Imperial College of Science, Technology and Medicine
Faculty of Natural Sciences
Division of Molecular Biosciences

Bayesian Nonparametric Approaches to Modelling Dependencies in Systems Biology

Justina Žurauskienė

A thesis submitted for the degree of
Doctor of Philosophy of Imperial College London, July 2014

©2014 Justina Žurauskienė
Typeset in Times by LaTeX

This report is the result of my own work.

No part of this dissertation has already been, or is currently being submitted by the author for any other degree or diploma or other qualification.

This dissertation does not exceed 100,000 words, excluding appendices, bibliography, footnotes, tables and equations. It does not contain more than 150 figures.

This work is supported by *The Leverhulme Trust* grant and completed in the Division of Molecular Biosciences at Imperial College, London.

All trademarks used in this dissertation are acknowledged to be the property of their respective owners.

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Abstract

All living organisms exhibit complex behaviour, and this is a result of the underlying regulatory mechanisms that occur at cellular and molecular levels. For this reason such reactions are of central importance in the field of systems biology. Throughout this thesis we are concerned with mathematical models that allow us to better understand and represent the biological phenomena behind experimental data, and equally to make predictions about key regulatory processes happening in the cells. Specifically, this work explores and demonstrates how modern Bayesian nonparametric techniques, namely Gaussian process regression and Dirichlet process mixture models, can be applied in order to model complex systems biology data.

Here we have developed a new technique based on Gaussian process regression approaches to model metabolic regulatory processes at the cellular level. Our technique allows us to model noisy metabolite time course data and predicts dynamical metabolic flux behaviour in the associated pathways; we demonstrate that by learning the dependencies between several metabolites we can strengthen our predictions in sparsely sampled regions. We furthermore discuss when Gaussian processes can accurately reconstruct the underlying functions and when they are subject to the Nyquist limit.

Next we proceed to modelling biological processes that occur at the molecular level. Here we are interested in studying large and diverse functional genomics datasets. A variety of computational techniques allow us to analyse such data and model biological processes underlying them; an important class of these methods are techniques that permit the detection of heterogeneity in experimentally observed data. Here we employ Dirichlet processes to estimate the number of clusters within such genomic datasets and further propose a new method to tackle the data fusion problem. Our technique primarily relies on the outcomes from nonparametric Bayesian clustering approaches and is based on graph theory concepts, but in parallel we also discuss and show how this graph-theoretical approach can be extended to integrate results from non-Bayesian type clustering algorithms. We show that by integrating several data types we can successfully identify e.g. sets of genes that are regulated by similar

transcription factors.

Acknowledgements

First and foremost I would like to express a sincere gratitude to my supervisor, Prof. Michael Stumpf, for taking me and for providing the opportunity to complete my PhD in an excellent research environment. I am grateful for all his support and encouragement.

Next, I would like to thank Dr. Paul Kirk for his advising, I have learnt a great deal from his insights about Gaussian process regression. As well, thank you for working with me on genomic data fusion.

Also, I thank in a very special way Dr. John Pinney for always being there to give all the support and very helpful guidance.

Equally, I would like to acknowledge all the Theoretical Systems Biology group members. You all provided a wonderful research environment and helped me throughout these PhD years.

Finally, this work would not be possible without the support from my husband Mantas. Thank you for all the strengths you gave me to reach this peak!

Abbreviations

αKG	α ketoglutarate
ΔglnG	Isogenic glnG deletion
ARANDI	Adjusted RAND index
BIG	Bayesian information criterion
BCC	Bayesian consensus clustering
BPPI	Binary protein-protein interaction
CPR	Chinese restaurant process
DNA	Deoxyribonucleic acid
DP	Dirichlet process
DPM	Dirichlet process mixture
EM	Expectation-maximisation
FBA	Flux balance analysis
FT	Fourier transform
GDH	Glutamate dehydrogenase
GLN	Glutamine
GLU	Glutamate
GP	Gaussian process
GPR	Gaussian process regression
GS	Glutamine synthetase
GS-GOGAT	Glutamine synthetase-glutamate synthase
i.i.d.	Independent and identically distributed
IGMM	Infinite Gaussian mixture model
LLN	Law of Large Number
MAP	Maximum a posteriori estimate
*m*RNA	Messenger ribonucleic acid
*mi*RNA	Micro *m*RNA
MCMC	Markov chain Monte Carlo
MDI	Multiple data integration
MGP	Multiple-output Gaussian process

NMR	Nuclear magnetic resonance
ODE	Ordinary differential equation
PM	Polymyositis
PPI	Protein-protein interaction
SMC	Sequential Monte Carlo
sIBM	Sporadic inclusion body myositis
TCA	Citrate cycle
TF	Transcription factor
WT	Wild type

Contents

Abstract **i**

Acknowledgements **iv**

Abbreviations **v**

1 Introduction **1**

1.1 Overview of modelling strategies 1

1.2 Motivation and objectives of this work 4

1.3 Thesis outline 5

1.4 List of publications 7

2 Theoretical Background **8**

2.1 Introduction 8

2.2 Markov chain Monte Carlo 9

2.3 Finite Gaussian mixture models 14

2.4 Dirichlet processes 19

2.5 Infinite Gaussian mixture models 22

2.6 Gaussian process regression 28

2.7 Multiple-output Gaussian process regression 31

3 Modelling Gene Expression and Imputing the Missing Data with Multiple-output Gaussian Processes 35

3.1 Introduction 35

3.2 Applications 37

3.2.1 Standard model for gene expression 37

3.2.2 Imputing the missing data in *p53–Mdm2* 40

3.2.3 Applications to *Hes1* regulatory system 43

3.3 Conclusions and Discussion 46

4 Derivative Processes for Modelling Metabolic Fluxes 48

4.1 Introduction 48

4.2 Derivative processes 50

4.3 Applications and results 54

4.3.1 Oscillating signals 54

4.3.2 Lotka–Volterra *predator–prey* model 56

4.3.3 Linear pathway 58

4.3.4 Branched pathway 61

4.3.5 *Escherichia coli* nitrogen assimilation 63

4.4 Discussion and conclusions 66

5 Bayesian Nonparametric Approaches to Oscillatory Systems and the Nyquist Limit 68

5.1 Introduction 68

5.2 Methodology 71

5.2.1 Nyquist ratio for oscillatory systems 71

5.2.2 The role of multiple-output Gaussian process regression 72

5.3 Phase and frequency dependence 73

5.3.1 Variations in phase 73

5.3.2 Variations in frequency and phase 77

5.4 Applications to systems biology data 78

5.4.1 *p53-Mdm2* system 80

5.5 Discussion 81

5.6 Conclusions 84

6 A Graph Theoretical Approach to Data Fusion 85

6.1 Introduction 85

6.1.1 Bayesian integration of multiple datasets 88

6.1.2 Overview of Bayesian consensus clustering 89

6.2 Methodology 90

6.2.1 Dirichlet process mixtures 90

6.3 Ways of integrating results from different sources 95

6.3.1 Heuristic score of clustering similarity 98

6.4 Applications and results 99

6.4.1 *Saccharomyces cerevisiae* time course dataset 99

6.4.2 Integrating cell cycle datasets 100

6.4.3 Breast cancer data 106

6.4.4 Sporadic inclusion body myositis 107

6.5 Discussion and conclusions 111

7 Conclusion **114**

7.1 Summary 114

7.2 Potential future work 116

A A Supplementary material for chapter 6 **117**

A.1 Introduction 117

A.2 Inference of the hyper-parameters 117

A.2.1 Saccharomyces cerevisiae example 119

A.2.2 Yeast cell cycle datasets example 119

A.3 *Yeast cell cycle* datasets 120

B Flux Balance Analysis and Gaussian Process Constrains **122**

Bibliography **127**

List of Tables

2.1 Estimated values of parameters of two component mixture model . . . 19

2.2 Contingency table for two clusterings 28

3.1 Description of parameters used in *p53–Mdm2* ODE model 40

6.1 Comparison between BCC overall and *Approach 1* final clusterings . . . 107

6.2 A list of proteins that interact with human genes used in our study . . 110

A.1 List of genes that cluster together across gene expression, ChIP-chip and protein-protein interaction datasets. 120

A.2 List of genes that cluster together across gene expression and ChIP-chip datasets. 121

