw} to diagnose the model could often have undesired consequences. The raw residuals are defined as

$$r^{\text{raw}} = y_i - \hat{y}_i,$$

which is the difference between the estimated curve, and the actual observation.

The residuals will often be correlated, and have unequal variance. To reduce this problems, some standardisation method is used, the most common way is to is to divide the raw residuals by the estimated standard deviation, giving *standardised residuals*[1].

Residual Diagnostics

How does the discrepancy between the predicted value and the observed value behave? This will tell something about the fit of the model. Due to the increased complexity in LME, this is mostly done by visual inspection, using plots. The first step is generally to analyse the errors with a QQ-plot, which will tell if they are normally distributed or have tail heaviness. If the residuals fall onto a straight line they are approximately normally distributed. The easiest way to spot patterns in the estimated error is to plot them, either against their indexes or against the fitted values. The purpose is to detect any possible dependences between the fitted values and the error i.e. as the values in $\hat{y}$ increased, the error should not. If the errors are plotted against their indexes, any pattern related to heteroscedasticity should then be revealed.

The random effects are also assumed to be normal and should also be plotted. A QQ-plot will tell if the assumptions hold, or if the random effects do exhibit irregularities.

2.5.1 Influence Diagnostics

Likelihood methods (LM and REML) as well as least squares are sensitive to outliers and extreme values. It is therefore important to identify extreme value observations that

[1] This is often also called *studentized residuals* since they approximately follows the student t-distribution.

may influence the estimated model negatively. There are many ways of identifying influential observations, two common ways are to use Cook's distance and the leverage. Tail-heaviness in the residual distribution is also an indication of multiple outliers.

It has been suggested that a top-down strategy, may help identify outliers. That is, to first assess the fit of the whole parameter set (denoted $\boldsymbol{\theta}^{\text{full}}$). Next, reduce the data or the parameter vector to identify if there is a subset of the observations that has much influence on the modelled parameters. A way of comparing fit with a reduced parameter vector is the deviance (sometimes refereed to as likelihood distance) given by

$$D = -2\left(l\left(\boldsymbol{\theta}^{\text{full}}\right) - l\left(\boldsymbol{\theta}^{\text{less}}\right)\right).$$

Here D is approximately χ^2-distributed, with degrees of freedom equal to the total number of dimension of the full parameter vector, minus the dimension of the reduced parameter vector. The trace of the precision matrix (inverse covariance matrix) has also been suggested as a way to identify the impact of a parameter, but this has no known associated distribution, as the deviance has.

Since the model is fitted minimizing the errors, assessment of fit might be to optimistic using the residuals. A common way around this, is to use cross-validation. This will also identify any influential observations, or a set of influential observations. It has however been reported that cross-validation may not give a good assessment of fit, especially leave-one-out cross-validation.

2.6 Bibliographic Note

Most of the concepts described here draws from the the book by Pinheiro and Bates (2000), as well as a master's thesis by Østgård (2011). The proof of consistency of REML can be found in Jiang (1997), and a proof that the Gauss-Markov theorem can be extended to random effect, is found in Harville (1976). The diagnostic section is mainly inspired by West et al. (2006), but for a more thoroughly review of diagnostics in LME see Schabenberger (2004).

The origin of REML has been attributed to Thompson (1962). Using the software R (R Development Core Team, 2011), there is an implementation of LME, developed by Pinheiro et al. (2011), that handles both linear and non-linear LME. Although many other software packages have been made available, we choose to use only this in our thesis.

3

TIME COURSE WITH LME

Often data trends are moving in non-linear smooth ways and a simple linear approximation does not fit the data well. One can then construct more complex functions to describe the change in the mean trend in time.

A naive approach is to use some transformation of the observed values, i.e. using logarithm or trigonometric functions. A more sophisticated method is to construct the *mean value function* as a finite linear combination of some basis functions, which is known as basis expansion. It will be demonstrated that if the expansion is done in a reasonable way, the estimation of the parameters can be done using the LME framework presented in Chapter 2.

If there is some moving time dependent trend in the data and the model accounts for this, then we will assume that the errors are independent and approximate normal distributed with mean zero.

3.1 Basis Expansion

To extend the model in a non-linear fashion in a covariate (illustrated as a variable t), but still keep the linearity in the coefficients, we use a finite linear combination of some functions h and g. We express the changing trend in the covariate, as

$$y(t) = \sum \beta_i h_i(t) + \sum u_j g_j(t), \tag{3.1}$$

where the $h(t)$ and $g(t)$ can be arbitrary functions. Observe that the basis expansion can be used for both the fixed and the random effects in an LME model, independent of each other. In this thesis, we will only use expansion for the fixed effect, and have not considered expansion in the random effect.

Setting $h_n(t)$ to $a_n \sin(nt)$ and $b_n \cos(nt)$ will give the Fourier basis, while $h_n(t) = t^n$ is the standard polynomial basis. The cubic spline basis is obtained by dividing the interval into a number of pieces (knots), then fitting a cubic polynomial to each interval and joining the curve at all the knots. If the curve is twice differentiable in all the knots and if it has linear functions at the end pieces, it is called a Natural Cubic Spline (NCS).

With matrix notation, the expansion Equation (3.1) can be written as in Equation (2.2), but now with the basis functions constituting the columns in the design matrix.

$$\mathbf{y}_k(t) = \mathbf{H}(t)\,\boldsymbol{\beta} + \mathbf{G}(t)\,\mathbf{u}_k + \boldsymbol{\varepsilon}_k. \tag{3.2}$$

Here the matrix $\mathbf{H}(t)$ correspond to the design matrix in Equation (2.2), but with a basis expansion in t, and $\mathbf{G}(t)$ as the expansion for the random effect. When the number of basis functions increases, they will tend to interpolate the observed data points. This may result in increasing the variance in the model, and making it a poor predictor for the underlying phenomenon.

In basis expansion, one often wants the basis functions to be orthogonal and there are many different functions that exhibit this property, among those the Fourier basis, Hermite polynomial, and the Legendre polynomials. In this thesis, the Legendre polynomials are used as they are orthogonal with respect to the $\mathcal{L}^2[-1,1]$ inner product.

3.2 The Legendre Polynomials

The Legendre polynomials are a sequence of polynomials, that are orthogonal with respect to the $\mathcal{L}^2$ inner product.

Definition 3.1. *The Hilbert space $\mathcal{L}^2[-1,1]$ is the set of all functions f mapping $[-1,1]$ to a subset in $\mathbb{R}$, that are integrable with*

$$\int_{-1}^{1} f(t)^2 \,\mathrm{d}\,t < \infty,$$

and with inner product

$$\langle f, g \rangle \overset{\text{def}}{=} \int_{-1}^{1} f(t)g(t) \,\mathrm{d}\,t.$$

The inner product $\langle f, f \rangle^{1/2}$, is called the $\mathcal{L}^2$ norm.

Definition 3.2. *Two functions h and g are said to be **orthogonal** with respect to the $\mathcal{L}^2$ inner product if*

$$\langle h, g \rangle = 0 \qquad h \neq g.$$

Definition 3.3. *If a sequence of orthogonal functions $f_1, f_2, \ldots$, has the following property*

$$||f_i||_2 = \langle f_i, f_i \rangle^{1/2} = 1 \qquad \forall\, i,$$

*then the sequence is said to be **orthonormal**.*

Observe that a sequence of orthogonal functions, can easily be made orthonormal by dividing by a normalizing constant, since from Definition 3.1, the inner product and norm have be finite.

The following statement is given without proof.

Theorem 3.4. *Any polynomial defined recursively as*

$$\mathcal{P}_n(t) = (t - a_n)\mathcal{P}_{n-1}(t) - b_n\mathcal{P}_{n-2}(t),$$

with $\mathcal{P}_0(t) = 1$, $\mathcal{P}_1(t) = t - a_1$, *and*

$$a_n = \langle t\mathcal{P}_{n-1}(t), \mathcal{P}_{n-1}(t)\rangle / \langle \mathcal{P}_{n-1}(t), \mathcal{P}_{n-1}(t)\rangle \tag{3.3}$$
$$b_n = \langle t\mathcal{P}_{n-1}(t), \mathcal{P}_{n-2}(t)\rangle / \langle \mathcal{P}_{n-2}(t), \mathcal{P}_{n-2}(t)\rangle \tag{3.4}$$

is orthogonal with respect to the $\mathcal{L}^2$ inner product.

The common way to define the Legendre polynomials are through the Legendre equation, which gives rise to the following differential equation in Definition 3.5.

Definition 3.5. *The **Legendre Polynomials** l_n are the sequence of polynomials, that solve the differential equation*

$$l_n(t) = \frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}\, t^n}(t^2 - 1)^n.$$

Since this equation uses the n-order derivative, it is not easy for a computer to evaluate this formula fast. Another approach is to calculate the sequence recursively by the form

$$(n + 1)\, l_{n+1}(t) = (2n + 1)\, t\, l_n(t) - n\, l_{n-1}(t), \tag{3.5}$$

and using that $l_0(t) = 1$, $l_1(t) = t$. It can be shown that the Definition 3.5 holds for this sequence. It can also be shown that the Legendre polynomials are the only sequence of polynomials, that are orthogonal on the $\mathcal{L}^2[-1, 1]$ norm.

It is then easy to verify that the Legendre polynomials are orthogonal either by direct calculation or by using Theorem 3.4. From the definition of the Legendre polynomial, one can see that the inner product (norm) is

$$\langle l_i, l_j\rangle = \int_{-1}^{1} l_i(t) l_j(t)\, \mathrm{d}\, t = \frac{2}{2i + 1}\delta_{ij}, \tag{3.6}$$

where δ_{ij} is the Kronecker delta.

A major issue when working with higher order polynomials, is their erratic behaviour at the boundary. NCS solves this problem by restricting the fitted curve to a linear function at the end pieces, but for a general polynomial it is harder. One way of resolving this problem, is to restrict the estimation problem on some interval $[a, b]$. If the interval chosen is around $[-1, 1]$, then the behaviour of the polynomial is not to erratic. However this is only a good fix for curve fitting inside this interval, for predictions outside this interval, any polynomial will still be highly volatile.

3.2.1 Construction of the Legendre Basis

Since polynomials are continuous by construction, they must be discretized to be applied into the design matrix. For each observed value y_i, there is an associated time point t_i, and the relation $y_i|t_i = f(t_i)$ is a Legendre polynomial of degree $p + 1$, in one point t_i. Then a set of points $\{f(t_i)\}$, which is interpolated by a linear combination of $p + 1$ Legendre polynomials, is the basis for the design matrix.

The basis $\mathbf{H}$ in Equation (3.7) is then the j'th order Legendre polynomial evaluated at the time points $\{t_i\}$, placed on row j. Then the element h_{ij} the basis matrix $\mathbf{H}$ is then $l_j(t_i)$. The coefficients will be fitted, using only the time points $\{t_i\}$, but the coefficients will still uniquely define a continuous polynomial, that can be used to predict $y(t)$, for a continuous t, in the interval $[-1, 1]$.

This gives the equation for the observed values in a time point t_i, as

$$y(t_i) = \beta_0\, l_0(t_i) + \beta_1\, l_1(t_i) + \beta_2\, l_2(t_i) + \beta_3\, l_3(t_i) + \beta_4\, l_4(t_i). \tag{3.7}$$

The rows in the design matrix $\mathbf{H}$, are replaced with the $l_j(t)$ for all the observed time points t_i.

In Figure 3.1, the polynomials represented in Equation (3.7) are plotted, on the interval $[-1, 1]$. The red dots represents the points at which the expansion is used (normalized from Table 8.1). Since the design matrix is discrete in its construction, the basis cannot be represented continuously. The basis is then the value of the function, in a given time point. In Figure 3.1 this is shown only for the second order Legendre polynomial, for easy interpretation. Any other order might as well be highlighted.

3.3 Best Estimator

When using a polynomial basis expansion, different kinds of polynomials will give the same result in terms of information utilization. The reason to use orthogonal polynomials is that all the coefficients will then be independent, that is, zero correlation between the

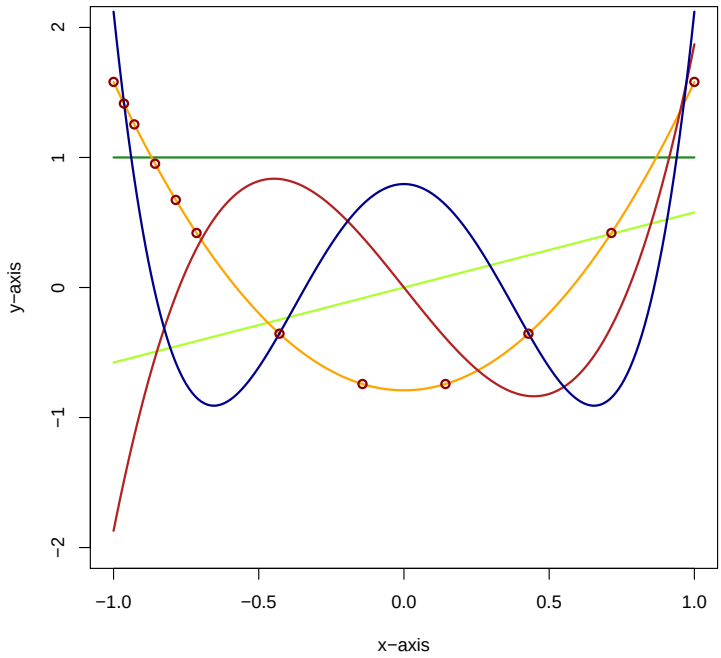


Figure 3.1: *The five first Legendre polynomials, with the orange points illustrating the values used as the discrete basis for the second order expansion.*

coefficients. If one coefficient is somehow biased, then that should not affect the estimation of the other coefficients, since they are uncorrelated.

For any given smooth (analytical n-differentiable) function f, one can approximate this arbitrarily well by letting the number of polynomial basis function go towards infinity (Taylor Theorem). The two next theorems shows this under different conditions for f, and are given without proof.

Theorem 3.6. ***Taylor's Theorem*** *If a function f has continuous derivatives up to order $n+1$, in a closed interval I, then the function f can be approximated as a linear combinations of polynomials, up to order n, giving*

$$f(t) = \sum_{i=0}^{n} \frac{f^{(i)}(c)}{i!}(t-c)^i + \mathcal{E}_{n+1}$$

for any t and c contained in the interval. Where the error term $\mathcal{E}_{n+1}$ is on the form

$$\mathcal{E}_{n+1} = \frac{f^{(n+1)}(\xi)}{(n+1)!}(t-c)^{n+1} = \mathcal{O}(h^{n+1}),$$

for some $\xi \in I$.

If the function has infinitely many derivatives, then the error term can be made arbitrarily small. However, if one only assumes that the function f is continuous, then there still exists a polynomial that approximates it arbitrarily well.

Theorem 3.7. ***Weierstrass Approximation*** *If f is continuous on a closed interval I, then, for any $\varepsilon > 0$, there exists a polynomial $p(t) \in \mathcal{P}^n$ such that*

$$|f(t) - p_n(t)| < \varepsilon \qquad \forall t \in I,$$

where n is finite.

This ensures existence of a best estimator under a uniform restriction. The Weierstrass approximation is in the sup norm, which may not be unique, and the theorem does not say how to obtain the best polynomial, unlike Tailor's Theorem, which gives an expression for all the coefficients. But, those might not be the best approximation under the $\mathcal{L}^p$ norm. Under squared error loss (the $\mathcal{L}^2$-norm), the minimizer will be unique and can be found. This follows from the projection properties in $\mathcal{L}^2$.