List of Figures

1.1 Overview of mathematical and computational methods 3

2.1 Performance of Metropolis algorithm. 11

2.2 Data classification. 15

2.3 Graphical representation of finite mixture model. 16

2.4 Performance of Gibbs sampling. 17

2.5 Artificial data set to illustrate IGMM. 27

2.6 Illustration of linear filter. 32

3.1 A simplified scheme for gene expression. 38

3.2 Predictions with MGPs model for gene expression data. 39

3.3 Imputing data in *p53–Mdm2* system. 42

3.4 Illustration of *Hes1* regulatory system. 44

3.5 Predictions with MGP model for *Hes1* system 46

4.1 Derivative processes for two oscillating signals. 55

4.2 Derivative processes for Lotka–Volterra predator–prey model. 57

4.3 Pathway information. 59

4.4 Derivative processes for linear metabolic pathway. 60

4.5 Derivative processes for branched metabolic pathway. 62

4.6 Predicting metabolic fluxes in *E. coli* with derivative processes. 64

4.7 Flux and glutamine synthetase (GS) comparison. 66

5.1 Six simulated datasets. 74

5.2 MGPs for two signal that have variation in phase. 76

5.3 Independent Gaussian process models. 77

5.4 MGPs for two signal that have variation in phase and frequency. 79

5.5 Multiple-output Gaussian process model for *Mdm2-p53* system. 80

5.6 MGPs for the *p53-Mdm2* system. 82

6.1 Illustration of graph-theoretical approach to data integration. 87

6.2 Data integration across six artificial gene expression datasets. 98

6.3 Yeast cell cycle: integration of transcription factor binding and gene expression data. 102

6.4 Comparison between *Approach 1* and MDI. 103

6.5 Yeast cell cycle: integration of transcription factor binding, gene expression data and protein-protein interaction data. 105

6.6 Integrating biomedical myositis datasets. 109

A.1 Yeast cell cycle: MCMC diagnostics plots. 118

B.1 A metabolic system. 124

B.2 Graphical MGPs model for metabolic fluxes. 126

Chapter 1

Introduction

1.1 Overview of modelling strategies

All cells are constantly exposed to ever-changing environmental conditions; and in order to adapt, these cells should alter their internal mechanisms – for instance metabolism (Cairns et al., 2011; Kotte et al., 2010; Nielsen, 2003). On the other hand, how cells respond to different environmental stimuli is highly dependent on the processes that occur at the molecular level; and errors at this stage may have deleterious consequences on the overall cell behaviour and function (Behrens et al., 2014; Ross and Kaina, 2013; Lord and Ashworth, 2012).

Experimental techniques play an important role in studying regulatory processes at cellular and molecular levels; and recent developments in high throughput technology allow us to monitor and collect data from genomics, transcriptomics, proteomics and metabolomics experiments (Chen and Snyder, 2012; Gehlenborg et al., 2010; Joyce and Palsson, 2006). Collectively these experiments provide a way to explore the majority of cellular components and their functions across a wide range of cell types, tissues and organisms. Although experimental techniques continue to evolve

and contribute to the overall our understanding about biological complexity and undergoing processes, we still necessitate advanced methodology that could process such data and guide us towards concise conclusions.

Mathematical reasoning and model development conveys the idea how we understand the world; and such approaches are being increasingly proposed as modern tools to interpret biological processes behind experimental data. Depending on the biological knowledge and amount of data available, mathematical and computational techniques can be broadly classified as being either mechanistic (knowledge driven) or empirical (data driven) methods (see Figure 1.1) (Birtwistle et al., 2013; Hua et al., 2006; Janes and Yaffe, 2006). Mechanistic approaches require thorough biochemical knowledge of the system, and are based on physical laws and principles that govern it. This means that modelling necessitates a detailed specification of cellular (molecular) reactions, which are occurring in the system, and corresponding reaction rates. Differential equations have become the most widely used tool for capturing how such systems evolve over the time (Aldridge et al., 2006). However, there are difficulties associated with them, it is hard or nearly impossible to consider all cellular species in the model; for this reason differential equation models might be limited only to small pathway studies. Equally, it is hard to obtain all necessary kinetic parameters using both experimental and computational tools (Ashyraliyev et al., 2009). Moreover, differential equations are impossible to apply if knowledge about the system is incomplete, and data driven modelling might be preferred to reverse engineer the properties of the system (Villaverde and Banga, 2014; Quo et al., 2012).

High throughput experiments generate quantitative data. For this reason potential relationships can be extracted using, for example clustering, logic-based and network inference methods (Thorne et al., 2013; Morris et al., 2010; Janes and Yaffe, 2006; Friedman et al., 2000). Such techniques allow us to explore the data and propose hypotheses about the underlying mechanics of the system. On the other hand, at the

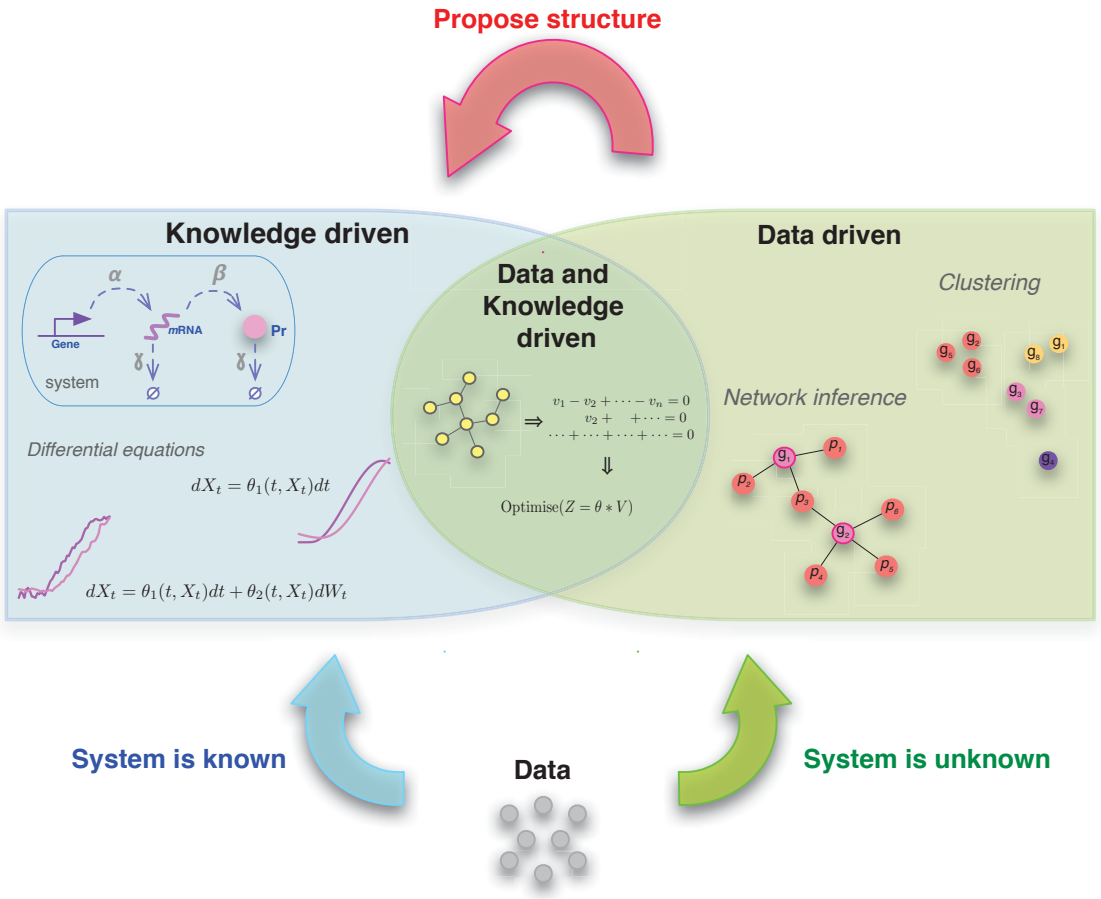


Figure 1.1: Modelling tools should be carefully selected to capture the most from the experimental data. If the underlying biological mechanism is not fully understood it is more appropriate to apply data driven techniques, such as clustering or network inference, to explore the data and generate hypotheses about the underlying mechanisms. By contrast, if explicit knowledge about the interacting species and kinetic rates is available, the data can be modelled using mechanistic approaches, such as differential equations. In between these two extremes there are methods that are data driven and are based on a partial/incomplete mechanistic knowledge.

interface of these two modelling extremes are positioned methods that are driven by both data and knowledge, which is usually incomplete (e.g. only the stoichiometry of the system is known).