3.3.1 Projection

If the function f is contained in the space spanned by the basis functions, then the best estimator f^{best} is equal to the function f. When the function one wishes to estimate, is not contained in the space spanned by the basis functions, under the $\mathcal{L}^2$-norm, the best approximation is

$$f^{\text{best}} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} ||f - g||^2. \tag{3.8}$$

Which is the same as the minimizer under squared error loss. Projecting this down to the space spanned by the basis functions, the coefficients becomes

$$f^{\text{best}} = \sum_{\forall i} \langle h_i, f \rangle h_i, \tag{3.9}$$

if and only if, h_i constitutes an orthonormal basis. Otherwise, the basis has to be orthonormalized by the Gram-Smith algorithm.

If the estimator is unbiased, the estimated coefficients will be approximately the same as the coefficients in Equation (3.9). This is because the LM and REML often lies close to

the solution of least squares. In a test case where f is known or if one assumes something about the structure of f, then this can be used to calculate the bias for the approximation. This becomes important in model assessment shown in Chapter 5 with the introduction of Mean Squared Error and in Chapter 7 where it is applied to assess the effects of the sample size.

3.4 Bibliographic Note

Basis expansion is outlined in both Hastie et al. (2009) and Ramsay and Silverman (2005). The background on the Legendre polynomial are found in Kincaid and Cheney (2002), as well as a proof of Taylor's theorem (Theorem 3.6) and of Theorem 3.4. The inner product presented, and its applications are given in Young (1998). The Weierstrass approximation theorem with proof, can be found in Kreyszig (1978). A probabilistic representation of the same theorem can also be found in Karr (1992).

4

METHODS FOR ANALYSIS OF TIME COURSE GENE EXPRESSION DATA

In this chapter we present different methods that have been previously applied to time-continuous gene expression data. A selection of five articles and books is presented in an effort to provide some insight into the different kinds of approaches used.

4.1 Methods for Time Series

A method proposed already in 2005 by Storey et al., aims to identify *significantly expressed* genes, where significantly expressed refers to a significant change in the expression level over time. This method can also be used if there are two time course samples to be compared, i.e. case/control. The objective is to model some smooth function (basis expansion) of the data, one that is common to all data μ, and one where the case and control are modelled separately as μ_1 and μ_2. Then, the hypothesis that the case and the control are the same for a gene i, can be stated as

$$\begin{aligned} &\mathrm{H}_0 : \mu^i = \mu_1^i = \mu_2^i \\ &\mathrm{H}_1 : \mu_1^i \neq \mu_2^i. \end{aligned}$$

To test the hypothesis, a F-statistic is used to compare the fit of the models, both those using a common mean μ^i, and those where the mean differs, as μ_1^i and μ_2^i.

$$F_i = \frac{SS_i^0 - SS_i^1}{SS_i^1},$$

where SS^0 is the sum of squares under the null hypothesis, and SS^1 is the sum of squares under the alternative hypothesis. The sum of squares is calculated using the predicted values under the given hypothesis,

$$SS_i^0 = \sum \left(y_i - \hat{y}_i^0\right)^2 \qquad SS_i^1 = \sum \left(y_i - \hat{y}_i^1\right)^2.$$

The authors showed that this statistic will be approximately F-distributed (see their supporting appendix).

This method is highly flexible, allows for longitudinal clustering with different individuals and can handle missing values. Parts of this model have strong similarities with the LME

framework, in that the case/control group can have additional effect parameters. It is however, unclear whether the parameters contained in μ_1 and μ_2 are modelled as random or fixed effects. The authors have tested different basis functions for the mean value function μ including both spline, and polynomial basis. They argue that by setting the number of basis functions fix may lead to an inflation of significantly expressed genes due to overfitting. Although the polynomial basis was effective, the author decided to use a natural cubic spline (NCS) due to its flexibility, and that it makes fewer assumptions about the underlying structure of the curve. The model is implemented such that one person contribute to one time point, and they used multiple persons at different age to construct a smooth function describing the change in a particular gene over a long time interval.

The downside to this method is that the software presented (EDGE) is not properly maintained any longer, and does not work with R versions higher than 2.9.2. In addition, to use NCS (or splines in general) we think that the time course has to have a sufficient length to provide for estimation that are stable.

A more recent approach is from Ma et al. (2009), where they apply a method they call ANOVA LME. In this method, they look specifically at the time-treatment interaction and the mean function μ has even less restrictions than from EDGE, only to be piecewise smooth in time. The authors considered the model to be non-parametric, where the expression profile Y_{ki} for a gene i of subject k, under condition $g(k)$ at time point t_{ki}, given as

$$Y_{ik} = \mu(t_{ki}, g) + \mathbf{z}_k^T \mathbf{u}_k + \varepsilon_{ki}.$$

Where $\mathbf{z}_k^T \mathbf{u}_k$ are random effects, and the condition g can typically be a case/control indicator. The mean function μ can be decomposed to

$$\mu(t, g) = \tilde{\mu} + \mu_1(t) + \mu_2(g) + \mu_{1,2}(t, g),$$

here $\tilde{\mu}$ is the overall mean, $\mu_1(t)$ is the time effect, $\mu_2(g)$ is the group effect and $\mu_{1,2}(t, g)$ is the interaction with time and group.

The authors differentiate between genes where the interaction is non-zero (called NPDE Non-parallel Differentially Expressed Genes) and where the group effect is non-zero (PDE Parallel Differentially Expressed Genes). If a gene is PDE, then the interaction $\mu_{1,2}(t, g)$ is zero, and the difference in $\mu_2(g_1)$ and $\mu_2(g_2)$ are constant.

Ma et al. (2009) propose a method to test if a gene is NPDE using functional ANOVA, which under normal conditions has an asymptotically distribution as a mixture χ^2-distribution. The authors did point out that this approximation may not work well under finite sampling, and that bootstrap sampling could approximate the null distribution better. To test if a gene is PDE is the same as to test if $\mu_2(g) = 0$, discarding any time dependency. A two-way ANOVA test can detect a group effect if there are differences in expression levels for some groups. This method does seem somewhat to complicated, and the tests are preformed in a convoluted way, that may not fit well with questions asked in the introduction (Chapter 1).

From a Bayesian viewpoint, two main methods have been proposed, a Hidden Markov Chain model (Wei and Li, 2011) and an implemented computer algorithm known as "Bayesian software for Analysing Time Series microarray experiments" (BATS), by Angelini et al. (2008)). BATS uses hierarchical modelling of the gene expression observations, by introducing latent non-Gaussian variable, and a variable number of basis functions for each gene. For a Hidden Markov Chain model to work, the gene expression profile must have the Markov property, which is not likely to hold for most microarray time course experiments, accordingly to Ma et al. (2009). We found it desirable to first build a non-Bayesian model, and then expand the model framework if necessary and if there where sufficient time available.

4.2 Functional Data Analysis

Functional Data Analysis (FDA) is a concept where the data are thought of as coming from an underlying, smooth function, with little errors in the observations. It was popularized by Ramsay and Silverman in 2005, and have since become widely popular in medical statistics according to Coffey and Hinde (2011).

FDA relies heavily on basis expansion, but put it in a rigorous mathematical setting. Ramsay and Silverman (2005) displays a wide range of techniques built from continuous function estimation, from regularization to discriminant analysis and functional PCA. The mathematical basis of FDA allows for meaningful statistical inference, such an example is the functional t-test, presented in Coffey and Hinde (2011). The t-statistic is based on permutation of the different replicas of case/control, however if the number of replicas are small, this test breaks down.

FDA assumes that the underlying function is smooth, and thus with associated derivatives. These derivatives can give valuable information about the underlying phenomenon, and may deserve consideration as well. Using a discrete differential operator, FDA allows one not only to construct a model that fits the observed values, but also one that agrees with the observed derivatives. These derivatives can itself give great insight about the function/ generator and help to understanding the origin and structure of the data.

The discrete differential operator $\mathcal{D}$, takes in an array of observations $\mathbf{y}$, and returns a crude estimate of the derivative

$$\mathcal{D}(y(t_j)) = \frac{y_{j+1} - y_j}{t_{j+1} - t_j} \qquad \mathcal{D}(\mathbf{y}) = \dot{\mathbf{y}}.$$

This can be further extended to higher order differentials by taking the differentials from a differentiated vector of observations. This often is an essential in part of modelling in physics and biology where there exist a meaningful mathematical model for the derivatives. But for modelling time continuous gene expression however, rigorous mathematical

or empirical models for the derivatives do not yet exist. This result in that the although FDA presents many opportunities and strengths, its full potential may not be utilized. What often seems to be the case, is that there are many replicas/clusters used to make accurate construction of the expression profile. For few replicas, with an small number of observations in each, the method may appear to be less reliable. This since it has only been presented with a large number of replicas/clusters, e.g. (Ramsay and Silverman, 2005) and (Coffey and Hinde, 2011).

The FDA approach is to have some form of a *smoothing matrix* $\mathbf{S}_{\lambda,\{C\}}$ when takes in the observed data $\mathbf{y}$, and returns a smooth function that fits the observed data. Under some conditions/restrictions $\{C\}$, and often with a tuning parameter λ. Such that the predicted values becomes linear in $\mathbf{y}$, as

$$\hat{\mathbf{y}} = \mathbf{S}_{\lambda,\{C\}}\mathbf{y}.$$

A common smoothing penally is to use the integral of the second derivative of the estimated function,

$$\hat{f} = \arg\min_{f \in C} \left\{ \|f(\mathbf{t}) - \mathbf{y}\|_2 + \lambda \int_{\Omega} \left(f''(t)\right)^2 \mathrm{d}\, t \right\},$$

the solution to this minimization problem is the NCS. This follows from the regularization penalty, since the cubic spline is the function with a least second derivative, that still has high flexibility. See Green and Silverman (1994) for further details.

4.3 Correlation difference and metrics

Two proposed methods that do not explicitly use the time structure in their estimations are the Hilbert-Schmidt Independence Criterion (HSIC) by Shah and Corbeil (2011), and the Mean Absolute Rank Difference (MARD) by Cheng et al. (2006).

MARD was developed to rank the genes based on a correlation to its *neighbourhood*, where the neighbourhood can be defined in different ways, i.e. similarity in expression profile, large distance from the gene in terms of *behaviour*. The cut-off can then be set, and a number of genes will then be differentially expressed based on the MARD index. This index/statistic does not have any known distribution associated with it, and is a completely non-parametric test, using only permutation and cut-off values to determine if a gene is differential expressed under a case/control time course experiment.

A new and different way of looking at short time series microarray data, was given by Shah and Corbeil in 2011. They introduced the concept of differentiating genes using the HSIC with Reproducing Kernels Hilbert Space (RKHS). This approach is from a more machine learning perspective, and relies heavily on mathematical theory from advanced analysis.

An interesting observation is that very few articles published on the subject of time continuous gene expression estimation, seem to explore and model the Markov property. This might coincide with the often observed lack of any Markovian behaviour (observed in the ACF or PACF), or that the proposed mean value function seems to model the observed values sufficiently well thereby making the residuals behave almost as iid normal variables.

4.4 Summary

We have here presented a few selected articles and texts that we found most relevant. It should however be evident that most of the concepts presented here can not directly answer the question posed in the introduction. Although they provide insight and inspiration toward finding general methods for modelling gene expression profiles, they do not solve many of the problems regarding use of controls in a time series microarray experiment.

Had the EDGE software worked, it could have been used to estimated the time series. However, the ability for different persons to contribute with measurements at different time points to construct one time series does seems to little rigorous for our approach.

5

MODEL ASSESSMENT

This chapter outlines properties that are associated with an estimator of a some constant or function. Most notable is the Mean Squared Error (MSE), which is a composite of the variance and the bias of an estimator.

How the MSE is constructed and how it can be calculated are the main considerations in this chapter. Since an analytical calculation is not possible in many cases, we show how it may be approximated using Monte Carlo integration. We also suggest an expansion of the MSE to handle continuous curve estimates instead of point estimates. This can be used to evaluate an estimator under different conditions, and in this thesis it will be used to see how the sample size and model complexity affects the estimate.

5.1 Mean Squared Error

Defining and evaluating an estimator/predictor can be difficult since one first must define what is meant by a "good" estimator. A popular criterion for assessment of an estimator, is the mean squared error (MSE) which has well known mathematical properties.

Definition 5.1. *Assuming there is some constant parameter (or vector) θ, and a set of observed values $X_1 = x_1, \ldots, X_n = x_n$. We now wish to predict the value of θ in some point, using a random variable Z, assuming that there exists some relationship between θ and X, such that $\theta \approx Z|\mathbf{X} = f(\mathbf{X})$. Then the MSE of this predictor is defined as*

$$\mathrm{MSE}_\theta(Z) \stackrel{\mathrm{def}}{=} \mathbb{E}\left[(\theta - Z|X = x)^2\right],$$

provided that the expected value exists.

The Z that minimizes the value of MSE if it exists, is called the Minimum Mean Squares Estimate (MMSE) and is also known as a prediction in the $\mathcal{L}^2$-norm. Since the minimizer Z is a function of the observed random variables, it is itself a random variable, with associated expected value and variance.

Definition 5.1 is a general way of defining the MSE estimator, if the unknown quantity one wish to estimate is a constant θ, by using some statistic $\hat{\theta}$. The MSE estimator becomes

only a function of the observed values $Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n$,

$$\mathrm{MSE}_\theta(\hat{\theta}) = \mathbb{E}\left[(\theta - \hat{\theta}|Y = y)^2\right] = \mathbb{E}\left[(\theta - T(\mathbf{Y}))^2\right]. \tag{5.1}$$

Since θ is a constant, the only thing that influence the expected value is $\hat{\theta}$, through the statistic $T(\mathbf{Y})$. The Equation (5.1) can be a more easy way to grasp how the MSE is constructed, and with this definition of the MSE, it can then easily be broken down to the variance and the squared bias. This can become very handy when calculating the MSE of an estimator.

5.1.1 Bias-Variance Decomposition

Theorem 5.2. *The MSE of an estimator $\hat{\theta}$ for an unknown constant θ, can always be decomposed to the variance and squared bias. As*

$$\mathrm{MSE}_\theta(\hat{\theta}) = \mathbb{E}\left[(\hat{\theta} - \theta)^2\right] = \mathrm{Var}(\hat{\theta}) + \mathrm{Bias}^2(\theta, \hat{\theta}),$$

and is always positive or equal to zero.

Proof.

$$\begin{aligned}
\mathrm{MSE}_\theta(\hat{\theta}) &= \mathbb{E}[(\hat{\theta} - \theta)^2] \\
&= \mathbb{E}[\hat{\theta}^2] - 2\mathbb{E}[\theta\hat{\theta}] + \mathbb{E}[\theta^2] \\
&= \mathbb{E}[\hat{\theta}^2] - 2\theta\mathbb{E}[\hat{\theta}] + \theta^2 \\
&= \mathbb{E}[\hat{\theta}^2] - \mathbb{E}^2[\hat{\theta}] + \mathbb{E}^2[\hat{\theta}] - 2\theta\mathbb{E}[\theta] + \theta^2 \\
&= \left(\mathbb{E}[\hat{\theta}^2] - \mathbb{E}^2[\hat{\theta}]\right) + \left(\mathbb{E}^2[\hat{\theta}] - 2\theta\mathbb{E}[\hat{\theta}] + \theta^2\right) \\
&= \mathrm{Var}(\hat{\theta}) + \left(\mathbb{E}\left[\hat{\theta} - \theta\right]\right)^2 \\
&= \mathrm{Var}(\hat{\theta}) + \mathrm{Bias}^2(\theta, \hat{\theta})
\end{aligned}$$

□

Positivity of the MSE is obvious, since it is a sum of two non-negative parts. This Theorem can also be extended to continuous function estimators, however, the bias of the estimator may then not be uniquely defined.

The MSE can now be calculated if the variance and bias of an estimator are known or can be estimated. If the only criterion to be considered is an unbiased estimator, the estimator

may still have large variance and is thereby undesirable. However by allowing for some bias in the estimator, the variance could decrease substantially. This is why the MSE is often considered to give a better assessment of an estimator, since it contains both the variance and the bias. From an MSE point of view, large variance is just as bad as large (squared) bias, what counts is the sum.

5.2 Calculation of MSE

We will here turn to look at the MSE of an continuous estimator $\hat{y}(x)$ of some underlying function $y(x)$.

Calculation of MSE can sometimes be cumbersome, depending on the distribution of the estimator. If the probability distribution for the estimator is known, the MSE may be calculated directly. However, the true distribution is often unknown and even if it is known, the calculation can be to difficult to do analytically. In test cases, it is always possible to estimate the MSE by using Monte Carlo integration.

The idea behind using Monte Carlo integration in calculation of MSE, is the law of large numbers, such that, with enough simulations the average value of the square of the difference between $\hat{y}(x)$ and $y(x)$ will converge to the MSE. The MSE can either be approximated for each estimated value, thus giving a bound around the curve (similar to that of a confidence interval), or it can be summed up to give a single statistic of the estimated curve (the integrated MSE (iMSE)).

5.2.1 Pointwise MSE

For a set of observed values $\{y_i\}$, and corresponding estimated values $\{\hat{y}_i\}$

$$y_i|x = f(x) \qquad \hat{y}_i = T(x, i) \qquad i = 1, \ldots, n.$$

Then the MSE for the estimator, is the expected values of the squared difference, which also is a function of x. That is

$$\mathrm{MSE}(\hat{y}_i) = \mathbb{E}\left[(y(x_i) - \hat{y}(x_i))^2\right]. \tag{5.2}$$

By the strong law of large numbers, when the numbers of simulations k increases, an estimate for the MSE is then the average of the observed values minus the predicted at a point x_i,

$$\lim_{k\to\infty} \frac{1}{k} \sum_{j=1}^{k} \left(y_i^{(j)} - \hat{y}_i^{(j)}\right)^2 \xrightarrow{\text{a.s.}} \mathbb{E}\left[(y(x_i) - \hat{y}(x_i))^2\right], \tag{5.3}$$

provided that $\mathbb{E}\left[|\hat{y}_i - y_i|\right] < \infty$, and that $\hat{y}_i$ and y_i are iid. This equation however, is only valid when the number of samples goes to infinity. To overcome this problem, bootstrap or Monte Carlo integration can be used to approximate the right hand side of Equation (5.3). Using Monte Carlo integration, the underlying function has to be known. This is only true in a test cases, where the generator function f is constructed, along with the random error. Then the fitted model is compared with the true function, and one realisation of the left hand side of Equation (5.3) is obtained.

5.2.2 Integrated MSE

For a point estimator, the MSE can be seen as the expected value from Equation (5.1). However, for a continuous estimator, for example a function defined on an interval, a summary statistic may still be desired if one wishes to compare two or more curve estimates. One way of doing this is to integrate the estimated function, giving the integrated MSE (iMSE) as;

$$\mathbb{E}\left[\int (f - \hat{f})\,\mathrm{d}\,x\right]^2 = \int \left[\int (f - \hat{f})\,\mathrm{d}\,x\right]^2 \mathrm{d}\,\mathbb{P}. \tag{5.4}$$

One can observe that since the function $\hat{f}$ is a random variable, this integral is not well defined. However, in Monte Carlo simulations, f will be known and the estimated statistic $\hat{f}$ is a single realization on a defined interval. It is then possible to obtain one realisation of the inner integral in (5.4).

Using the law of large numbers with realisations from $\hat{f}_i$, and f known, the estimated value $\widehat{i\mathrm{MSE}}$ is then

$$\lim_{k\to\infty} \frac{1}{k} \sum_{j=1}^{n} \left(\int \left(f - \hat{f}^{(j)}\right)\mathrm{d}\,x\right)^2 \xrightarrow{\text{a.s.}} \mathbb{E}\left[\left(\int (f - \hat{f})\,\mathrm{d}\,x\right)^2\right], \tag{5.5}$$

provided that $\mathbb{E}\left[\left(\int |f - \hat{f}|\,\mathrm{d}\,x\right)\right] < \infty$ and that $\hat{f}^{(j)}$ are iid, where $\hat{f}^{(j)}$ is the j'th simulation.

Then, by Jensen's inequality, the integral of a squared function is greater then, or equal to the square of the integrated function, such that

$$\frac{1}{n}\sum \left(\int \left(f - \hat{f}^{(i)}\right)^2 \mathrm{d}\,x\right) \geqslant \frac{1}{n}\sum \left(\int \left(f - \hat{f}^{(i)}\right)\mathrm{d}\,x\right)^2. \tag{5.6}$$

By this, the left hand side in Equation (5.6) will be an upper bound of the approximation of the MSE for some estimator $\hat{f}$.

In a Monte Carlo study, the sums in Equation (5.6) are easy to approximate, knowing the underlying generator f and the estimated function $\hat{f}$. Assuming both functions are well behaved, numerical integrations will be both accurate and fast.

This leads to the discussion of a natural extension of bias to continuous estimates. In its most basic form, the variance and bias (MSE) can be calculated for a single point (often as a function of the number of observations). When the estimate is a continuous function, the MSE will then also be a continuous function. What is then the best summary of this function? The prime candidates are the $\mathcal{L}^1$ and $\mathcal{L}^2$ norm plus the maximum norm. While the maximum norm can be considered somewhat stronger then the $\mathcal{L}^p$ norm, the $\mathcal{L}^p$ norm may be preferable due to their inner product properties. Another key difference is that the $\mathcal{L}^p$ norm is strongly dependent on the interval of which the estimator defined on due to the integral in the calculation. This is not the case for the maximum norm, but the maximum norm will also not detect large deviation over a substantial interval.

The whole problem of analysing bias to continuous estimate, is then reduced to a choice of norm. This choice is dependent on the desired properties, and possible outcomes for the bias statistic.

5.3 Simulation and Estimation

Since the integration in Equation (5.5) has to be done numerically, it can be done with on a dense grid (i.e. to let $\Delta x_i \to 0$). This gives the continuous $i\widehat{\mathrm{MSE}}$, where the name continuous reflects that it is estimated almost continuous in x. Another approach with similarities to the sum of squares, is to let Δx_i be the difference between the observations. That is to let $\Delta x_i = x_{i+1} - x_i$ which is denoted discrete $i\widehat{\mathrm{MSE}}$, this since the approximated integral does not achieve that same accuracy as the continuous $i\widehat{\mathrm{MSE}}$.

$$\lim_{m,j\to\infty} \frac{1}{jm} \sum_{j} \sum_{k=1}^{m} \sum_{i=1}^{n} \left(\hat{y}_i^{k(j)} - y_i^{k(j)} \right)^2 \Delta x_i \approx \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \left(\int \left(f - \hat{f}^{(j)} \right) \mathrm{d}\, x \right)^2 . \tag{5.7}$$

Here the subscript $i \in n$ refers to the actual observations, the label k is the cluster (LME), and the superscript (j) is the number of Monte Carlo replications The two different approaches are closely linked, in that they approximates the same quantity but with different margins of error.

The discrete $i\widehat{\mathrm{MSE}}$ can be viewed as a weighted sum off squares where the weight is the distance between the time points, thus it represents a rough estimate of the area of the difference between the estimated curve and the true underlying curve. Comparing this to the same formula but not using the true underlying function, instead measuring the area between the observed values and the estimated values, weighted with the distance between the measurements is called the observed discrete $i\widehat{\mathrm{MSE}}$. Both this two estimates should be quite close to the sum of squares in most cases, but may deviate somewhat if the distances (weights) $(x_{i+1} - x_i)$ are much larger then one. But, it may still be of interest to compare these methods.

The behaviour of the $i\widehat{\mathrm{MSE}}$ is examined in more detail in Chapter 7, where it is used to asses the effect of adding or removing data from an estimator.

5.4 Bibliographic Note

Some of this chapter is inspired by Hastie et al. (2009), with Definition 5.1 (slightly altered) from Karr (1992) along with a proof of the strong law of large numbers for the iid case (the variance does not need to be finite). For a proof of LLN if the observations not are identically distributed, only independent and finite variance see McDonald and Weiss (1999).

We found non articles that discussed and extended the bias-variance decomposition to the estimation of continuous curves, and how this relates to the bias-variance in the estimated underlying parameters of the curve.

6

THE MODEL FRAMEWORK

This chapter accumulates what is presented in the previous chapters, and we present the models used, and the settings applied to them.

6.1 Presenting the Models

6.1.1 Basic Model

The simplest model we used was an LME model with a fixed effect function $\eta_i(t)$ and one random effect u_{ki} for each gene i.

$$f_{ik}(t_j) = \eta_i(t_j) + u_{ki} + \varepsilon_{jki} \qquad t_j \in [t_0, t_{12}]\, k = 1, 2,\ i = 1, \ldots, n_k, \tag{6.1}$$

where k is the biological replica, $\eta_i(t)$ is the mean value function for gene i, and t_j are the time points given in Table 8.1, after normalization. Then each gene gets an estimated mean, and for each biological replication (cluster) of that gene, there will be a cluster specific offset. The function $\eta(t)$ is a linear combination (basis expansion) of the first four orthonormalized Legendre polynomials.

Basis expansion is also possible for random effects u_k, but may not be suitable, this due to a number of reasons. If there is little evidence that each expression profile has additional random effects, then the terms should not be included in the model.

6.1.2 Extension with Backwards Dependences

Expanding the model with additional covariance structure, as suggested in Section 2.2, we here chose to use an AR(1) model for the errors. This should remove time dependency of lag one in the observations that is not explained by the mean value function.

Formulated mathematically for one gene, the model becomes

$$\begin{aligned} f_k(t_j) &= \eta(t_j) + u_k + \varepsilon_{kj} \\ \varepsilon_{k,j} &= \phi\, \varepsilon_{k,j-1} + \varepsilon^{\star}_{kj} \\ \varepsilon^{\star}_{kj} &\overset{\mathrm{d}}{=} \mathcal{N}(0, \sigma_{\varepsilon^{\star}}), \end{aligned} \tag{6.2}$$

where $\eta(t)$ is the same function as in Equation (6.1). Observe that since there is an AR-dependency in the errors, then there is also an AR-structure in the observed values.

To justify this increased complexity, there has to be a substantial impact of fit to the data. In order to test if there is such a dependency in the data, one can use an ANOVA test or a Likelihood Ratio test to compare two models, with and without an AR-dependencies (formulated in Equation (8.3)).

6.2 Classification and Clustering

To investigate further, one can classify each gene into different classes (not to be confused with the clusters used in LME). A class can for example be, the group of all genes that are significantly expressed. Elements from machine learning could also be used to find natural clusters in the data. In this thesis, we used an expansion of the K-means algorithm known as Partitioning Around Medoids (PAM).

6.2.1 Significantly Differentially Expressed Gene

A gene is classified to be significantly differentially expressed, based on the p-value from the ANOVA test presented in Section 2.4. If one of the coefficients in the LME fixed effect is different from zero we have an estimated gene expression profile that is different from zero. The estimated curve is not equal to zero in at least one point.

Another classification could be the genes where the frame of reference disagrees, that is when using an unstimulated control gives a different answer than using only the staring point to determine changes in the expression, see Chapter 8 Section 8.3 and Table **??**.

6.2.2 Cluster Analysis with Medoids

Partitioning Around Medoids (PAM) is an extension of the K-means algorithm and is used to find an optimal number of clusters, where the clustering is more loosely defined than in K-means. The genes are assigned to a cluster according to some similarity or distance measure. Like most clustering method, this method is highly dependent on the choice of similarity. Common choices for the distance measure is the euclidean distance

and the maximum distance. The euclidean distance measures the sum of the squared difference between two genes at all time points, and the genes that minimizes this distance is considered closest (most similar).

Similarity can also be based on the correlation between the genes, through the correlation distance metric. This measure is also called Pearson distance measure, and is based on the Pearson product moment correlation coefficient. The distance is often defined as

$$d(x, y) = 1 - r_{x,y} \tag{6.3}$$

where $r_{x,y}$ is the Pearson correlation between gene x and y, over all time points. One can observe that the distance measure in Equation (6.3) is finite, i.e. it has a range of $[0, 2]$, with increasing distance if the vectors are highly negatively correlated.

6.3 Frame of Reference

The *frame of reference* is how a change in the gene expression is observed or measured, as such, it is the reference point to determine a change in the gene activity for a case/control experiment. There are three common reference frames:

1. To measure the activity relative to the change from the starting point (t_0).
2. To measure the activity relative to an unstimulated control sample at each time point.
3. To measure the activity relative to the change in internal housekeeping genes.

Only the two first will be considered in this thesis, this because they are quite similar in the estimation approach. Since both references give a single time series of the same form, the model in Equation (6.1) is used in both cases. All these different approaches will affect the whether a gene is classified as significantly expressed or not. This makes the choice of reference important when deciding which gene is affected by the treatment (gastrin) and which is not affected.

6.4 Bibliographic Note

We found no scientific publication that have in detail discussed the frame of reference and its implications. In most articles where there is some control mechanism similar to that of an unstimulated control sample, the approaches have not been sufficiently clarified and/or justified.

7

RESULT FROM MONTE CARLO SIMULATIONS

To assess performance of the models, and how the number of clusters affected the precision and error, we conducted a large Monte Carlo simulation. The algorithmic procedure of this approach can be found in Appendix C, and some additional R code is given in Appendix D.

The Monte Carlo setting was constructed by first choosing a generator function, then to create observations by adding random errors. We then tried to recreate the generator function, using an implementation of LME in R. Each realization was then give a score, according to the estimated $i\widehat{\mathrm{MSE}}$ value (see Equation (5.5) and (5.6)), the $i\widehat{\mathrm{MSE}}$ scores presented is always integrated on a dense grid, if not otherwise stated. The number of time points in each LME cluster were the same number as in the actual microarry experiment (12, see Table 8.1), unless otherwise stated.

Due to the large number of simulations done, the code had to be parallelized using multiple processor cores. This can easily be done in R using an implementation of the Message Passing Interface (MPI).

7.1 The Parameter Impact

One key issue in the Monte Carlo simulation, was to select realistic parameters for a gene expression setting. The parameters that need to be set, are the covariance parameters (σ_R^2 and σ_D^2), and the generator function f. The parameters where chosen using the data set presented in the introduction (Chapter 1).

Based on the observed variances, using a kernel density estimation from Figure 7.1, we picked three values for the residual variance σ_R^2, and present the two simulation with highest value of σ_R in Figure 7.2. In the same figure we tested the effect of varying the ICC, this instead of selecting the variance of the random effects (σ_D^2). For the generator curve f there are two choices, one is to use a function that is contained in the space spanned by the basis functions ($f \in \mathcal{H}$), or it can be any continuous function that is not a 4th order polynomial. However, some functions are more realistic than others, and in our test case, we picked the transcendental function $\exp(2t)$ and a 4th order polynomial. If the function is an element in the basis space, then the maximum likelihood is (asymptotically) unbiased,

and the estimated $\widehat{i\mathrm{MSE}}$ should then only reflect the variance, since the estimated curve should have no systematic bias. Any difference between the estimated curve and the true underlying generator function, could be contributed to random error, and instability due to lack of data (finite sampling).