1.2 Motivation and objectives of this work

Cellular behaviour is affected by a broad range of internal and external factors, and in order to fully understand how cells function as a whole, systems biology seeks to bring together ideas from a broad range of disciplines, e.g. statistics, machine learning or computing. With increasing diversity and complexity of biological data grows the pressure for new sophisticated tools that could explore what the data can provide and equally broaden our understanding about the underlying processes.

In this work we seek to combine experimental data with novel theoretical developments in statistics and machine learning approaches in order to propose new modelling tools for studying regulatory processes at the cellular and molecular levels. For this reason this thesis will be focusing on modelling from a Bayesian nonparametric perspective.

In recent years Bayesian nonparametric modelling has been the subject of rigorous research, and example techniques include Dirichlet process mixture models and Gaussian process regression techniques. These Bayesian nonparametric tools are particularly attractive as they promise us a lot of flexibility in modelling complex data. This means that the complexity of a model can adapt to the data structure, and Bayesian nonparametric tools can naturally take into account a model selection step. For example, in Gaussian process regression this would mean that with the growing number of observations we can better estimate a correlation structure, and in turn obtain a better fit to the data; and Dirichlet process mixture models allow us to infer the number of mixture components, and this means that we can infer the most appropriate model that describes our data (Orbanz and Teh, 2010). Bayesian nonparametric techniques can be viewed as data driven modelling tools due to their ability to explore the data and propose a structure. By combining data driven tools and partial mechanistic knowledge of the biological system, this work aims to study regulatory processes

that occur at the cellular level. Further, by applying novel nonparametric modelling tools on diverse data types we aim to generate new hypotheses about the underlying regulatory mechanisms that occur at the molecular level.

1.3 Thesis outline

Chapter 2

In this chapter we provide an overview of theoretical material that stands as a basis for the following research chapters.

Chapter 3

In this chapter we explore the applicability of multiple-output Gaussian process approach to model gene expression, capture the dependencies between *m*RNA and protein levels and to impute the missing measurements. The method is data driven and uses partial knowledge of the underlying system.

Chapter 4

In this chapter we develop a new approach to model the dynamical behaviour of metabolic fluxes. This technique is based on multiple-output Gaussian processes and uses stoichiometric knowledge about the system; our method is driven by both partial knowledge of the metabolic system and data. We apply this technique to a range of biological problems and explore regulatory processes at the cellular level.

Chapter 5

In this chapter we apply multiple-output Gaussian processes to reconstruct oscillatory systems. We further explore the effects of sparse sampling on the ability to accurately reconstruct the underlying signals.

Chapter 6

In this chapter we develop a new approach to integrate a broad range of genomic data sources. Our method is based on Dirichlet process mixture models and uses graph theory concepts to identify shared clusters. This allows us to generate hypotheses about underlying regulatory mechanism at the molecular level.

Chapter 7

In this chapter we summarise the results and discuss guidelines for future research.

1.4 List of publications

- Bayesian non-parametric approaches to reconstructing oscillatory systems and the Nyquist limit Žurauskienė J., Kirk P., Thorne T., Stumpf M.P.H.; Physica A: Statistical Mechanics and its Applications, Vol. 407, p.33 – 42, 2014.

- Derivative processes for modelling metabolic fluxes. Žurauskienė J., Kirk P., Thorne T., Pinney J., Stumpf M. Bioinformatics. 2014 Jul 1;30(13):1892-8. doi: 10.1093/bioinformatics/btu069. Epub 2014 Feb 26.

- StochSens – Matlab package for sensitivity analysis of stochastic chemical systems. Komorowski M., Žurauskienė J., Stumpf M.P. Bioinformatics. 2012 Mar 1;28(5):731-3. doi: 10.1093/bioinformatics/btr714.

Chapter 2

Theoretical Background

2.1 Introduction

In order to perform Bayesian nonparametric modelling it is necessary to introduce several theoretical concepts, and this chapter covers all necessary theoretical background material. The theory discussed here will serve as a starting point for the subsequent research chapters.

Below we introduce: (2.2) the Markov chain Monte Carlo concept; this is the most popular computational tool that can be applied for parameter inference; (2.3) Gaussian mixture models, a technique to perform parametric density estimation and clustering; (2.4) Dirichlet processes, an introduction to nonparametric modelling; (2.5) Infinite Gaussian mixture models, a way to construct Dirichlet processes with a focus on clustering; (2.6) Gaussian processes regression, the basics how to perform regression in a nonparametric fashion; and (2.7) multiple–output Gaussian process regression, explains how to perform nonparametric regression and how to include the dependencies while modelling several variables.

2.2 Markov chain Monte Carlo

During the last three decades there has been growing interest in the development of computational Markov chain Monte Carlo (MCMC) techniques for applications in stochastic systems and especially for computation in Bayesian statistics (Gamerman and Lopes, 2006). A substantial amount of work has been done with the aim of investigating the mathematical properties and characteristics of these methods and expanding existing generic techniques for researching new challenging questions. With the development of MCMC techniques it has become possible to analyse many complex statistical problems that previously were not tractable.

One of the most popular application areas of MCMC methodology involves Bayesian inference, where the Bayesian approach necessitates the integration of high dimensional functions (probability distributions) in order to make a prediction or inference about model parameters. In most cases this step of integration is computationally extremely difficult and, occasionally, impossible. However, MCMC techniques enable this integration to be performed by drawing samples from the distribution of interest by constructing a Markov chain where the subsequent sample value depends on the preceding value of the sample. These MCMC methods were first developed by Metropolis et al. (1953) and later the Gibbs sampling technique (Geman and Geman, 1984) was developed as a special case of the Metropolis–Hastings algorithm.

In order to take advantage of the Metropolis–Hastings algorithm and the Gibbs sampler, it is necessary to introduce the concepts of Markov chains and Monte Carlo integration.

The Monte Carlo integration method explains how to evaluate the integral

$$\int_a^b f(x)\mathrm{d}x, \tag{2.1}$$

where the integrand $f(x)$ is a function with potentially complex structure. This particular integration is a numerical computational technique that employs a random number generator and replaces the complex integrand $f(x)$ with a probability density function $p(x)$ and a function $g(x)$, so that the integral (2.1) can be expressed as

$$\int_a^b f(x)\mathrm{d}x = \int_a^b p(x)g(x)\mathrm{d}x = E[g(X)],$$

where E is the expectation of function $g(x)$. By drawing samples $\{X_i, i = 1, 2, ..., n\}$ from distribution $p(X)$, the expectation $E[g(X)]$ can be approximated in the following way

$$E[g(X)] \approx \frac{1}{n}\sum_{i=1}^{n} g(X_i) \tag{2.2}$$

If samples $\{X_i\}$ are independent, the accuracy of this approximation improves with the number of samples n; this is guaranteed by the Law of Large Numbers[1]. One of the major reasons why it is difficult to generate independent samples from $p(X)$ is the complex or intricate structure that it has in many problems of interest. Nevertheless, one way of overcoming this problem is by constructing a Markov chain where the function $p(X)$ is chosen to be the appropriate stationary distribution.

A Markov chain is a sequence of random variables $\{X_0, X_1, ...\}$ where the transition probabilities or transition kernels between different values in the state space $\mathbb{S}$ depend only on the current state of the random variable. The mathematical description of the Markov chain is

$$P(X_t \in A|X_0, X_1, ..., X_{t-1}) = P(X_t \in A|X_{t-1}),$$

where A is a given set and the conditional probability P is called a transition kernel of the Markov chain. The above means that knowledge of X_{t-1} determines the future probability distribution of the random variable X_t.