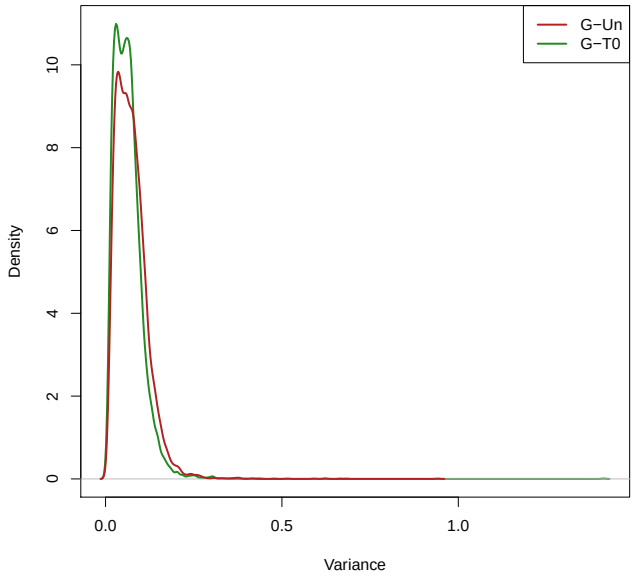
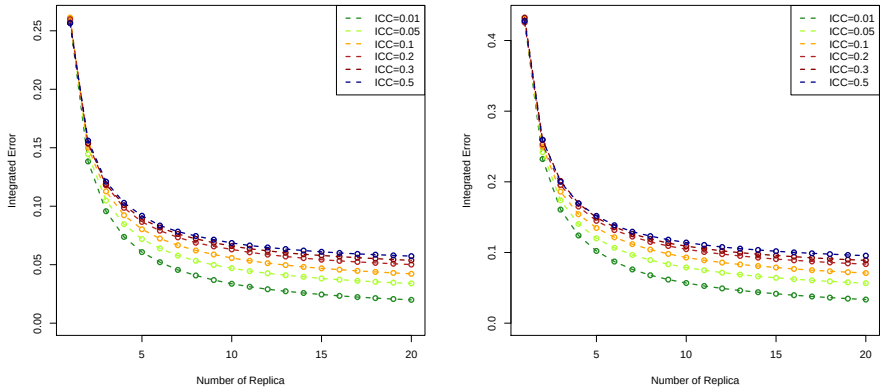


Figure 7.1: *Estimated density of the variance σ_R^2 from the fitted LME, using kernel density estimation. The variance is both for the gastrin minus t_0 series (green), and gastrin minus the unstimulated control (red).The bandwidth is automatically selected by R.*

From Figure 7.2 it became clear that lower a ICC gives a higher precision, and also that the impact of σ_R^2 is proportional to the iMSE. An increase in 66% in σ_R^2 gives approximately 66% increase in iMSE. The reduction in iMSE when the ICC goes down, can be explained by the fact that all the data point have lower correlation, and thereby behave almost independent. I.e. every new data point added, gives more information then if they were correlated. Figure 7.2 also showed that for a low number of clusters, the iMSE estimates were closer than for a higher number of clusters.



(a) $\sigma_R^2 = 0.3$ **(b)** $\sigma_R^2 = 0.5$

Figure 7.2: *Simulation for different number of clusters with fixed residual variance and changing ICC, each with 12 time points. The number of simulations is* 10 000*, and the test function is a polynomial of degree 4. Note the differences on the scaling in the y-axis, which both represents the average value of the estimated error.*

7.1.1 On the Number of Clusters Used

In the introduction, we asked if it is enough to only use two biological replicas. To analyse this, we started the Monte Carlo simulation with a high number of replicas (20), the data set were sequentially reduced with one replica, and a new estimate of the mean value function where obtained, based on the reduced data set. This would tell us what impact the size of the data set had on the $i\widehat{\mathrm{MSE}}$ estimate. One can observed from the Figure 7.2 a clear downwards trend, until the $i\widehat{\mathrm{MSE}}$ stabilizes.

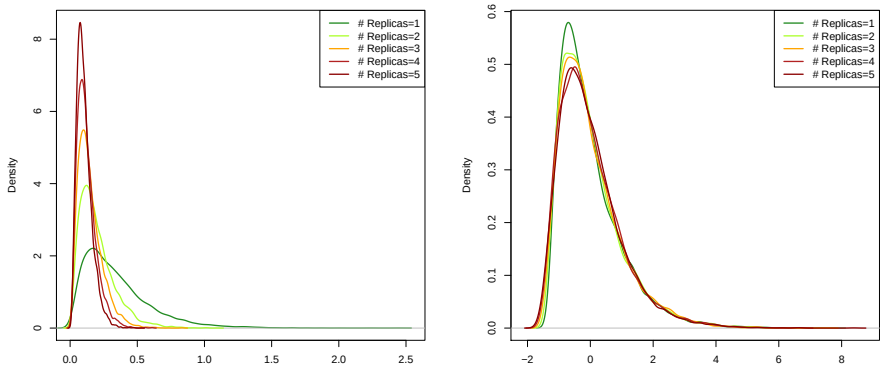
It became clear that for small data samples, a large decrease in the $i\widehat{\mathrm{MSE}}$ could be obtained by adding another replica. However, if one assume that the error is small (in the order of magnitude 10^{-1}), the precision will still be high with only a few replicas. There appeared to be a fast drop in the $i\widehat{\mathrm{MSE}}$, until it slowly converged to its minimum. This is clearly shown in Figure 7.2, where the $i\widehat{\mathrm{MSE}}$ stabilizes from around 10 clusters, this means that there is no additional gain in precision after around 10 replicas, given that each replica contains 12 time points. As the number of clusters increases, the difference between the $i\widehat{\mathrm{MSE}}$ estimate also increases, for few clusters the difference is not substantial, but after stabilizing the difference seems almost constant.

7.1.2 *Empirical Distribution of* $i\widehat{\mathrm{MSE}}$

Since the distribution of the $i\widehat{\mathrm{MSE}}$ is unknown, the results from the Monte Carlo study can be difficult to interpret. The $i\widehat{\mathrm{MSE}}$ is constructed such that it can be used to compare the performance of different continuous models, but with no known probabilistic behaviour. We therefore plotted the distribution over all Monte Carlo samples for the five first replicas, which is given in Figure 7.3.

Normalizing the observed $i\widehat{\mathrm{MSE}}$ by subtracting the mean and dividing by the standard deviation, it seems as their distribution tend somewhat towards a normal distribution. However, it is skewed to the right, with a tail that is far to heavy to be a "true" normal distribution. This can be observed in Figure 7.3, where both the normalized and unnormalized (raw) density estimation is plotted.

From Figure 7.3a it became clear that the spread in the $i\widehat{\mathrm{MSE}}$ scores is decreasing as the number of replicas goes up, this is somewhat intuitive, as that the accuracy of the estimation should go up when the number of data points increases. When the $i\widehat{\mathrm{MSE}}$ is normalized the distributions seems to come from the same scale-location family, as they appear to all have similar shapes in Figure 7.3b.



(a) *Density estimate of error over all Monte Carlo simulations for the cumulative five first replicas.*

(b) *Normalized density estimate over all Monte Carlo simulations for cumulative five first replicas.*

Figure 7.3: *Kernel density estimate of continuous* $i\widehat{\mathrm{MSE}}$ *over all Monte Carlo simulations, with* $ICC = 0.1$ *and* $\sigma_R^2 = 0.4$*, the bandwidth is automatically selected by R. The test function is a 4th order polynomial. On the x-axis is the estimated* $i\widehat{\mathrm{MSE}}$ *and the y-axis represents the density.*

The tail heaviness if Figure 7.3b is not so easy to observe for the higher number of replicas in Figure 7.3a, and does not have any intuitive explanation. What is also unexpected, is the similarities in Figure 7.3b, as the density for only one replica is far more tail heavy than the rest in Figure 7.3a.

Since the $i\widehat{\mathrm{MSE}}$ measure is only useful in test cases, we constructed a simulation where we compared the continuous $i\widehat{\mathrm{MSE}}$, the discrete $i\widehat{\mathrm{MSE}}$ and the observed discrete $i\widehat{\mathrm{MSE}}$, plotted in Figure 7.4. This to see how the discrete $i\widehat{\mathrm{MSE}}$ would relate to the others if the underlying function is unknown. As one can observe from this figure the result was somewhat surprising, this since estimated error increased with the amount of data points before stabilizing. The figure showed that for a few number of clusters, the estimated observed discrete $i\widehat{\mathrm{MSE}}$ could be to optimistic compared to the same estimation using the true underlying curve. The result also indicate that even if the observed discrete $i\widehat{\mathrm{MSE}}$ is large, the estimated curve (parameters) may still fit very well with the true underlying function. The observed discrete $i\widehat{\mathrm{MSE}}$ (plotted in read in Figure 7.4) seemed larger then the estimated error in Figure 7.1, which may make it somewhat dubious as an estimator of fit, in cases where the true underlying function is not known.

In Figure 7.4, the continuous $i\widehat{\mathrm{MSE}}$ (green) is calculated using a numerical approximation of Equation (5.5) on a dense grid for each cluster, which is the same as all the other $i\widehat{\mathrm{MSE}}$ presented so far. For the discrete $i\widehat{\mathrm{MSE}}$ (orange) the same formula was used, but with only the time points as the step size in the grid. The difference between the discrete and the observed discrete $i\widehat{\mathrm{MSE}}$ (red), is that in the observed $i\widehat{\mathrm{MSE}}$, only the observed values and the predicted values is used, where the predicted values and the known underlying values is used in the discrete $i\widehat{\mathrm{MSE}}$.

7.2 On the Number of Time Points

From the figures in Figure 7.2, there seems to something to gain by increasing the number of biological replicas. Figure 7.5 supports this, as the models with highest number of time points in each replica have the lowest $i\widehat{\mathrm{MSE}}$, which is a natural consequence of additional data. But, if the number of experiments is to be kept constant, the number of time points in each replica have to be decreased in order to increase the number of replicas. A question is then; what does then happen if the number of time points in each cluster is reduced? With a fixed ICC and σ_R^2, on can simulate the effect of changing the number of time points. The trouble is that when the number of time points goes down, the number of basis functions is still constant. This will reduce the number of degrees of freedom in the simulation, and cause higher instability in the estimated function.

Reducing the number of time points beyond a certain point, the $i\widehat{\mathrm{MSE}}$ seems to be higher

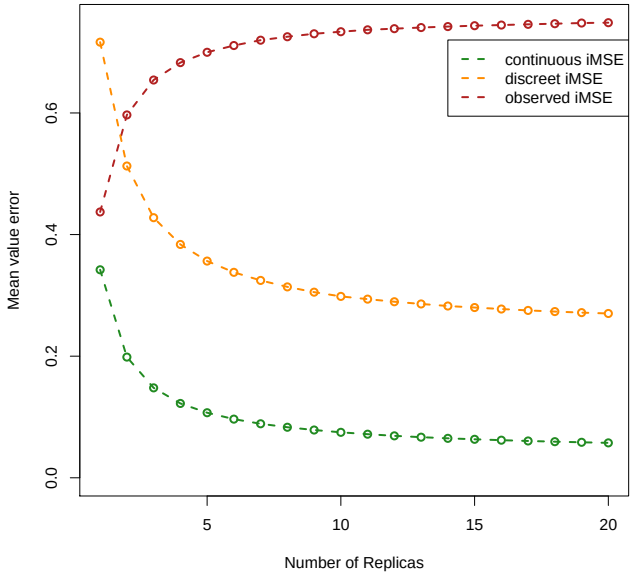


Figure 7.4: *Comparison of the continuous* $i\widehat{\text{MSE}}$ *and the* $i\widehat{\text{MSE}}$ *estimate based only on the observations. The continuous* $i\widehat{\text{MSE}}$ *is based on a calculated on a dense grid, while the discrete* $i\widehat{\text{MSE}}$ *is calculated only in the time points.* $\sigma_R^2 = 0.4$ *and ICC is* 0.1*, the number of simulations is* $30\ 000$*, and the generator function is* $f(t) = e^{2t}$*.*

than for the same number of total experiments but with higher number of clusters, seen in Figure 7.5. This shows that there is clearly a trade-off between the number of time points and the number of clusters. To see if there were an optimal number of clusters, with a fixed total number of time points, we simulated different arrangement of time points and clusters, but with the total sum fixed. In Figure 7.6 we observed that it is clearly better to increase the number of clusters, and thus decrease the number of time points in each cluster. A problem arises when the number of time points in each cluster is not high enough to cover the decreased number of degrees of freedom, due to loss of time points. This since the mean value function itself uses a number of parameters, as well as each cluster needs its own offset, then there might not be enough time points left in each cluster to account for this.

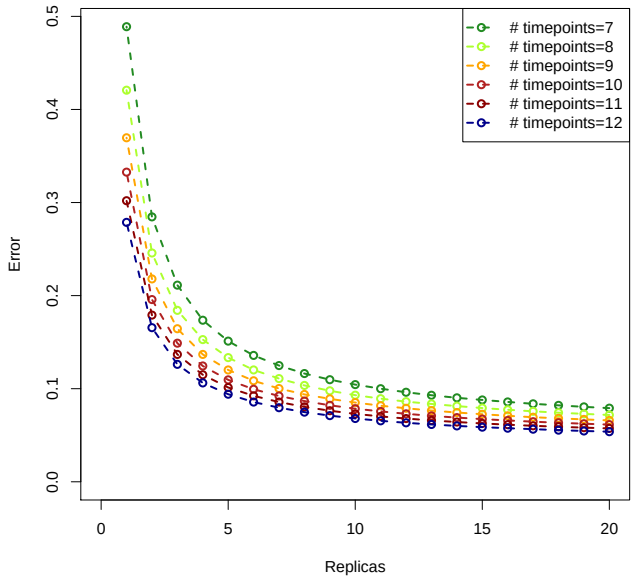


Figure 7.5: *Effect of changing the number of time points, 12 time points is used as a reference to compare the other alternatives. The number of simulations is* $30\,000$*,* $ICC = 0.1$*, and variance* $\sigma_R^2 = 0.4$*, the underlying function* f *is* e^{2t} *and the number of basis functions is 5 (4th order Legendre polynomial). The error on the y-axis is the continuous* $i\widehat{\text{MSE}}$.

7.3 Discussion

All the results here originates from computer simulations, and are thus built upon a simple known underlying model. This will affect all aspects of the conclusions drawn, but if the test case is close to reality, valuable information can still be obtained. The exponential functions was chosen due to its complexity as a Taylor polynomial and its smoothness, and by setting it to $\exp(2t)$, will give higher value on the higher coefficients (4th and 5th order). Still, as seen in Figures 7.7, even few polynomials will approximate this function well. The figure uses only four basis functions (3rd order polynomial), and with five, there is almost no detectable difference. It turns out that if the true underlying function is smooth with no fast drops or spikes, a low order polynomial will work reasonably well. However, if the generator function has a drop faster than a 4th order polynomial can follow, the bias can ends being substantially. But, this may not be very realistic in a gene expression

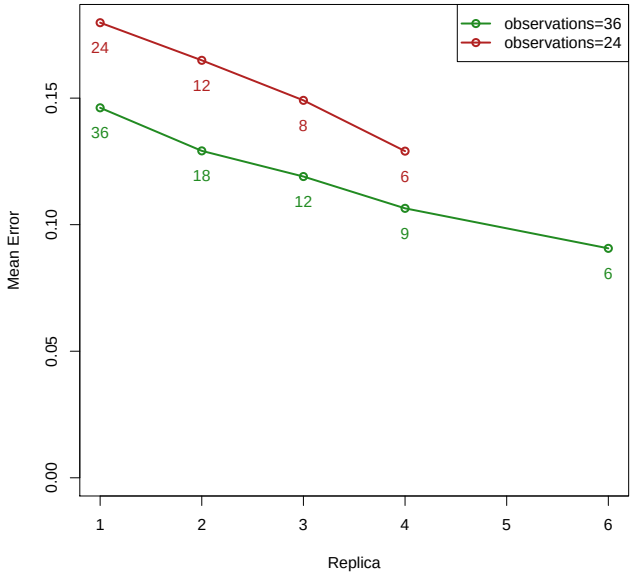


Figure 7.6: *Different arrangement of time points and clusters, given a fixed amount of experiments. The numbers indicates how many time points there are in each cluster. The number of simulations is* 80 000*,* $ICC = 0.1$*, variance* $\sigma_R^2 = 0.4$*, and* $f(t) = e^{2t}$*. The error on the y-axis is the continuous* $i\widehat{\text{MSE}}$*.*

setting, and should be detected in an array of diagnostic checks (e.g. if the residual variance considered to large).

If the function was not contained in the basis, with high coefficients in higher order polynomial (typically, around a 9th order polynomial), the bias was disastrous. Still, for a transcendental function, as $\exp(x)$ and $\sin(x)$, which can be defined as an infinite polynomial series, the approximation using only a small number of polynomials (e.g. 4), works reasonably well if the coefficients higher of higher order, do not become dominant. This means that any change that is not growing to fast or has a sudden steep drop, can be approximated well by a polynomial basis on the interval $[-1, 1]$. This can be seen in Figure 7.7, where a curve is approximated deterministically to the transcendent function $\exp(2t)$, with only three basis functions.

For a fixed number of time points, it then seems optimal to increase the number of cluster and place fewer observations within each cluster. The $i\widehat{\text{MSE}}$ estimate for the last cluster in

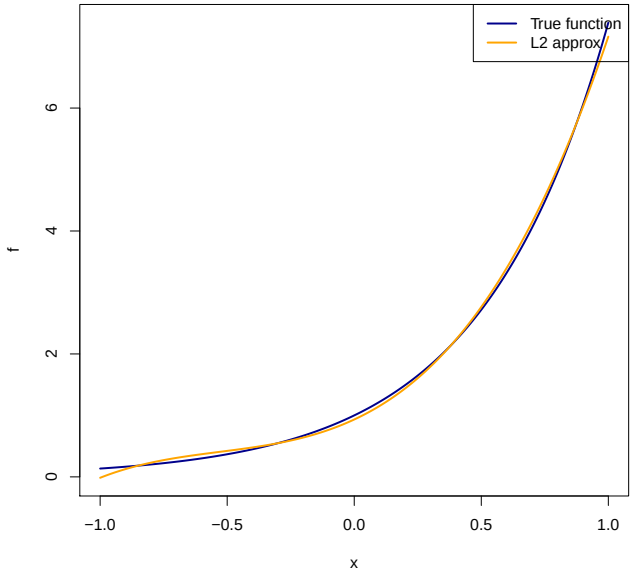


Figure 7.7: *Deterministic approximation of* $\exp(2t)$ *using the* $\mathcal{L}^2$ *norm (squared error loss), with the four first Legendre polynomial (including order zero). This shows the best possible theoretical approximation with infinite uncorrelated data points. (The basis is orthonormalized).*

Figure 7.6 may be to optimistic, since to few observations in each cluster will negatively affect the stability of the estimate, due to the loss of degrees of freedom.

The low ICC in the simulations may not reflect of true ICC in microarray experiments, the values of $0.5 - 0.01$ have however, been chosen on findings in the actual data presented in Chapter 8, as a conservative estimate. For a ICC less then 0.1 it has been suggested that an ordinary linear model will give approximately the same results as the LME, since the data appear to be uncorrelated. But with the development in computer technology and new fast algorithms, a low ICC is now longer a concern and the LME framework is used whenever the data structure dictates it.

From Figure 7.4 it became clear that the $i\widehat{\text{MSE}}$ estimation using the observed values only, gives a high error estimate, but could maybe reflect more of the variance, since the value lies closer to the σ_R than the two other calculated $i\widehat{\text{MSE}}$. That there is a difference is not surprising, but the magnitude may be considered somewhat large, and the fact that it

increases and not decreases. All the three error estimates seems to stabilize at the same number of clusters (around 10), but while the true error is decreasing, the observed error goes up before stabilizing. The difference in in the two discrete $i\widehat{\mathrm{MSE}}$s should not be large, and intuitively the observed $i\widehat{\mathrm{MSE}}$ should be almost the same as the true discrete $i\widehat{\mathrm{MSE}}$, as the only difference is that the observed has been adjusted for by the errors (ε_i). When evaluated on a dense grid, the estimated continuous $i\widehat{\mathrm{MSE}}$ is less then σ_R^2, this has to do with the close fit of the estimated function and the true function, and what is integrated is the squared difference between the generating curve $f + u_k$ and the estimated curve $\hat{f} + \hat{u}_k$. Still, it can give an indication of how well the method works on the observed (simulated) data, but works best for comparing different model in a simulation setting.

Summing up from this chapter, what we have found is that a higher number of clusters is desired, even if it means lower number of time points in each cluster. But, the clusters should not be so small that they compromise the integrity of estimated function. This is also according to medical established routine, as one wants as many individual samples as possible. From the error estimate, an approach with LME and low order basis expansion seems to work surprisingly well even with a transcendental function with somewhat high Taylor complexity.

7.4 Bibliographic Note

Two **R** packages were used in the simulation, the `Rmpi` for parallelization, and `nlme` package for the LME framework. The Rmpi is implemented in R by Yu (2002), based on the Message Passing Interface. The simulation was done using the `lme()` function to reconstruct the curve, implemented by Pinheiro et al. (2011). A number of additional packages was used, and a complete list is found in Appendix E.

8

GASTRIN EFFECT ANALYSIS

This chapter presents some of the findings based on the data set from the experiment introduced in Chapter 1, with the aim to answer the question raised in that chapter. The issue of the number of biological replicas was discussed in the previous chapter, leaving here the analysis on the frame of reference.

Note: when talking about significance and a significant gene, we mean a gene that has a detectable significant change in the expression profile over the considered time interval, with a cut-off of 0.05 on the adjusted p-values from the Benjamini & Hochberg method.

8.1 The Data Set

The time points at which the measurements where taken, are presented in Table 8.1. We chose to normalize the time points to the interval $[-1, 1]$, with the internal distance preserved. This was done using the equation;

$$t^{\text{new}} = \frac{2t}{\max(t)} - 1.$$

The normalization makes the time points more suitable for modelling with the Legendre polynomials, and since the scaling preserves the internal distances, no information is lost. This also scale the problem down, which reduces the probability that the polynomials behave erratic at the boundaries.

Table 8.1: *Time table for measurements, in the microarray study.*

Time points for data measurements in minutes and hours												
min	0	15	30	60	90	120	240	360	480	600	720	840
hour	0	.25	.5	1	1.5	2	4	6	8	10	12	14

Cells treated with gastrin were harvested at 11 different time points between 15 minutes and 14 hour, given in Table 8.1. The samples from untreated control cells were harvested at time zero and throughout the time course with the treated sample (12 time points)[1].

[1] Which are the exact same time points as for the gastrin stimulated sample.

Two independent experiments (biological replicates) were performed. Afterwards, the data were preprocessed using a $\log_2$ transformation and a quantile normalization. The experimental set up was such that the gastin stimulated and the unstimulated control samples were coupled, i.e. measured at the same time, and thereby should be subject to the same external forces outside of the experimenter's control. The expression was measured for close to 8960 genes.

The observations can then be presented as two primary time series

$$U_t \qquad t = 0, \ldots, 11 \tag{8.1a}$$
$$G_t \qquad G_{t_0} = U_{t_0}. \tag{8.1b}$$

Here G_t is the gastin stimulated sample, and U_t is the unstimulated control sample. There are two replicas for each gene, however, from a notation point we let the two biological replicas be contained in G_t and U_t. The secondary time series can be formulated as

$$X_t = G_t - G_{t_0} \qquad t = 0, \ldots, 11 \tag{8.2a}$$
$$Y_t = G_t - U_t \qquad G_{t_0} = U_{t_0}. \tag{8.2b}$$

On each microarray chip, there where additional negative control spots. The negative control spots are synthetic RNA from fish (unknown species) which is known to not exist in the cell line under study (AR42J). The binding at this spots should then in theory be only background noise.

8.2 Data Analysis

For each gene, an LME model was then constructed separately for each of the time series U_t, X_t, and Y_t. As a basis we used the five first (orthonormalized) Legendre polynomials (a 4th order polynomial), and each replica was given a random effect. Next, the basis was reduced to the four first Legendre polynomials, this to see what effect a less flexible basis has on the significance of the estimated expression profiles. To see if there was a significant change time, it is sufficient that at least one fixed effect parameters are larger than zero. This is true expect for intercept which only determines the baseline of the expression. The test can be preformed with the ANOVA method given in Equation (2.19) giving a p-value for the test against the null model, which does not contain any parameters except for intercept.

8.2.1 Lack of Backwards Dependency

Since there is a clear ordering of the data, there is the possibility that all errors are correlated in time even after the trend have been removed. Such a sequence of observations

that are correlated in time may be modelled with an AR(1) model, as presented in Equation (6.2).

An AR dependency can be detected by visual inspection of the ACF, for each gene. However, since the number of genes is so large, this is not feasible. Our approach was to fit both an LME with and without an AR covariance structure. We then compared the two models, using an ANOVA test, to keep or reject the null hypotheses stated in Equation (8.3).

$$\begin{aligned} &H_0 : \text{LME without AR(1)} \\ &H_1 : \text{LME with AR(1).} \end{aligned} \tag{8.3}$$

The result from this test should tell which model to prefer for the tested gene, with a low p-value as evidence against the null hypothesis. From Figure 8.1 it is clear that adding additional AR structure to the error, does not improve the fit of the model.

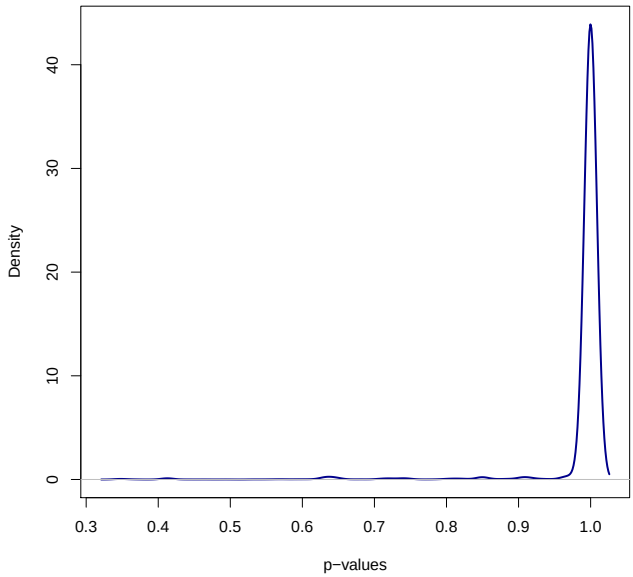


Figure 8.1: *Benjamini & Hochberg adjusted p-values for ANOVA test of AR(1) structure in the unstimulated control sample. For the gastrin stimulated samples X_t and Y_t, the distribution was far more extreme towards one. No negative controls where used as filter and the total number of genes are 8950.*

The analysis for backwards dependencies was done for all genes, before filtering with negative controls, this should however not affect the result (see Section 8.2.3). When

testing for backwards dependencies with an ANOVA test, no genes had a p-value under 0.05 after correction with the Benjamini & Hochberg method (FDR). This was true for both the unstimulated control sample U_t, the X_t series, and the Y_t series.

8.2.2 Observed Intra Class Correlation

The ICC is defined in Chapter 2 (Definitions 2.3 and 2.4) for each gene, and there are different strategies to reduce this to a single statistic, given a collection of genes. One way, is to take the hyperbolic tangent of all the ICC, before summing and transforming back with the hyperbolic arctangent function. This is called the consensus correlation, as it describe the overall intraclass correlation for all the genes. This is given in Table 8.3, where the first column is the calculation using $\tanh$ and $\operatorname{arctanh}$, and the second column is using the `duplicateCorrelation()` function in **R** within the `limma` package. The downside using this procedure is that it can give negative correlation, since the arctangent hyperbolic maps $\mathbb{R}$ onto $[-1, 1]$, however, that should mostly be the case of numerical instability, when the mean ICC is close to zero. This method has a closer relationship to the geometric mean than to the arithmetic mean.

When investigating the random effects, they turned out to be quite small, such that $\sigma_D^2 \ll \sigma_R^2$. This gives a low ICC, indicates that the observations behaved almost as independent random variables. The ICC's had a range from 10^{-14} to 3×10^{-1} in magnitude, but were skewed to the left with a low mean. From Figure 7.2 there was a faster drop in the error (iMSE) for lower ICC, as the number of replicas increased. This suggest that an increase in the number of replicas might be more favourable with lower ICC.

Table 8.2: *Summary statistics for the intraclass correlation for the different series.*

Series	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
U_t	0.000	0.038	0.289	0.308	0.529	0.913	2
X_t	0.000	0.000	0.013	0.121	0.187	0.861	
Y_t	0.000	0.000	0.000	0.011	0.439	0.586	1

The observed ICC is given in Table 8.2. The 3rd quantile in the Y_t sample may not be correct as **R** originally calculated it to be 0. The values are calculated by ordering the estimated ICC for all the genes and then to pick the quantiles, except for the 3rd quantile in Y_t which is calculated by kernel density estimation.

The low ICC does not have any clear biological explanation, but it does mean that there is low correlation within each replica, and that the variance in the error dominates the data. With a high ICC it is easier to discriminate between the replicas since observations from the same clusters tend to lie close the same value. With a low ICC, the observations in the replicas overlap such that it is difficult to tell the difference between the clusters.

Table 8.3: *Consensus correlation for the different time series.*

Series	$\tanh(\cdots)$	Dupcorr
U_t	0.37016	0.36947
X_t	0.07511	0.05051
Y_t	0.00073	-0.05508

8.2.3 Using Negative Control Spots to Filter the Data

If a gene in the Unstimulated control sample have a low expression over the entire time interval, it may have be unsuitable to use it as a correction in the X_t series (see Section 8.3.1). It could therefore be an advantage to remove those samples before analysing for any trend. This removal could be done by filtering the genes with negative control spots.

A negative control spot does not have any binding site in common with the organism studied in the experiment. This gives that in a negative control there should be no expression at all, and if there are, it should be regarded as background noise only. The negative control spots (825 of them) where located on each microarray which means that at each time point, a new negative control was taken. However, we chose to pool all the negative controls together, before deciding on a common cut-off for all the unstimulated control samples. With this approach, only whole time series were removed, and not individual time points, this to ensure that there where enough time points left in the time series to model all the parameters.

The negative controls are used to construct a threshold (cut-off), such that all expressions below that point are regarded as background noise. A common cut-off is the α quantile, for some value of α, which is shown in Figure 8.2, as an illustration of the .90 quantile cut-off. Another strategy is to use the maximum value of the negative controls as a threshold. This is a much stronger requirement than that of the α quantile, and can increase the probability for a type II error, by removing a higher number of time series. In this thesis we chose to use a .99 quantile, this since the threshold needed to be quite high in order to remove around 4% of the unstimulated control time series.

8.3 Frame of Reference

We fitted an LME to the series U_t, X_t, and Y_t separately. The resulting FDR adjusted p-values for any trend in time from the X_t and Y_t series can be found in Figure 8.3.