[1] LLN: for large n the sample mean approximates the expectation

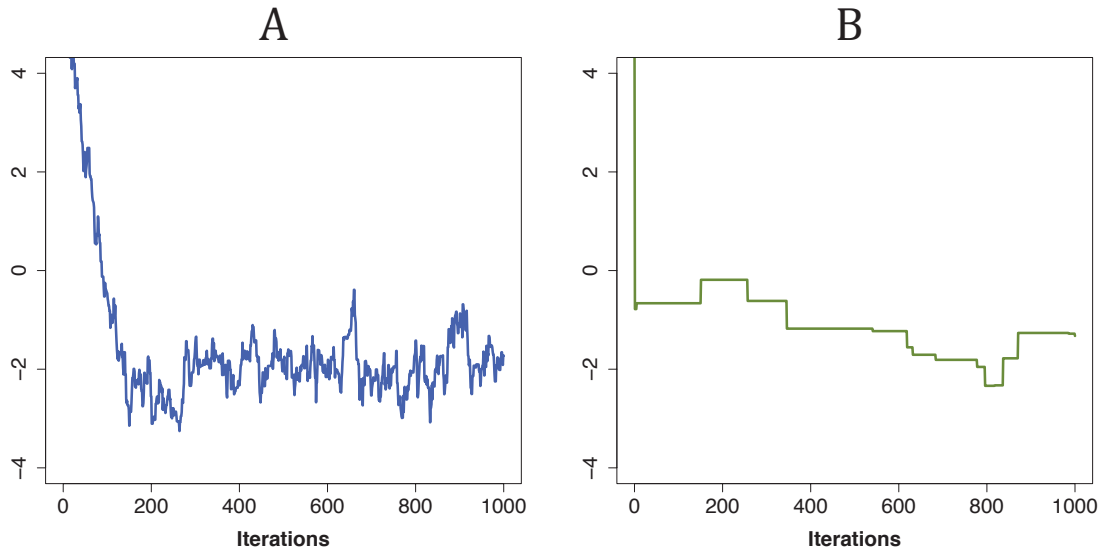


Figure 2.1: Performance of Metropolis algorithm with the stationary distribution $\mathcal{N}(-2,1)$. **(A)** The proposal distribution $\mathcal{N}(X,10)$; **(B)** The proposal distribution $\mathcal{N}(X,0.5)$.

In the case where $\mathbb{S}$ is a discrete state space, the probability that the chain is at state $x_j \in \mathbb{S},\ j = 1,..,r$ at time t can be written in a special matrix form $\mathbf{P}$. Let us denote $P(X_t = x_j) = \pi_j^{(t)}$ when $\pi^{(t)}$ is a vector of all state probabilities. If $\mathbf{P}$ is a probability transition matrix, the probability of the chain moving from one state into another is:

$$\pi^{(t)} = \pi^{(t-1)}\mathbf{P}.$$

It is easy to show that

$$\pi^{(t)} = \pi^{(t-1)}\mathbf{P} = \pi^{(t-2)}\mathbf{P}^2 = ... = \pi^{(0)}\mathbf{P}^t.$$

That is, the transition matrix $\mathbf{P}$ and starting distribution $\pi^{(0)}$ explain the probability distribution for the state for all times (Wilkinson, 2006).

The Markov chain is called irreducible if there is a positive probability for any state to be reachable from any other state. The property of aperiodicity stops the chain from

oscillating between different states (Gilks et al., 1996). If the Markov chain satisfies irreducibility and aperiodicity conditions, it has the general property that the chain can reach a stationary distribution $\pi : \pi = \pi\mathbf{P}$, such that probabilities of being in any given state are independent of the starting value. As a result the Markov chain can be used in order to estimate the expectation of function $g(X)$; for this purpose it is common to take an ergodic average (Wang et al., 2000; Geyer, 2011) given by

$$\bar{g} = \frac{1}{n-m} \sum_{t=m+1}^{n} g(X_t), \tag{2.3}$$

where m refers to a number of discarded or “burn-in" samples on which dependence on initial conditions is lost.

The construction of such a chain is expressed by the general Metropolis-Hastings algorithm or another widely applied algorithm, the Gibbs sampler, which is a particular case of the Metropolis-Hastings algorithm.

Though there exist well-studied methods for constructing a Markov chain, there are some difficulties in the determination of how many iteration steps are required for the chain to converge to a stationary distribution. If the stationary distribution is reached quickly from a starting position, then the chain has rapid mixing - that is desired behaviour (see Figure 2.1**A**). By contrast, Figure 2.1**B** illustrates a poorly mixing chain. To assess the mixing time m_t the chain can be run several times with a different starting positions and estimates (2.3) can be computed for comparison. It is worth to note that running a Markov chain for n times does not necessarily guarantee that it has reached a stationary (target) distribution; nevertheless it is possible to perform some tests in order to assess the convergence. The most common practice is to perform visual tests by plotting the chain values agains all iterations (Figure 2.1). Such a plot is frequently known as a *traceplot* and might be accompanied by a plot of moving averages. Equally, there exist more sophisticated statistical tests for chain convergence analysis, and more details can found in e.g. Gelman and Rubin (1992), Geweke (1992), Cowles and

Carlin (1996).

There are several application areas where this MCMC methodology is particularly useful. These techniques are especially convenient in Bayesian parameter inference when it is required to sample from posterior distributions. For example, MCMC techniques are of particular importance when applied in Bayesian mixture modelling framework to cluster heterogeneous data.

Note on MCMC algorithms

Here we review some common algorithms to construct Markov chains that converge to a target probability distribution. Metropolis-Hasting and the Gibbs sampler are the fundamental techniques that are most widely used for this purpose (for more thorough discussion see e.g. Gilks et al. (1996) or *"A primer on Markov Chain Monte Carlo*" by Peter J. Green in Barndorff-Nielsen et al. (2000)). Below, we briefly explain the main principles.

Assume we would like to estimate some parameter θ (e.g. mean of a normal distribution, variance etc.); let the current value of this parameter be θ_t. Using the Metropolis-Hastings algorithm, the update for this parameter, θ_{t+1}, is calculated by sampling a candidate point $\theta^\star$ from a proposal distribution $q(|\theta_t)$. We accept this candidate point with a probability $\alpha(\theta_t, \theta^\star)$, where

$$\alpha(\theta_t, \theta^\star) = \min\left(1, \frac{\pi(\theta^\star)q(\theta_t|\theta^\star)}{\pi(\theta_t)q(\theta^\star|\theta_t)}\right).$$

Above, π denotes a distribution of interest. If we accept a candidate point, $\theta^\star$, then we set $\theta_{t+1} = \theta^\star$; otherwise, we reject it and thus, the update for this parameter is set to be the same, $\theta_{t+1} = \theta_t$.

The Gibbs sampler is a special case of Metropolis-Hastings algorithm where all candi-

date points are accepted with a probability $\alpha(\theta_t, \theta^\star) = 1$. And we sample each variable one by one conditioned on most recent values of all other variables. In this case a proposal distribution has a special form and it is chosen to be a conditional distribution (see more in books by e.g. Bishop (2006); Murphy (2012)).

2.3 Finite Gaussian mixture models

Bayesian modelling provides a great variety of computational techniques that can be applied for statistical data analysis. An important class of these methods are clustering techniques that permit the detection of similarities between experimentally observed data. A finite mixture model is one such parametric modelling technique that has received much attention over the years due to its flexibility in estimating unknown densities, in modelling complex shapes of distributions and, in applications of this framework as an efficient technique to explain and characterise complex data. For example such analysis allows us to cluster heterogeneous data, where each component in the mixture model represents a cluster within the population. Other widely applied clustering techniques such as hierarchical or k-means rely on distance functions between the data; the mixture model provides simultaneously density estimation for each cluster, offers an option to select a parametric family for mixture components and naturally incorporates the classification property of the data (see Figure 2.2). Equally, the statistical inference of mixture model parameters can be done via sophisticated computational techniques, such as expectation-maximisation (EM) algorithm (Dempster et al., 1977; Bilmes et al., 1998), sequential Mote Carlo (SMC) (Del Moral et al., 2006) or MCMC.

A finite mixture of distributions can be defined as a convex combination (Marin et al., 2005),

$$\sum_{k=1}^{K} \pi_k f_k(x), \qquad \pi_k > 0, \qquad \sum_{k=1}^{K} \pi_k = 1, \tag{2.4}$$

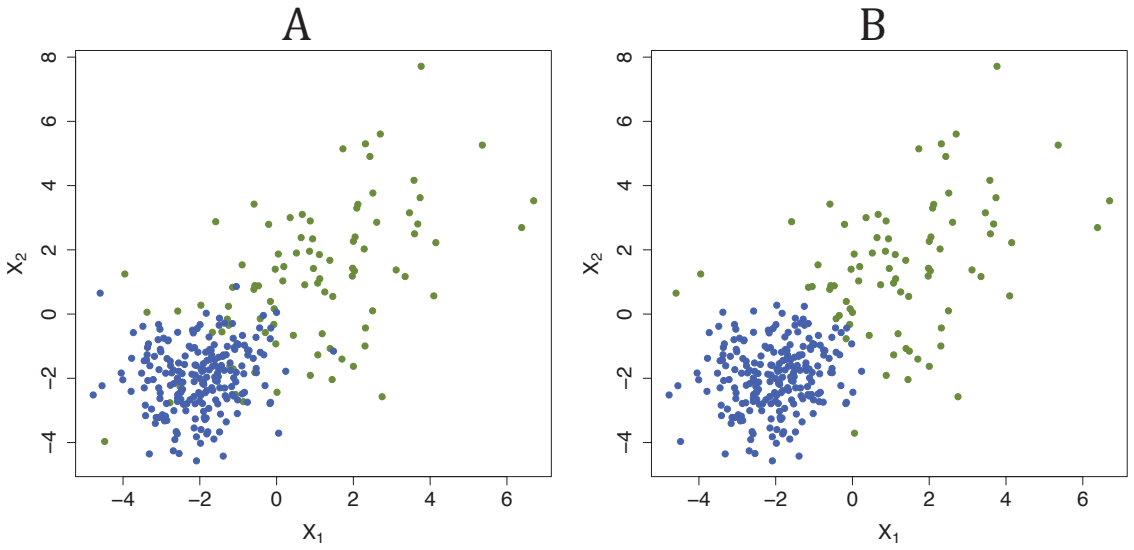


Figure 2.2: Data classification. **(A)** Simulated data from $F(x_i) = \pi_1 \mathcal{N}(x_i; \mu_1, \Sigma_1) + \pi_2 \mathcal{N}(x_i; \mu_2, \Sigma_2)$, $i = 1, ..., 300$ model; where blue colour denotes component with parameters $\mu_1 = (-2, 2)$ and $\Sigma_1 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$; green – with parameters $\mu_2 = (1, 1)$, $\Sigma_2 = \left(\begin{smallmatrix} 4 & 2 \\ 2 & 4 \end{smallmatrix}\right)$; **(B)** the same data classified with a finite Gaussian mixture model.

where f_k represents a distribution and π_k are weights. In parametric mixture models, f refers to a parametric family of functions, for example *exponential*, that are parameterised with a set of parameters θ_k,

$$\sum_{k=1}^{K} \pi_k f(x|\theta_k).$$

In many practical cases f corresponds to a Gaussian distribution $\mathcal{N}(\mu_k, s_k^{-1})$ with mean μ_k and precision s_k (inverse of the variance).

Next we discuss the applications of Gaussian mixture models (Medvedovic et al., 2004; Pan et al., 2002). Let $\mathbf{x} = \{x_1, x_2, ..., x_N\}$ be the observed data, which is independently drawn from some distribution $g(\mathbf{x})$ and N is the sample size. It is intended to model the distribution from which the data x_i are drawn as a mixture of distributions. That is, the true density function $g(\mathbf{x})$ of the sample $\mathbf{x}$ can be approximated with the parametric

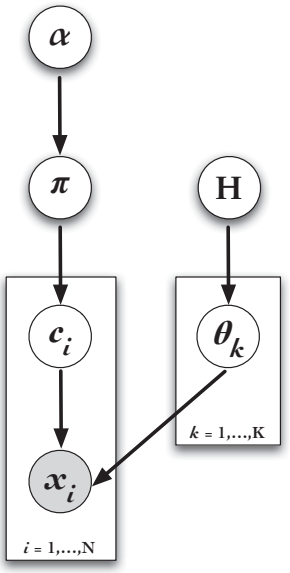


Figure 2.3: Graphical representation of finite mixture model, where nodes represent random variables; shaded nodes – observed variables; arrows/links – probabilistic relationships; and plates correspond to variables that repeat in this model. (See more about graphical models in e.g. (Bishop, 2006)).

finite Gaussian mixture model $\bar{g}(\mathbf{x})$ with K components,

$$g(\mathbf{x}) \simeq \bar{g}(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, s_k^{-1}), \tag{2.5}$$

where $\mathcal{N}$ is a Gaussian distribution with mean μ_k and precision s_k; π_k are the mixing proportions that satisfy the last two properties of equation (2.4). For the purpose of estimation of (2.5), the statistical model with conjugate distributions can be applied,

$$\begin{aligned} \theta_k &\sim \mathrm{H}, \quad k = 1, ..., K \\ \pi &\sim \mathscr{D}\left(\tfrac{\alpha}{K}, ..., \tfrac{\alpha}{K}\right) \\ c_i|\pi &\sim \mathrm{Discrete}(\pi) \\ x_i|\theta_{c_i} &\sim \mathrm{F}(\theta_{c_i}). \end{aligned} \tag{2.6}$$

The above mixture model has an incomplete data structure because it is not known which mixture component generated which data point. It means that missing infor-

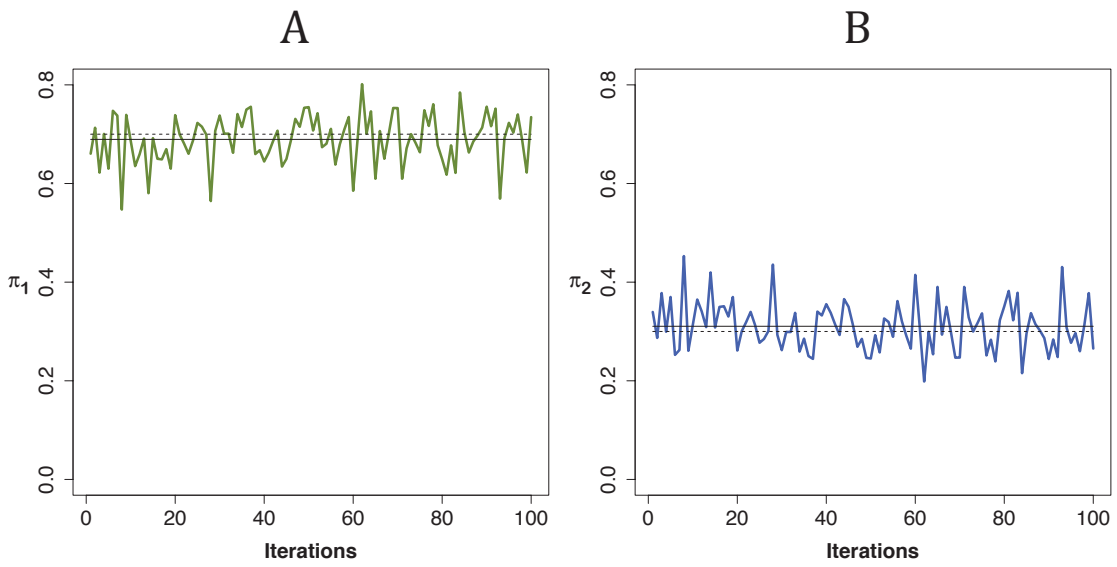


Figure 2.4: Performance of Gibbs sampling. **(A)** and **(B)** illustrates the stationary Markov chains after 12,000 iterations, taking first 2,000 samples for burn-in period, and recording each 100 sample. Here solid black lines corresponds to the true values of $\pi_1 = 0.7$ and $\pi_2 = 0.3$, and dashed lines represent the estimated values respectively. **(A)** Stationary Markov chain for mixing proportion π_1. **(B)** Stationary Markov chain for mixing proportion π_2.

mation has to be introduced, which is precisely the cluster membership. The complete data structure is denoted by a set $\{(x_i, c_i), i = 1, ..., N\}$, where c_i takes the values from $1, ..., K$; c_i is a "stochastic" variable and encodes this information in the model above; we draw it from a discrete (Multinomial) distribution given that proportions π were generated from a Dirichlet distribution with concentration parameter α. The Bayesian paradigm necessitates putting the prior over unknown parameters θ_{c_i} and mixing proportions π_k in order to make inference about them. The graphical representation of this model is given in Figure 2.3.