In Figure 8.3 it is clear that the frame of reference does an have impact on the number of significantly expressed genes. As the figure shows, there seems to be an inflation of the p-values testing for any time effect, when only adjusting against t_0. This since the Y_t se-

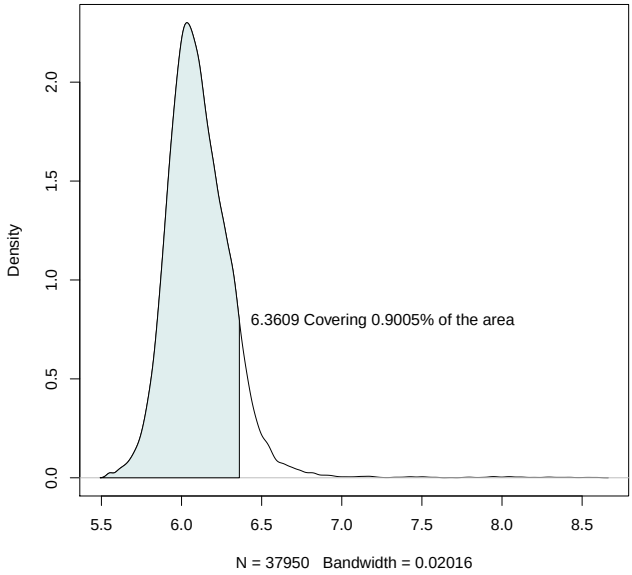


Figure 8.2: *Distribution of expression values of all negative controls over all time points. The shaded area covers approximately 90% of the area, giving a cut-off at expression level* 6.36 *for the* $\alpha = .90$ *quantile. The* $.99$ *quantile had a cut-off at expression level* 6.71*. The number* $N = 37950$ *is the total number of negative control spots, 46 time points times 825 spots.*

ries has a much higher tail heaviness, giving fewer genes that are significantly expressed. This may be attributed to the effect of removing trends in the gastrin stimulated sample, which corresponds to a trend in the unstimulated control sample. If there is a significant increasing trend in the gastrin stimulated sample and also an increase in the unstimulated control, then the time series X_t has a significant increasing trend, but the difference between the gastrin stimulated and the unstimulated control sample could remain constant, with no detectable time effects.

The low ICC in the unstimulated sample, with an even lower ICC in the gastrin stimulated sample, showed that there is little variation between each replica, and they behave as if they were independent. The difference is not large enough to say that it comes from choosing a different frame of reference, but can be interesting in its own right. When the random effects are small enough, it can be argued that an ordinary linear model can be sufficient, since the data behave practically independent (i.e. the observations can still be correlated

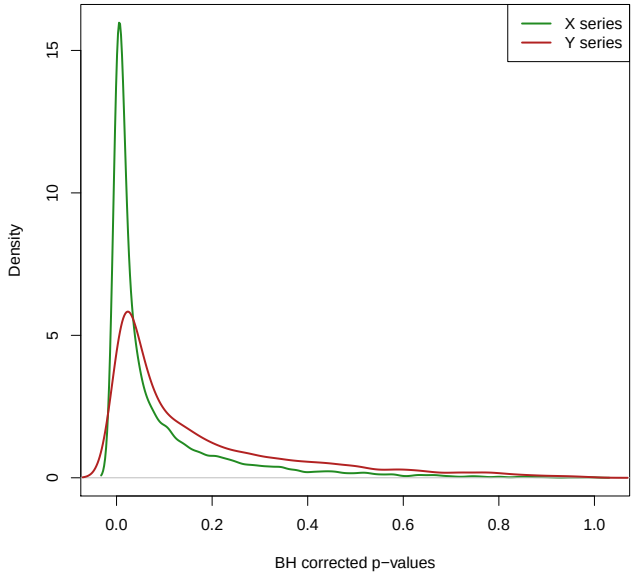


Figure 8.3: *Distribution of Benjamini & Hochberg adjusted p-values when testing for any time effect.*

and dependent, but for an observer they behave independently).

8.3.1 Effect on Taking Differences's

An argument against using Y_t instead of X_t, is that it might lead to increased variance, this however, is only true if the U_t and G_t series are uncorrelated or negatively correlated. This is clearly stated in Equation (8.4), where the covariance is subtracted.

$$\mathrm{Var}(Y_t) = \mathrm{Var}(X_t) + \mathrm{Var}(U_t) - 2\,\mathrm{Cov}(X_t, U_t) \qquad t > 0, \tag{8.4}$$

If the expression profiles between X_t and U_t are highly correlated, the variance of Y_t could actually be lower than the variance in X_t.

When calculating the variances for all the genes, it turned out that X_t had less variance in most cases, with $\sigma^2_{X_t} < \sigma^2_{Y_t}$ in 64% of all genes. However, from Figure 7.1 it may be assumed that the difference is very small.

The small difference could indicate that for some genes there is a large covariance, and for some the expression profile behaves as if X_t and U_t are independent, such that the effect cancels out. A (positive) correlation between the gastrin stimulated sample and the unstimulated control sample, means that if the gastin sample increase over time, then the unstimulated control also increase.

However, a large correlation between X_t and U_t is something that can not be counted on each time this kind of experiment is conducted. However, the variance could be also somewhat controlled by using a negative control. This since the negative controls will remove samples that behaves as background noise, which will cause higher variance in the time series if they are subtracted from the gastin sample, due to low correlation between the unstimulated and the gastrin stimulated time series. However, if there is a negative correlation, this may not be detected using the negative controls and will cause an increase in the variance it Y_t series.

8.3.2 Classification of Expression in Time Series

When analysing the two time series X_t and Y_t, we wish to classify them according to a presence or absence of a time effect. From a theoretical viewpoint, if the underlying true expression profile is known, there are five possible groups based on true underlying expression profiles, listed in Table 8.4.

Table 8.4: *Table showing different underlying groups, where '+' is truly significantly expressed and '–' is not truly significantly expressed.*

Groups of Gene Expression					
Label	a	b	c	d	e
U_t series	+	+	+	–	–
X_t series	+	+	–	+	–
Y_t series	+	–	+	+	–

In group a all the expression profiles are significant, this is a result of either U_t and X_t moving in the opposite directions or one of the time series is growing (decreasing) much faster than the other. Using both X_t and Y_t will draw the same conclusion; that gastrin has some effect on this particular gene. In group d there is a significant trend in the X_t series that is detected using both X_t and Y_t, which indicate that gastin has an effect on this gene. In group e no effect from gastrin is detected in either X_t and Y_t.

We will no focus on the groups where X_t and Y_t give different results. In group b, the gastin stimulated sample has a significant change, but so does the unstimulated control. This could indicate that the observed change in the sample is the result of external factors, and Y_t should be trusted more. In group c, there is a time effect in the unstimulated control

sample, which is detected in the Y_t series. The X_t series have no detectable trend in time, which mean that the gastrin had a cancelling effect on whatever affected the unstimulated control sample. From this setting, it is not clear which series to use for deciding if there is an effect. But from a biological viewpoint, this conflicting behaviour still requires further investigation.

Due to error and randomness, all the 8 (2^3) possible configuration of '+' and '–' can be observed in practise, Table 8.5 shows all the different groups. To avoid confusion with Table 8.4, we have labelled the groups by 1–8, instead of using letters. This since Table 8.4 shows only biological underlying configurations.

Table 8.5: *Table showing the different configuration, where '+' is observed significantly expressed and '–' is not observed significantly expressed.*

Observed groups of Gene Expression								
Label	1	2	3	4	5	6	7	8
U_t	+	+	+	+	−	−	−	−
X_t	+	+	−	−	+	+	−	−
Y_t	+	−	+	−	+	−	+	−

The additional columns in Table 8.5 as compared to Table 8.4 are four, six and seven, none of them have any clear biological interpretation. The most likely reason that the genes have a time effect that is almost significant in U_t or X_t, but the difference is significant, or opposite.

The different explanation for the different columns may vary, but we will present what we think is the most likely reason for the different categories.

Group One: Here both the X_t and Y_t series agree, and this indicate that X_t and U_t either go in opposite direction, or one of the series grow much faster then the other. Gastrin is likely to have an effect on this gene.

Group Two: The U_t and G_t series follows each other, such that the difference cancel out. Gastrin probably has no significant effect on that particular gene.

Group Three: When the change in U_t is somehow cancelled out by the gastrin, this would still be detected as significant change in Y_t series, but not in X_t series. Gastrin has the effect of cancelling the influence that the unstimulated control are subject to.

Group Four: This group has close similarities with group three, that the G_t series probably follows the U_t series, but not enough to be significant. Unclear if gastrin has an effect or not on this gene.

Group Five: Gastrin has a clear significant effect on this gene, that is detected in both X_t and Y_t.

Group Six: The most likely explanation is that gastrin has an effect that is detected in the X_t series, but is cancelled out in Y_t due to noise in the unstimulated control series. This is the case where using an unstimulated control sample clearly increases the variance in the time series.

Group Seven: The G_t and U_t series have an insignificant change in the opposite directions. This result in an significant change being detected in the Y_t series. It is Unclear if gastrin has an effect on this gene, or if the detected effect is a result of noise.

Group Eight: Gastrin does not have any significant effect on this gene.

Table 8.6: *Table showing the proportion of genes that fall into each class, according to Table 8.5. Analysed using 4th order Legendre polynomials. The rows are different filter strategies with negative controls.*

Classification of Genes									
Label	1	2	3	4	5	6	7	8	# Genes
99% quantile filter	21	12	1	3	26	15	3	19	8903
Maximum value filter	21	10	2	6	36	15	3	9	5042
No filtering	18	10	2	8	30	17	3	14	8944

It became clear during the analysis that the choice of cut-off in the negative control had some effect on the classification. This is reflected by the differences in the rows in Table 8.6. Of the groups not present in Table 8.4, only group six have a substantial proportion. This could be due to a significant trend in the stimulated sample that cancelled by random noise. The difference in the proportion in group six could be used as argument for a negative control. This since a negative control would be expected to reduce the proportion of genes that end up in this group, by removing samples that behaves as background noise.

Lower Number of Basis Functions

Using a more restrained set of basis functions by removing the highest polynomial order, the classification changes substantially. This suggest that the classification is strongly dependent on the choice of model. The higher number in group eight is a natural consequence of the decreased flexibility in the model. When the fourth order polynomial is removed, the ability for the model to respond to rapid increase in the observed data is lost. This can reduce the number of false positive, as well as to increase the number of false negatives, due to lack of flexibility. This can be observed in group 8, as the entries are doubled from Table 8.6 to Table 8.7.

Table 8.7: *Table showing the proportion of genes that fall into each class, accordingly to Table 8.5. Analysed using 3rd order Legendre polynomials.*

Classification of genes									
Label	1	2	3	4	5	6	7	8	# Genes
99% quantile filter	10	14	3	9	12	11	3	37	8903
Maximum value filter	9	9	3	15	17	15	4	27	5042
No filtering	9	9	2	17	14	16	3	30	8944

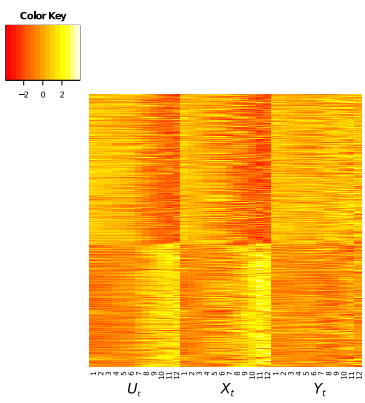
The large differences in Table 8.6 and 8.7 raise some questions, and give rise to concern about the stability of the model. However, there seemed to be some stability in the classification to the groups where X_t and Y_t give different result, as the entries in those groups does not change much.

8.3.3 Clustering

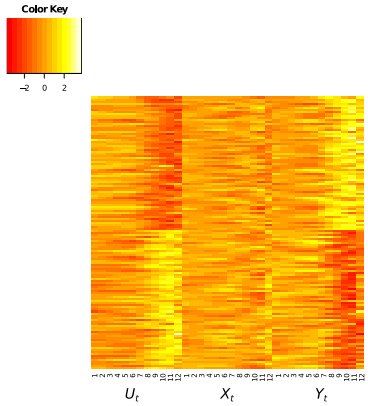
To identify possible clusters of expression profiles, we used the clustering methods outlined in Section 6.2.2. This method could help to identify if there is some clustering within each group. Of special interest is the groups in Table 8.5 where the X_t and Y_t series gives different results, that is, group 2,3,6, and 7. For all the genes in these groups, the optimal number of clusters was 2.

In all the other groups (1,4,5,8), based on all accessible information, we believe that the correct decision is taken regarding the significance of a gene. In the groups where the two reference frames give different results, it is of importance to see if there is some internal structure that can be explain the difference, and be used in order to decide whether to trust the X_t or the Y_t series, this since on of them will give the wrong decision. To increase the reliability of the different frames of reference, clustering could be used to see if there is some ordering within all genes assigned to the respective groups.

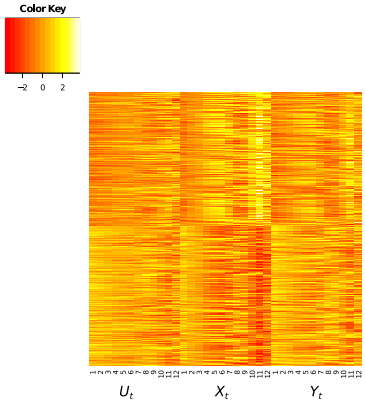
In Figure 8.4a we clearly observed only two different behaviours, both where the U_t and X_t followed each other closely, confirming our suspicion that X_t misclassify the genes as gastrin having an effect. The genes classified in group 3 can be roughly divided into two groups, which is clearly seen in Figure 8.4b. Here the gastrin stimulated genes had no significant change in time, but the unstimulated control had either an increasing trend (first cluster) or an decreasing trend (second cluster). This may be contributed to gastrin having the opposite effect, or that gastrin somehow stabilized these genes. In this group it is unclear if gastrin had an effect on the genes, however, the stability in the X_t series are remarkable compared to the U_t series, which cluster into two distinct clusters. This stability can be used as an argument for gastrin having no effect on those genes. In Figure 8.4c, all the genes clustered roughly into two clusters, one with something similar to



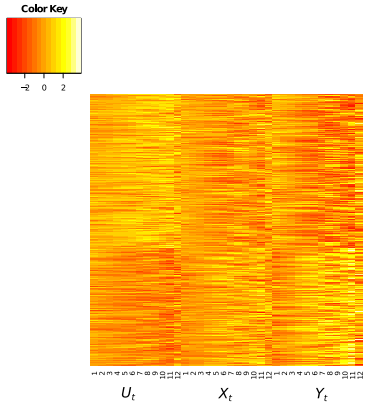
(a) *Group 2*



(b) *Group 3*



(c) *Group 6*



(d) *Group 7*

Figure 8.4: *Heatmap of genes in group 2,3,6, and 7 from Table 8.5 using no filtering with negative control. On the x-axis are the predicted time point for each time series U_t, X_t, and Y_t, and on the y-axis are the genes in the respective groups. Four basis functions plus intercept is used.*

an early response curve, and one with an increasing trend. Both clusters seems to have a significant trend in X_t that is lost in the perturbation noise, and using Y_t is likely to cause misclassification. The genes in group 7, which are plotted in Figure 8.4d separates

into two clusters, in both the trend in U_t and X_t goes in the opposite directions, but it is unclear if this is due to noise or the effect of gastrin treatment. This group does have some similarities with group 3 in that the gastrin treated samples seems more stable than the unstimulated control samples, which also here could be used as an argument that gastrin does not have any effect.

All the figures in Figure 8.4 helps somewhat to identify if an unstimulated control should be used to infer the level of change from gastrin on a particular gene.

8.4 Discussion

By the lack of any AR(1) structure, it can be assumed that there is no detectable higher order AR structure in the time series. This may be a bit surprising, since the gene expressions are measured over time. In previous cases with similar data sets, however, functional data analysis and basis expansion have been shown to be sufficient. And, with only 12 data time points, to investigate for any AR trend is highly optimistic.

The extremely low observed ICC, could make the LME model ineffective compared to a standard linear model. However, with fast computers and stable numerical algorithms, this is no longer any issue. With a data structure that may exhibit any extended covariance structure or clustering, an LME model is generally used. The model's stability and number of basis functions are possible shortcomings in the calculations, as it seems that the model is highly dependent on the choice and size of basis expansion. What may cause this, is that there are differences in the change over time for the different genes. With some genes that are proportional to a second order polynomial, while others have behaviour proportional a fourth order polynomial. What have been previously noted (in Chapter 4), is that setting the basis expansion of constant size, may lead to an inflation of significant genes.

The strong requirements on the cut-off value for the filtering with the negative controls of the unstimulated samples, means that there is a general trend of a high expression in the unstimulated control. With a cut-off of .90 quantile only one gene is removed, and increasing to two genes at the .95 cut-off. With the .99 quantile only 41 genes are removed, and using the maximum value then 3902 genes are removed (43%), which again, could be seemed to much.

The negative control spots were measured at each time point, but we chose to pool all time points before deciding on a cut-off. The negative control spots could also be used to filter out genes at each time point. This may have more difficulties connected to it, since removing only part of the time series, can cause the model to break down. Genes have to be removed in all time points, not in individual time points. It is not clear how much effect the two different strategies have on the classification from Table 8.5.

8.4.1 A p-value Approach

Throughout this chapter, p-values have been used to indicate fit of the model. This is convenient, due to their mathematical properties such as well ordering, that they are bounded by $[0, 1]$, and that they represents a probability. This can be especially useful when the number of genes is so large that each gene cannot be investigated itself, but one can say something about the collection of all the genes using p-values. If the distribution of the p-values stays the same for two or more different models, it may be reasonable to assume that the model utilizes approximately the same amount of information, and the decision based on the different models will be the same.

When investigating multiple p-values, the problem of multiple testing arises. In this thesis, we chose to use the Benjamini & Hochberg method to control the False Discovery Rate (FDR), although other adjustment strategies could have been used, but this was not the focus in this thesis.

This approach gives clear indication in Figure 8.3 that the use of an unstimulated control is to prefer. But, when analysing the classification in Table 8.6 and 8.7, it is not equally clear. Only observing Figure 8.3, it could be assumed that the proportions in group 2 in Table 8.6 and 8.7 should be higher.

Overall, the change in the distribution of p-values (removing inflated p-values), the removal of any backwards dependencies and the reduction in ICC makes a strong case for the use of an unstimulated control sample. Negative controls also seems to have some benefits, this by removing unstimulated samples that are indistinguishable from noise. The downsides are the misclassified ones, as the genes that comes in group 6 and 7. When the number of genes classified in these groups outnumber the ones classified correctly only when using an unstimulated control, then it is more questionable to use an unstimulated control. However, to have an unstimulated control sample available will give the researcher much more control when determining the effect of any treatment over a time interval.

8.5 Bibliographic Note

The R code used in to process and unpack the genetic information from the microarray chip, was the `lumi` (Du et al., 2008) package. In the actual modelling, both `limma` (Smyth, 2004, 2005; Smyth et al., 2005) and `nlme` (Pinheiro et al., 2011) was used. For clustering the genes from the different groups, we used the `pam()` function, implemented in the `cluster` packages by Maechler et al. (2002). The consensus correlation method was proposed by Smyth et al. (2005), and is also implemented in the limma package in **R**.

9

DISCUSSION & CONCLUDING REMARKS

As the work of this thesis unfolded a number of decisions were taken, regarding the model framework and approaches. In retrospect, some of these decisions could have been done differently, and would possible resulted in different conclusions. This chapter discusses some of the issues with the approaches that were taken.

9.1 Discussion

In this master's thesis we have investigated aspects of experimental design for time continuous microarray gene expression profiles, using the Linear Mixed Effects framework. We have in particular looked at the number of biological replicas used, and the effects of changing the frame of reference. With the biological replications, we have analysed what impact the number of replicates had on the precision of the estimated expression profile.

Key issues in the modelling are the level of smoothing and the scaling of the estimated curve. The scaling down to the interval $[-1, 1]$ put some constraints on the estimated curve, but it also ensured that the polynomial does behave nicely on the boundary. Since the internal distances are preserved, no information should be lost in the scaling. What also was pointed out in Chapter 8 is that the number of basis function does matter, and gives a large degree of difference in the classification when the basis is changed. This dependency on the Taylor complexity and smoothing is somewhat problematic, and should be developed further. As a workaround to this problem, it has been suggested to use a variable number of basis functions. This however creates a new problem of developing a suitable rule for deciding the number of basis functions for each gene, but some form of ANOVA testing could solve this problem. As it has been pointed out in Chapter 4, a fixed number of basis functions for all genes is likely to induce inflated p-values by allowing too much flexibility for many genes.

The low number of biological replicas is also a concern. In Figure 7.2 we observed that the precision could be largely improved by adding multiple replicas, even at the expense of a shorter time series (Figure 7.6). Having more replicas could also possible be used in some way to estimate some form of variance components, and giving a more reliable error estimate. This should result in higher precision in the coefficients and thus higher precision in

the estimated expression profile. The high number of genes that had no significant change could work as a variance stabilizer as they are likely to have little variance, resulting in an underestimate of the overall variance. This could maybe also explanation some of the surprisingly low intra class correlation, along with the genes where the observation that the replication tends to agree, i.e. they lie close to each other.

Clustering of the genes have only been outlined in this thesis and are not thoroughly reviewed, but does seem to help with understanding the classification of the genes.

An issue that has not here been properly addressed is the model diagnostics for the LME model. The nature of the diagnostics methods, as they often rely heavily on visual inspection, are not feasible to do for each gene. This results in that heteroscedasticity cannot be checked for each gene, which is among the key issue in the modelling. The large number of genes also gives that we cannot evaluate each gene separately, but have to analyse quantities for each gene simultaneously. Still, comparing the observed variance in Figure 7.1 with the estimated observed discrete $i\widehat{\mathrm{MSE}}$ (which is close to the sum of squares) in Figure 7.4, the model seems to fit well for most genes.

Another issue is the amount of genes where the use of an unstimulated control sample gives the wrong decisions (mainly group 6 and 7 from Table 8.5). If the number of genes where the correct decision is taken using the unstimulated control only slightly increases, while the ones were the wrong decision is taken using the unstimulated control is large, then this approach should be reviewed in a cost/benefit setting. However, with an unstimulated control sample in place, it is not required to use it in drawing inference about the effect of the treatment. This in total gives the researcher much more control over the decisions taken for each gene. But in general, the series where the unstimulated controls are subtracted should not in general be trusted more then those series which are adjusted for by the unstimulated controls.

9.2 Further Work & Possible Extension

To build upon this thesis, there are a number of ways to extend and develop models and framework for the analysis of time continuous microarray expression profiles.

Possible extensions that of the estimation is to use a smoothing spline, that would eliminate the choice of the number of basis functions, but introduce an new problem in form of the roughness penalty applied to each gene. Testing significance is also more difficult for smoothing splines then for regular linear basis expansions.

Another clear extension is the use of paired samples, that is, to integrate both time series in the same model. This could give classification with higher precision, and better understanding of the relationship between the unstimulated sample and the gastrin response.

With so few biological replicas, it may be suitable to use some for of Bayesian approach. This would include setting priors on the fixed and random effects, among others. The Bayesian framework opens up a whole new set of possibilities, with many new tools and settings. Some Bayesian techniques have already been well integrated with LME, among them the INLA platform.

It might also be worth looking into different norms and performance tests. This thesis defined and used the iMSE (see Chapter 5), but other way of defining bias, and precision could be considered. The theory behind iMSE is not fully developed, and its behaviour could be investigated more.

More complicated simulations experiments could also preformed, where one take into account different behaviours of the genes. This includes dependencies between the genes, difference in variance and a different filtering strategies.

It has also been suggested from a biological viewpoint, to use basis functions that have some biological meaning. This will often remove nice properties from the basis, such as orthogonality, but can give coefficients that are much easier to interoperate. This basis functions are constructed a priori, based on biological meaningful processes and behaviours.

9.3 Concluding Remarks

We have seen that over all, the LME framework can be well suited to estimate time continuous gene expression profiles and to draw conclusions about the behaviour of genes subjected to some treatment. There seems to be an increase in precision of the estimated expression profile if the number of replications are increased, even at the cost of decreasing the number of time points in each replication. However, each replication should have a sufficient amount of time points in order to capture the behaviour of the genes under a given treatment.

Using an unstimulated control sample will give more control in inferring the effect of some treatment over time interval in some cell culture, but should not be used without caution. We recommend to asses both the time series only adjusted for the starting point and the time series adjusted for the unstimulated control before drawing a conclusion about the treatment.

This concludes with analysing the unstimulated sample and if there is some detectable trend in a particular gene, then this should be adjusted for. However, if the unstimulated control sample for a gene behaves as noise, then adjusting can do more harm then good.

BIBLIOGRAPHY

Brazma A., Hingamp P., Quackenbush J., Sherlock G., Spellman P., Stoeckert C., Aach J., Ansorge W., Ball C. A., Causton H. C., Gaasterland T., Glenisson P., Holstege F. C., Kim I. F., Markowitz V., Matese J. C., Parkinson H., Robinson A., Sarkans U., Schulze-Kremer S., Stewart J., Taylor R., Vilo J., and Vingron M. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nature Genetics*, 29:365 – 371, December 2001.

Ioannis P. Androulakis, Eric Yang, and Richard R. Almon. Functional assessment of time course microarray data. *Annual Review of Biomedical Engineering*, 9:205 — 228, 2007.

Claudia Angelini, Luisa Cutillo, Daniela D. Canditiis, Margherita Mutarelli, and Marianna Pensky. BATS: a Bayesian user-friendly software for Analyzing Time Series microarray experiments. *BMC Bioinformatics*, 9(415), October 2008.

Alvis Brazma, Helen Parkinson, Ugis Sarkans, Mohammadreza Shojatalab, Jaak Vilo, Niran Abeygunawardena, Ele Holloway, Misha Kapushesky, Patrick Kemmeren, Gonzalo Garcia Lara, Ahmet Oezcimen, Philippe Rocca-Serra, and Susanna-Assunta Sansone. Arrayexpress – a public repository for microarray gene expression data at the EBI. *Nucleic Acids Research*, 21(1):68 – 71, 2003.

Michael D Burkitt, Andrea Varro, and D Mark Pritchard. Importance of gastrin in the pathogenesis and treatment of gastric tumors. *World Journal of Gastroenterology*, 15 (1):1 – 16, January 2009.

Chao Cheng, Xiaotu Ma, Xitin Yan, Fenghu Sun, and Lei Li. MARD: a new method to detect differential gene expression in treatment-control time courses. *Bioinformatics*, 22(21):2650–2657, August 2006.

Norma Coffey and John Hinde. Analyzing Time-Course Microarray Data Using Functional Data Analysis - A Review. *Statistical Applications in Genetics and Molecular Biology*, 10, May 2011.

James Curran. *Bolstad: Bolstad functions*, 2011. URL http://CRAN.R-project.org/package=Bolstad. R package version 0.2-21.

Pan Du, Warren A. Kibbe, and Simon M. Lin. lumi: a pipeline for processing Illumina microarray. *Bioinformatics*, 24(13), July 2008.

Peter J. Green and Bernard W. Silverman. *Nonparametric Regression and Generalized Linear Models*. Monographs on Statistics & Applied Probability. Chapman & Hall, 1994.

David Harville. Extension of the Gauss-Markov Theorem to Include the Estimation of Random Effects. *The Annals of Statistics*, 4(2):384 – 395, 1976.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning; Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, New York, USA, 2nd edition, February 2009.

Jiming Jiang. Wald Consistency and the Method of Sieves in REML Estimation. *The Annals of Statistics*, 25(4), 1997.

Alan F. Karr. *Probability*. Springer Text in Statistics. Springer-Verlag, New York, USA, 1st edition, 1992.

David Kincaid and Ward Cheney. *Numerical Analysis, Mathematics of Scientific Computing*. American Mathematical Society, Providence, Rhode Island, 3rd edition, 2002.

Ervin Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley & Sons. Inc., USA, 1978.

Ping Ma, Wenuan Zhong, and Jun S. Liu. Identifying Differentially Expressed Gene in Time Course Microarray Data. *Statistics in Bioscience*, 1(2):144–159, 2009.

Martin Maechler, Peter Rousseeuw, Anja Struyf, Mia Hubert, and Kurt Hornik. *cluster: Cluster Analysis Basics and Extensions*, 2002. R package version 1.14.2 — For new features, see the 'Changelog' file (in the package source).

John N. McDonald and Neil A. Weiss. *A Course in Real Analysis*. Academic press, 1st edition, January 1999.

María Nueda, Patricia Sebastián, Sonia Tarazona, Francisco García-García, Joaquín Dopazo, Alberto Ferrer, and Ana Conesa. Functional assessment of time course microarray data. *BMC Bioinformatics*, 10, June 2009.

Jose Pinheiro, Douglas Bates, Saikat DebRoy, Deepayan Sarkar, and R Development Core Team. *nlme: Linear and Nonlinear Mixed Effects Models*, 2011. R package version 3.1-102.

José C. Pinheiro and Douglas M. Bates. *Mixed-Effects models in S and S-PLUS*. Statistics and Computing. Springer, 1st edition, 2000. ISBN 0-387-98957-9.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL http://www.R-project.org/. ISBN 3-900051-07-0.

Jim O. Ramsay and Bernard W. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer, 2nd edition, 2005.

Oliver Schabenberger. Mixed Model Influence Diagnostics. *SUGI 29 Proceedings*, 2004. Paper 189 – 29.

Mohak Shah and Jacques Corbeil. A General Framework for Analyzing Data from Two Short Time-Series Microarray Experiments. *IEEE/ACM Transactions on computational biology and bioinformatics*, 8(1):14–26, February 2011.

Gordon K. Smyth. Limma: linear models for microarray data. In R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, and W. Huber, editors, *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pages 397–420. Springer, New York, 2005.

Gordon K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004. Article3.

Gordon K. Smyth, Joelle Michaud, and Hamish S. Scott. Use of within-array replicate spots for assessing differential expression in microarray experiments. *Bioinformatics*, 21(9):2067–2075, May 2005.

John D. Storey, Wenzhong Xiao, Jeffrey T. Leek, Ronald G. Tompkins, and Ronald W. Davis. Significance analysis of time course microarray experiments. *PNAS*, 102(36): 12837 – 12842, September 2005.

W. A. Thompson. The Problem of Negative Estimates of Variance Components. *The Annals of Mathematical Statistics*, 33(1):273 – 289, 1962.

Gregory R. Warnes. *gmodels: Various R programming tools for model fitting*, 2011a. URL http://CRAN.R-project.org/package=gmodels. R package version 2.15.1, Contributions from Ben Bolker and Thomas Lumley and Randall C Johnson and Randall C. Johnson.

Gregory R. Warnes. *gplots: Various R programming tools for plotting data*, 2011b. URL http://CRAN.R-project.org/package=gplots. With contributed by: Ben Bolker and Lodewijk Bonebakker and Robert Gentleman and Wolfgang Huber Andy Liaw and Thomas Lumley and Martin Maechler and Arni Magnusson and Steffen Moeller and Marc Schwartz and Bill Venables.