The inference for mixture models with known number of components can be done via MCMC methods that approximate the posterior distributions of model parameters $\{\pi_k, \mu_k, s_k\}$. Since the missing data structure is introduced in the model above, it is easy to implement the Gibbs sampling algorithm for inference of these model parameters. The Gibbs sampling updates each variable by drawing samples from a conditional

distribution, provided that all other parameter values of the model are given. The following example will illustrate the method. A bivariate mixture of two components from which we simulated a dataset of 300 particles can be written as:

$$F(x_i) = \pi_1 \mathcal{N}(x_i; \mu_1, \Sigma_1) + \pi_2 \mathcal{N}(x_i; \mu_2, \Sigma_2), \quad i = 1, ..., 300.$$

Each mixture component consist of two parameters: mean μ_k and precision matrix Σ_k^{-1}, $k = 1, 2$. For both means a multivariate Gaussian conjugate prior with fixed hyper-parameters λ and r is given:

$$p(\mu_k|\lambda, r) \sim \mathcal{N}(\lambda, r^{-1}),$$

where λ is mean vector, r is precision matrix. Equally, for both precision matrices a Wishart (multivariate Gamma) conjugate prior with fixed hyper-parameters β and ω is given (McLachlan and Peel, 2000):

$$p(\Sigma_k^{-1}|\beta, \omega) \sim \mathcal{W}(\beta, \omega^{-1}),$$

where β is a scalar and ω is matrix. After 12,000 iterations, updating all parameters in turn by sampling from conditional distributions and taking 2,000 burn-in for the period, and recording each 100 sample, the Figure 2.4 illustrates the performance of Gibbs sampling. The Table 2.1 summarises the true and inferred values of mixture model parameters.

In many real world situations we might not know *a priori* how many unique subgroups exist within the dataset. This is particularly the case when dealing with extremely large biological datasets were the number of clusters cannot be assessed at least visually. For this reason, it is more accurate to assume that the number of classes are not finite but instead at least in principle infinite. This perspective about the model and parameters is common in nonparametric Bayesian statistics such as Dirichlet process

Table 2.1: True values of parameters of two component mixture model and summary of their posterior distributions

	True value	Posterior mean	Posterior variance
μ_1	(-1.979, -1.953)	(-1.913, -1.890)	(0.006, 0.008)
μ_2	(1.321, 1.302)	(1.624, 1.564)	(0.141, 0.107)
Σ_1	$\begin{pmatrix} 0.855 & 0.068 \\ 0.068 & 0.942 \end{pmatrix}$	$\begin{pmatrix} 0.945 & 0.145 \\ 0.145 & 1.020 \end{pmatrix}$	$\begin{pmatrix} 0.012 & 0.007 \\ 0.007 & 0.010 \end{pmatrix}$
Σ_2	$\begin{pmatrix} 4.167 & 1.653 \\ 1.653 & 3.459 \end{pmatrix}$	$\begin{pmatrix} 3.704 & 1.142 \\ 1.142 & 2.930 \end{pmatrix}$	$\begin{pmatrix} 0.697 & 0.449 \\ 0.449 & 0.384 \end{pmatrix}$
π_1	0.7	0.737	0.002
π_2	0.3	0.263	0.002

mixture models or Gaussian process regression. These are the methods that will be discussed below.

2.4 Dirichlet processes

To perform a parametric Bayesian analysis requires the introduction of a detailed probability model for parameters and observations; by contrast Bayesian nonparametric procedures do not demand such a detailed probabilistic hypothesis. For this reason it is an alternative approach to parametric modelling.

The Dirichlet process mixture model (also known as infinite mixture model) is a Bayesian nonparametric data modelling technique that uses a stochastic Dirichlet process as a prior over the distribution function F (Ferguson, 1973). These mixtures with a countably infinite number of components have been extensively applied for density estimation and data clustering where the number of components is not determined a priori. Furthermore, Dirichlet process mixture models are an alternative to other computational methods that seek to estimate the number of components within the mixture, such as the Bayesian information criterion (BIC).

Dirichlet processes (DPs) are stochastic processes that can be understood as a distri-

bution over distributions (Teh, 2010), where each draw from this process is a discrete distribution. A formal definition of DPs originates in work by Ferguson (Ferguson, 1973) and was reformulated by Teh (Teh et al., 2006) in the following way: Let $(\mathbb{X}, \mathscr{A})$ be a measurable space and $\mathscr{A}$ be a σ-field of subsets, with H a probability measure[2] on the space. Let α be a positive real number. A Dirichlet process, $DP(\alpha, H)$, is defined as the distribution of a random probability measure G over $(\mathbb{X}, \mathscr{A})$ such that, for any finite measurable partition $(A_1, A_2, ..., A_m)$ of $\mathbb{X}$, the random vector $G(A_1), G(A_2), ..., G(A_m)$ is distributed as a finite dimensional Dirichlet distribution with parameters $(\alpha H(A_1),\ \alpha H(A_2), ...,\ \alpha H(A_m))$,

$$\big(G(A_1), G(A_2), ..., G(A_m)\big) \sim \mathscr{D}(\alpha H(A_1), \alpha H(A_2), ..., \alpha H(A_m)).$$

It is accepted to write $G \sim DP(\alpha, H)$ if a random probability measure G is distributed according to the DP. In addition, DPs can be understood as a measure on measures, because each draw G from DP can be expressed in terms of the infinite sum (Sethuraman, 1994),

$$G = \sum_{i=1}^{\infty} \beta_i \delta_{\theta_i} = 1, \tag{2.7}$$

where θ_i is an independent random variable distributed according to H; δ_{θ_i} is called an atom at θ_i. Alternatively, according to Ferguson (1973) θ_i is a measure that gives mass one to the point θ_i; and β_i are parameters that depend on the parameter α. This (2.7) sum is known as a definition of *Stick Breaking* prior (Sethuraman, 1994).

The two parameters that define the DP are the concentration parameter, α, and base distribution, H. The base distribution H can be understood as a mean distribution of the DP, that is, if A is any measurable subset of $\mathbb{X}$, then $E[G(A)] = H(A)$. The

[2] Let A be the set of subsets of a space $\mathbb{X}$. A measure $G : A \to \Omega$ assigns a nonnegative value to any subset of $\mathbb{X}$. G is called a probability measure when $\Omega = [0, 1]$.

concentration parameter α determines an inverse of variance, and we have

$$V[G(A)] = \frac{H(A)(1-H(A))}{\alpha+1}.$$

Hence it can be seen that the larger the concentration parameter α, the smaller the variance (Teh, 2010).

Since a draw G from $DP(\alpha, H)$ is a distribution, it means we can sample from G as well. Let $A_1, A_2, ..., A_m$ be a subset of $\mathbb{X}$, and let us independently sample a sequence of $\theta_1, \theta_2, ..., \theta_n \sim G$ that takes values in space $\mathbb{X}$. Let n_i be the number of observed values of θ_j in A_k. In Bayesian nonparametric modelling, DP is used as a prior over unknown distribution (in this case G), thus the intention is to find the posterior distribution of G given the values of the $\theta_1, \theta_2, ..., \theta_n$. The posterior distribution is also a Dirichlet process with updated parameters,

$$G|\theta_1, \theta_2, ..., \theta_n \sim DP\left(\alpha+n, \frac{\alpha H + \sum_{j=1}^{n} \delta_{\theta_j}}{\alpha+n}\right)$$

that can be used to approximate the fundamental distribution (Teh, 2010).

The Dirichlet process is a powerful technique that can be applied in clustering via mixture models and estimating unknown densities. This research will be concentrating on DPs that exhibit the clustering property, that is on a DP that is constructed as a Chinese restaurant process (CRP) (Aldous, 1985). The CRP is a metaphor that explains the DP construction, where tables are clusters and customers are particles. Initially, the CRP has an infinite number of tables that can seat an infinite number of customers. The first customer is always seated at the first table. The second customer can decide whether to sit at the first table or any other empty table. Summarising, the $n+1$ customer will sit either at any occupied table i with a probability proportional to the number of customers sitting there n_i or will join a new table with a probability proportional to α. This explanation follows from the Blackwell–MacQueen urn scheme (Blackwell and

MacQueen, 1973), a limiting process of the Pólya urn. Let $\theta_1, \theta_2, ..., \sim G$ be a draw from G in which all the components are conditionally independent given G, where the conditional distribution of θ_{n+1} given G is

$$\theta_{n+1}|\theta_1, \theta_2, ..., \theta_n, G \sim \sum_{j=1}^{n} \frac{\delta_{\theta_j}}{n+\alpha} + \frac{\alpha}{n+\alpha} H.$$

It means, that there is a positive probability that θ can have the same value. Let $\psi = (\psi_1, \psi_2, ..., \psi_k)$ denote the set of distinct values of $\theta_1, \theta_2, ..., \theta_n$, where $k \leq n$, it follows that the previous conditional distribution can be rewritten as,

$$\theta_{n+1}|\theta_1, \theta_2, ..., \theta_n, G \sim \sum_{l=1}^{k} \frac{N_l}{n+\alpha} \delta_{\psi_l} + \frac{\alpha}{n+\alpha} H,$$

where N_l is the number of θ_j taking values equal to ψ_l. The above specification allows us to specify CRP as a distribution on partitions that captures the clustering effect of the DP (Teh, 2010).

2.5 Infinite Gaussian mixture models

Infinite mixture models, otherwise known as Dirichlet process mixture models, can be used as a Bayesian nonparametric modelling tool to cluster data of interest. One way of introducing Dirichlet process mixture models is by taking the limit as K goes to infinity of the finite mixture models with a fixed number of components K (Neal, 2000).

Here we will consider a Gaussian finite mixture model with K components that was described in section (2.3). The inference of model parameters can be done via MCMC techniques, in particular Gibbs sampling, which allows the generation of the samples from complicated distributions. A Gibbs sampler updates each variable by drawing samples from a conditional distribution given all other parameters of the model. For

Bayesian model analysis it is necessary to specify the priors for each parameter and derive the conditional distributions required for Gibbs sampling. For this purpose we will follow the approach proposed by Rasmussen (2000) with slight variations (so that conditional posteriors would be in a good agreement with available *R* functions).

For component means μ_j and precisions s_j a Gaussian and Gamma priors with hyper-parameters λ, τ, β and ω are set,

$$p(\mu_j|\lambda,\tau) \sim \mathcal{N}(\lambda, \tau^{-1}),$$
$$p(s_j|\beta,\omega) \sim \mathcal{G}(\beta,\omega),$$

where λ is a mean, τ a precision; $\beta > 0$ and $\omega > 0$ are the shape and scale parameters of Gamma distribution.

Since it is not known a priori which mixture component generated the observation x_i, it is necessary to introduce the indicator variables c_i, $i = 1, ..., N$ that determines this information. In the mixture model framework these indicator variables are known as missing data.

For the mixing proportions a symmetric *Dirichlet* prior is given,

$$p(\pi_1, ..., \pi_K|\alpha) \sim \mathcal{D}\Big(\frac{\alpha}{K}, ..., \frac{\alpha}{K}\Big),$$

with concentration parameter $\frac{\alpha}{K}$. A conditional prior for a single indicator, all others being given, is obtained by integrating over the mixing proportions

$$p(c_i = j|c_{-i}, \alpha) = \frac{n_{-i,j} + \frac{\alpha}{K}}{N - 1 + \alpha},$$

where the subscript, "$-i$", is a short notation for all indicators excluding i; and $n_{-i,j}$ denotes the number of observations within the cluster j not including observation y_i.

By taking the limit as K goes to infinity the conditional prior has the following limits,

$$p(c_i = j|c_{-i}, \alpha) = \frac{n_{-i,j}}{N-1+\alpha}, \tag{2.8a}$$

$$p(c_i \neq c_{i'}, i' \neq i|c_{-i}, \alpha) = \frac{\alpha}{N-1+\alpha}. \tag{2.8b}$$

The conditional posterior distributions for the mean and precision parameters can be derived by multiplying the likelihood (which is a normal distribution) by appropriate priors. The following conditional posterior is for the mean,

$$p(\mu_j|\mathbf{c}, \mathbf{x}, s_j, \lambda, \tau) \sim \mathcal{N}\Big(\frac{\bar{x}_j s_j + \lambda\tau}{\tau + n_j s_j}, \frac{1}{\tau + n_j s_j}\Big), \tag{2.9}$$

where $\bar{x}_j$ denotes the sum of all observations associated with a cluster j, and n_j refers to the total number of observations in the same cluster; and below is summarised the conditional posterior for the precision,

$$p(s_j|\mathbf{c}, \mathbf{x}, \mu_j, \beta, \omega) \sim \mathcal{G}\Big(\beta + \frac{n_j}{2}, [\omega^{-1} + 0.5\sum_{i:c_i=j}(x_i - \mu_j)^2]^{-1}\Big) \tag{2.10}$$

When considering the application of infinite Gaussian mixture model it is usually hard to chose a scalar values for hyper-parameter $\lambda, \tau, \beta, \omega$. For this reason it might be necessary to assume an additional hierarchy in the model and try to infer these hyper-parameters from the data. Accordingly the priors must be chosen; for practical reason these priors might depend on the data,

$$\begin{aligned}
p\left(\lambda|\mu_x, \sigma_x^2\right) &\sim \mathcal{N}(\mu_x, \sigma_x^2),\\
p(\tau|1, \sigma_x^2) &\sim \mathcal{G}(1, \sigma_x^{-2}),\\
p\left(\beta\right) &\sim \mathcal{IG}(1/2, 1/2),\\
p(\omega|1, \sigma_x^2) &\sim \mathcal{G}(1, \sigma_x^2).
\end{aligned}$$

leading to the the following conditional posteriors,

$$p(\lambda|\mu,\tau) \sim \mathcal{N}\left(\frac{\mu_x\sigma_x^{-2}+\tau\bar{\mu}}{k\tau+\sigma_x^{-2}}, \frac{1}{k\tau+\sigma_x^{-2}}\right), \tag{2.11a}$$

$$p(\tau|\mu_j,\lambda) \sim \mathcal{G}\left(\frac{K}{2}+1, \left[\sigma_x^2+\frac{1}{2}\sum_{j=1}^{K}(\mu_j-\lambda)^2\right]^{-1}\right), \tag{2.11b}$$

$$p(\omega|s_j,\beta) \sim \mathcal{G}\left(K\beta+1, \left[\sigma_x^{-2}+\beta\sum_{j=1}^{K}s_j\right]^{-1}\right), \tag{2.11c}$$

$$p(\beta|s_j,\omega) \propto \Gamma(\beta)^k \exp\left(-\frac{1}{2\beta}\right)\beta^{K\beta-3/2}\prod_{j=1}^{K}(s_j\omega)^\beta \exp(-\beta\omega s_j), \tag{2.11d}$$

here $\sum_{j=1}^{K}\mu_j$. In order to infer the concentration parameter α we can use the approach proposed by Escobar and West (1995), where the prior on $\alpha \sim \mathcal{G}(a,b)$. This technique uses an auxiliary variable approach and first samples a value $\eta \sim \mathcal{B}(N,\alpha+1)$ and then α from a mixture,

$$p(\alpha|\eta,K) \sim \pi\mathcal{G}(a+K, b-\log(\eta)) + (1-\pi)\mathcal{G}(a+K-1, b-\log(\eta)), \tag{2.12}$$

where π are weights that can be defined by $\frac{\pi}{1-\pi} = \frac{a+K-1}{N(b-\log(\eta))}$. Thus the Gibbs sampling algorithm for the infinite Gaussian mixture model can be summarised as follows: Initialise the Markov chain with indicator variables $c = (c_1,...,c_N)$ and parameters $\theta_{rep} = (\theta_1,...,\theta_N)$, where $\theta_i = (\mu_i, s_i)$. This means that each data point is associated with a descriptive class and class parameters, and in total there are k_{rep} distinct classes. Listing 2.1 illustrates the algorithm and explains one iterative step. The algorithm consist of two **For** loops, with the first one corresponding to the Chinese restaurant process – each data point can be assigned to the existing clusters according to equation (2.8) or it can be assigned to a new cluster with a probability $p_{new} \sim \mathcal{N}(\mu_{new}, s_{new})$, where μ_{new} and s_{new} are randomly drawn from the appropriate prior distributions. The second **For** loop is precisely the Gibbs sampling step for the inference of model parameters and hyper-parameters.