Susan A. Watson, Anna M. Grabowska, Mohamad El-Zaatari, and Arjun Takhar. Gastrin – active participant or bystander in gastric carcinogenesis. *Nature Reviews Cancer*, 6 (12):936 – 946, December 2006.

Zhi Wei and Hong Zhe Li. A Hidden Spatial-temporal Markov Random Field Model for Network-based Analysis of Time Course Gene Expression Data. *Annals of Applied Statistics*, 2:408 – 429, February 2011.

Brady T. West, Kathleen B. Welch, and Andrzej T. Galecki. *Linear Mixed Models: A Practical Guide Using Statistical Software*. Chapman & Hall /CRC, November 2006.

Nicholas Young. *An introduction to Hilbert space*. Cambridge University Press, Cambridge, UK, 1st edition, 1998.

Hao Yu. Rmpi: Parallel Statistical Computing in R. *R News*, 2(2):10–14, June 2002.

Eirin Tangen Østgård. Statistical modeling and analysis of repeated measures, using the linear mixed effects model. Master's thesis, Norwegian University of Science and Technology, Trondheim, June 2011.

A

APPENDIX

Notation

Note on Notation

All vectors are written in bold, and assumed to be column vector in not stated otherwise. Error terms are usually written as an ε and follows the Normal distribution $\mathcal{N}(0, \sigma^2)$, unless other distribution is specified.

A observation is denoted as a y, whereas the underlying function generating the observation is written as $f(t)$ or $y(x)$. The $f(t)$ might not always have a variable assigned to it, and only denoted as f, however in those cases there should be made clear that it is a continuous function, and not a single value i.e. an observation.

Tables of Symbols

Table A.1: *Distributions and its Notation*

Distribution	Notation	Parametes
Standard Normal	$\mathcal{Z}$	None
Normal Distribution	$\mathcal{N}(\mu, \sigma^2)$	$\mu \in \mathbb{R}, \sigma^2 > 0$
Multi variate normal	$\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	$\boldsymbol{\mu} \in \mathbb{R}^p, \boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$
Chi-square distribution	χ^2_{df}	$df \in \mathbb{N}$
Fisher Distribution	F_{df_1, df_2}	$df_1, df_2 \in \mathbb{N}$

Table A.2: *Quantiles*

Quantile Notation	Distribution	Parameters
$t_{df,\alpha}$	The student t-distribution	α, df

Table A.3: *Math Symbols*

Symbol	Usage	Page
$\overset{\text{d}}{=}$	Equality in distribution	6
$\overset{\text{def}}{=}$	Equal by definition	31
$\mathbf{1}_n$	n dimensional one vector	
$\xrightarrow{\text{a.s.}}$	Almost surely convergence	33

Table A.4: *Latin Symbols*

Symbol	Usage	Page
$\mathbb{P}(A)$	Probability measure	34
$\mathbb{E}[X]$	Expected value of X	7
$\mathbb{I}(A)$	Indicator function for an event/set A	
$\mathbf{I}_n$	$n \times n$ identity matrix	6
$\det(M)$	Determinant of a matrix M	10
$\text{block}(M_i)$	Diagonal block construction of matrices	8
$\text{diag}(M)$	Vector with diagonal elements of a matrix M	9
$\text{rank}(M)$	The rank of a matrix M	13
$\text{se}(\hat{\theta})$	Standard Error of a estimator $\hat{\theta}$	13
$\text{Var}(X)$	Variance of a random variable	12
$\text{Cov}(X, Y)$	Covariance of two random variable	7
$\text{Corr}(X, Y)$	Correlation of two random variables	8

Table A.5: *Script Symbols*

Symbol	Usage	Page
$\mathcal{D}$	Discreet differential operator	27
$\mathcal{H}$	The space spanned by the basis functions	22
$\mathcal{L}^p$	L-p spaces	18
$\mathcal{P}^n$	The set of all polynomials up to order n	19
$\mathcal{O}$	Big O notation	22

Table A.6: *Greek Symbols*

Symbol	Usage	Page
ρ	Intra Class Correlation (ICC)	9
δ_{nm}	Kronecker delta	19
Δ_i	Partition i (Step size)	19
Λ	Eigenvalue decomposition	9

B

APPENDIX

List of Abbreviations

Table B.1: *A list of all abbreviations occurring in this thesis.*

Symbole	Usage	Page
ACF	Auto Correlation Function	29
ANOVA	Analysis of Variance	5
AR	Autoregressive Process	38
EM	Expected Maximization Algorithm	11
FDA	Functional Data Analysis	27
FDR	False Discovery Rate	53
ICC	Intra Class Correlation	9
iid	Independent and identically distributed	5
IKM	Department of Cancer Research and Molecular Medicine at NTNU	i
iMSE	Integrated Mean Squared Error	34
IMF	Department of Mathematics at NTNU	i
INLA	Integrated Nested Laplace Approximation	66
LME	Linear Mixed Effects models	5
MARD	Mean Absolute Rank Difference	28
MPI	Message Passing Interface	41
mRNA	Messenger Ribonucleic acid, transcript of DNA	1
MSE	Mean Squared Error	31
NCS	Natural Cubic Splines	17
NTNU	Norewegian University of Science and Technology	i
PACF	Partial Auto Correlation Function	29
PAM	Partitioning Around Medoids	38
REML	Restricted Maximum Likelihood	11
RKHS	Reproducing Kernels Hilbert Space	28

C

APPENDIX

Algorithms for Monte Carlo

Algorithm 1 Simulating Data, not in parallel

Require: library; nlme,
Set constants $\#_{\text{MC}}, tp[\,]$(time points), $\sigma_R, order$(of basis), $\#_{\text{clusters}}, grid, \rho$, and generator function f
Reserve space for the result vector $res[\,]$
$\sigma_D = \sqrt{\frac{\rho\sigma_R^2}{1-\rho}}$
$basis[tp[\,], order] = \texttt{legendre}(tp, order)$
for $j = 1$ to N_{MC} **do**
Generate error $\varepsilon[\,] \overset{\text{d}}{=} \mathcal{N}_{tp\times\#_{\text{clusters}}}(\mathbf{0}, \sigma\mathbf{I})$
Generate cluster offset $u[\,] \overset{\text{d}}{=} \mathcal{N}_{\#_{\text{clusters}}}(\mathbf{0}, \sigma\mathbf{I})$
Generate observations $y[tp[\,], \#_{\text{clusters}}] = f(tp[\,]) + \varepsilon + u[\,]$
for $k = \#_{\text{clusters}}$ to 1 **do**
Estimate LME; $\hat{f}_k^{(j)} = \texttt{lme}(y[, 1:k], basis, tag_{\text{cluster}})$
Ensure: $\exists \hat{f}_k^{(j)}$ {(convergence in lme)}
$D = \left(\hat{f}_k^{(j)}(t) - f(t)\right)^2$
$res[j,k] = \int_{grid} \Delta^2 \,\mathrm{d}\, t$ using numerical integration
end for
end for
return success

Algorithm 2 Parallel Simulation and evaluation of Data

Require: library; nlme, Rmpi
Set number of CPU cores $\#_{\text{CPU}}$
Set constants $\#_{\text{MC}}, tp[\,]$(time points), $\sigma_R, order$(of basis), $\#_{\text{clusters}}, grid, \rho$, and generator function f
Reserve space for the result vector $res[\,]$, set (number of) $tasks[\,] = 1, 2, \ldots, \#_{\text{MC}}$
$\sigma_D = \sqrt{\frac{\rho\sigma_R^2}{1-\rho}}$
$basis[tp[\,], order] = \texttt{legendre}(tp, order)$
Initiate $\#_{\text{CPU}}$ cores, and push all object to all CPUs, set $k = 1$
while $\dim((tasks) > 1$ **do**
Send following task to a free CPU;
Generate error $\varepsilon[\,] \overset{\text{d}}{=} \mathcal{N}_{tp\times\#_{\text{clusters}}}(\mathbf{0}, \sigma\mathbf{I})$
Generate cluster offset $u[\,] \overset{\text{d}}{=} \mathcal{N}_{\#_{\text{clusters}}}(\mathbf{0}, \sigma\mathbf{I})$
Generate observations $y[tp[\,], \#_{\text{clusters}}] = f(tp[\,]) + \varepsilon + u[\,]$
for $k = \#_{\text{clusters}}$ to 1 **do**
Estimate LME; $\hat{f}_k^{(j)} = \texttt{lme}(y[, 1:k], basis, tag_{\text{cluster}})$
Ensure: $\exists \hat{f}_k^{(j)}$ {(convergence in lme)}
$D = \left(\hat{f}_k^{(j)}(t) - f(t)\right)^2$
$res[j,k] = \int_{grid} D^2 \,\mathrm{d}\, t$ using numerical integration
end for
$k = k + 1$
$tasks[\,] = k, k+1, \ldots, \#_{\text{MC}}$
return CPU awaiting new task
end while
return success

Algorithm 3 $\widehat{i\mathrm{MSE}}$ estimation

Require: library; Bolstad
  Input; $\hat{\boldsymbol{\beta}}[\,], f[\,], grid[\,]$
  $\hat{f}[\,] = 0$
  for $i = 0$ to $\dim(beta)$ **do**
    $\hat{f}[\,] = \hat{f}[\,] + \hat{\boldsymbol{\beta}}[i] \times \texttt{legendre}(grid, i)$
  end for
  if $grid[h] - grid[h+1]$ is constant $\forall h$ **and** $\dim(grid)$ is even **then**
    $D[\,] = \left(\hat{f}[\,] - f[\,]\right)^2 \qquad \forall grid[\,]$
    $I = \texttt{sintegral}(grid[\,], D,)$
    return I
  else
    $tmp.l = \sum_{i=1}^{\dim(grid)-1} \left(\hat{f}[i] - f[i]\right)^2 \times (grid[i+1] - grid[i])$
    $temp.r = \sum_{i=2}^{\dim(grid)} \left(\hat{f}[i] - f[i]\right)^2 \times (grid[i] - grid[i-1])$
    $I = \dfrac{tmp.l + temp.r}{2}$
    return I
  end if

Algorithm 4 Basis Construction with Legendre Polynomials

  Input; $grid[\,], order = n, ortho = \mathbf{false}$
  $tmp = 0$
  $legendre[grid, n] = 0$
  for $k = 0$ to $n - 1$ **do**
    $tmp = tmp + (-1)^k \binom{n}{k}^2 \left(\dfrac{1 + grid[\,]}{2}\right)^{n-k} \left(\dfrac{1 - grid[\,]}{2}\right)^k$
    $legendre[, k+1] = tmp$
  end for
  if $ortho$ is **true then**
    for $k = 1$ to n **do**
      $norm = \sqrt{\dfrac{2k+1}{2}}$
      $legendre[, k] = legendre[, k] \times norm$
    end for
  end if
  return $legendre[\,]$

D

APPENDIX

R Code

This appendix contains some of the R code and functions used, to make a replication of the computer experiment easer.

Estimation of Gene Expression with LME

```
thisy <- as.double(geneData[i,])
try(thislme <- lme(thisy~lbasis,random=~1|biol,method="REML",na.action=na.omit),
                  silent=TRUE)
if(exists("thislme"))
{
        anovaP[i] <- anova(thislme)$"p-value"[2]
        ICC[i] <- ICC(thislme)
        sigma[i] <- thislme$sigma
        UnStim_T0.randomEffect[i,] <- thislme$coefficients$random$biol
        try(tmp <- update(thislme,correlation=corCAR1(value=.2,form=~1,
                        fixed=FALSE)), silent=TRUE)
        if(exists("tmp"))
             AnovaP.AR[i] <- anova(tmp,thislme)$"p-value"[2];rm(tmp)
        rm(thislme)
}
```

Parallelization in R Using Rmpi

```
task <- 1:SIMULATIONS
mpi.spawn.Rslaves(nslaves=NR_SLAVES)
estimate.parallel <- mpi.applyLB(tasks, fitLME.replica, Data.obj=Data.obj,
                        MonteCarlo.obj=MonteCarlo.obj, Result.obj=Result.obj)
down <- mpi.close.Rslaves()
```

E

APPENDIX

List of R Packages Used

The list presented here contains all essential R packages used, along with the reference and a short description. The method column, refers to additional methods that have later been implemented, and requires additional references.

Table E.1: *A complete list of R-packages used, be aware that dependencies is not covered. Some of the packages depend on other packages not listed here.*

Name	Method	Description	Author
nlme	lme	Linear and non-linear mixed effects	Pinheiro et al. (2011)
Rmpi	mpi.applyLB	Parallel computing (MPI)	Yu (2002)
Bolstad	sintegral	Numerical integration	Curran (2011)
lumi	lumiR	Import microarray data	Du et al. (2008)
limma		Linear models for microarray data	Smyth (2005)
limma	dupcorr		Smyth et al. (2005)
limma	lmFit, eBayes		Smyth (2004)
gmodels	estimable	Extends LME	Warnes (2011a)
gplots	heatmap.2	Extensions of Plotting in R	Warnes (2011b)
cluster	pam	Generalized K-means	Maechler et al. (2002)

F

APPENDIX

Linear Combinations of Legendre Polynomials

In this thesis, we have assumed that if one of the coefficients is different from zero, then the polynomial is different from zero, in at least one point. This assumption assumes that there is no linear combinations of the Legendre polynomial that gives a zero polynomial, if the norm of the linear combination is different from zero.

Theorem F.1. *Let $\mathcal{H}_j$ be the set of Legendre polynomial op to order j. Then, there exists no finite linear combination $\boldsymbol{\beta}^T\mathcal{H}_j = f(x)$, for any $\boldsymbol{\beta}$, with $\|\boldsymbol{\beta}\| > 0$, such that $f(x) \equiv 0\,\forall x$.*

Proof. **Outline of the proof:** We prove this by induction, by showing that it works for one basis function, then for two, and so on.

If there is only one basis function $h_0 = a_0x^0$, then this evidently hold, since $|\beta| > 0$ and $|a_0| > 0$ by construction. If there are two basis function $h_q = a_0x^q$ and $h_2a_1x^0 + b_1x^1$, then since $\|\boldsymbol{\beta}\| > 0$, only one of the beta's can be zero, but not both. If one is zero, we are back to the case of only one basis function. If both betas are non-zero, we can rearrange the terms, such that

$$\boldsymbol{\beta}^T\mathcal{H} = x^0\underbrace{(\beta_0a_0 + \beta_1a_1)}_{\phi} + \beta_1b_1x^1.$$

Now, the ϕ constant can be zero, but since both the beta's are assumed to be non-zero, $\beta_1b_1x^1$ is not a zero polynomial. This argument can then be extended for increasing order of basis functions.

Q.E.D. □

This ensures that if at least one coefficient is different from zero, the polynomial will also be different from zero in at least one point, for any linear combination with norm bigger then zero.

Uniqueness

The Legendre polynomial are the only infinite series of polynomials that have increasing order and are orthogonal on the $\mathcal{L}^2[-1,1]$ inner product. A proof is somewhat technical and difficult, and will not be given her, it can however be found in A Uniqueness Theorem for the Legendre and Hermite Polynomials by K. P. Williams in *Transactions of the American Mathematical Society* Vol. 26, No. 4, Oct., 1924.

An alternative approach can be to show that the Gram-Smith orthogonalization gives the Legendre polynomials. Since the Gram-Smith procedure is unique, the Legendre polynomials will then be unique.

The Legendre Polynomials

Here follows the five first Legendre polynomials. The polynomials are not normalized but this could be obtained by dividing each polynomial on the normalizing constants given in Equation 3.6.

Order	$l(x)$
1	1
2	x
3	$\frac{1}{2}(3x^2-1)$
4	$\frac{1}{2}(5x^3-3x)$
5	$\frac{1}{8}(35x^4-30x^2+3)$