Listing 2.1: IGMM algorithm

For all observations y_i, $i = 1,...,N$ **do**:
  For all distinct classes $j = 1,...,k_{rep}$ **do**:
    calculate $n_{-i,j}$
    if $n_{-i,j} > 0$ **then**
      $p_{old} \propto \frac{n_{-i,j}}{N-1+\alpha}\sqrt{s_j}\exp\left(-\frac{1}{2}(y_i-\mu_j)^2 s_j\right)$
    else
      $p_{old} \propto \frac{\alpha}{N-1+\alpha}\sqrt{s_j}\exp\left(-\frac{1}{2}(y_i-\mu_j)^2 s_j\right)$
    end if
  end
  $p_{unrep} \propto \frac{\alpha}{N-1+\alpha}(2\pi)^{(1/2)}\sqrt{s_{new}}\exp\left(-\frac{1}{2}(y_i-\mu_{new})^2 s_{new}\right)$
  normalise $p \longleftarrow (p_{old}, p_{new})$
  update c_i given such p
  update θ_i
end
update k_{rep}
For all distinct classes $k = 1,...,k_{rep}$ **do**:
  draw m_k from equation (2.9)
  draw s_k from equation (2.10)
end
draw λ from equation (2.11a)
draw τ from equation (2.11b)
draw ω from equation (2.11c)
draw β from equation (2.11d)
draw α from equation (2.12)

To illustrate the modelling with infinite mixture models, consider an artificial dataset that was generated from a six ($K = 6$) component mixture of Gaussian distributions (adjacency matrix in Figure 2.5(**A**) illustrates this dataset sorted into true clusters). Further assume, that only a dataset of 500 observations is available, whereas the information about Gaussian parameters, number of components and exact mixing proportions is not accessible. The goal is to learn how many unique component describe data the best and to classify all observations into appropriate clusters. For this reason we can apply the modelling described above and run the algorithm (2.1) for 15,000 iterations. Discarding the first 5,000 samples and recording each 20^{th} sample we arrive

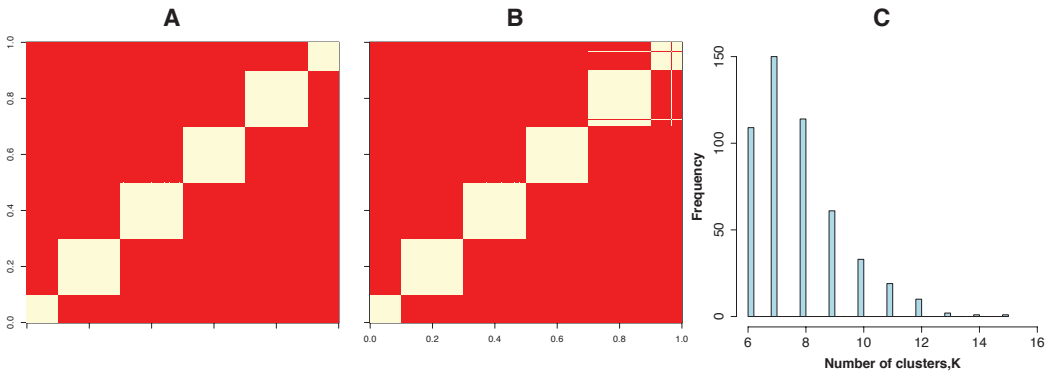


Figure 2.5: Artificial dataset of 500 data points generated from a Gaussian mixture model, $f(x) = 0.1\mathcal{N}(-5, 0.1^{-1}) + 0.2\mathcal{N}(-8, 0.2^{-1}) + 0.2\mathcal{N}(-2, 0.1^{-1}) + 0.2\mathcal{N}(-1, 0.1^{-1}) + 0.2\mathcal{N}(4, 0.4^{-1}) + 0.1\mathcal{N}(6, 0.2^{-1})$. **(A)** The co-occurence matrix that represents generated data; here yellow colour corresponds to a value 1, and illustrates two data points being in the same cluster; red – corresponds to 0 value, which means that these two data points do not cluster together. **(B)** The same data points classified using IGMM, where final clustering was estimated with Fritsch and Ickstadt (2009) criterion. **(B)** A histogram for the number of components K generated using samples from posterior.

at 500 samples from posterior distribution. Figure 2.5(**C**) shows the posterior distribution (a histogram) for the number of components, $K = 7$. Given these MCMC samples from posterior distribution it is not immediately clear how to identify the overall single clustering that would enable us to sort data points into appropriate clusters. This is due to the fact that K is varying across all samples. However, the overall clustering can be estimated by following Fritsch and Ickstadt (2009) approach, which is based on maximisation of posterior expected adjusted Rand index. The adjusted Rand index (ARANDI) can be understood as a measure of similarity between any two data partitions (e.g. estimated clustering $c^\star$ and true c), and Hubert and Arabie (1985) defined it using values from contingency table (see example Table (2.2)),

$$ARI(c_i, c_j) = \frac{\sum_{k,l}\binom{n_{kl}}{2} - \sum_k \binom{a_k}{2}\sum_l \binom{b_l}{2}/\binom{N}{2}}{\frac{1}{2}\left[\sum_k \binom{a_k}{2} + \sum_l \binom{b_l}{2}\right] - \sum_k \binom{a_k}{2}\sum_l \binom{b_l}{2}/\binom{N}{2}},$$

However, in order to identify the overall clustering, Fritsch and Ickstadt (2009) pro-

Table 2.2: Example of contingency table for two clusterings c_i and c_j, where $n_{kl} = |c_i \cap c_j|$

	c_i^1	$\cdots$	c_i^L	sum
c_j^1	n_{11}	$\cdots$	n_{1L}	a_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
c_j^K	n_{K1}	$\cdots$	n_{KL}	a_K
sum	b_1	$\cdots$	b_L	N

posed to use MCMC samples,

$$\frac{1}{M}\sum_{m=1}^{M} ARI(c^\star, c^{(m)}), \qquad \text{here } c^{(m)} \text{ are MCMC samples from posterior,}$$

in order to approximate the posterior expectation $E(ARI(c^\star, c)|y)$, where y is the data, $c^\star$ corresponds to the potential clustering estimate, and c refers to the true clustering (which is unknown in practice). Figure 2.5**B** illustrates data classified into clusters using this approach, more precisely we used R package "*mclust*" in order to post-process our MCMC samples and to evaluate the estimated final clustering.

Dirichlet process mixture models deals with a function estimation problems and these functions are unknown densities; in the following sections we will focus on Bayesian nonparametric techniques for a function estimation in a regression framework.

2.6 Gaussian process regression

Gaussian processes are another Bayesian nonparametric technique that can be applied for regression and classification, which are considered as function approximation problems (MacKay, 1997, 1998; Rasmussen and Williams, 2006). Although modern theory on Gaussian processes gained its popularity due to the work carried out by Rasmussen and Williams (2006), they have been used as a curve fitting technique previously by O'Hagan and Kingman (1978). For nonparametric regression models a Gaussian process (GP) defines a prior distribution over functions. To define a GP it is enough to

specify the mean and covariance function that are the two characteristics of a GP. The covariance function, that is assumed to have a fixed form, such as squared exponential, controls the GP in terms of hyper-parameters. The hyper-parameters can be estimated by maximum likelihood or by sampling from the posterior distribution using MCMC methods (Neal, 1997). More formally a GP is a collection of random variables, any finite numbers of which have a joint Gaussian distribution (Rasmussen and Williams, 2006). It means that the function f is distributed according to a GP if for any finite set of input point $x_1, ..., x_n$, the random variable $[f(x_1), ..., f(x_n)]^T$ has a multivariate Gaussian distribution.

To recover the underlying process from noisy observations a Bayesian nonlinear method, Gaussian process regression (GPR), can be applied. GPR is a nonparametric Bayesian technique that puts the prior directly on functions instead of the parameters of this function. Let $\mathbf{x}$ and $\mathbf{z}$ be continuous n-dimensional real valued vectors; $\mathbf{x} = \{x_1, \ldots, x_n\}$ represents inputs and $\mathbf{z} = \{z_1, \ldots, z_n\}$ corresponds to the outputs. In a regression framework we relate $\mathbf{x}$ and $\mathbf{z}$ through a function, $\mathbf{z} = f(\mathbf{x})$. The observed values of the dependent variable, $\mathbf{z}$, may be related to the independent variables, $f(\mathbf{x})$ by

$$y(x_i) = f(x_i) + \epsilon, \quad i = 1, \ldots, n,$$

where ϵ is a noise term, which for simplicity, is generally assumed to follow a Gaussian distribution, $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. In GPR we place a Gaussian process (GP) (MacKay, 1998) prior over the functions $f(\mathbf{x})$, i.e. $f \sim GP$. In simple terms this means that the nonlinear function f evaluated at a finite number of input points $x_1, \ldots, x_n$ has a multivariate Gaussian distribution with zero mean and there exists a covariance function, K,

$$[f(x_1), ..., f(x_n)]^T \sim \mathcal{N}(0, K(\mathbf{x}, \mathbf{x}')).$$

The covariance function can be chosen to meet e.g. specific criteria imposed by the data considered, or to facilitate computational evaluation. Here we make a generic

and flexible choice and set the covariance function to be a squared exponential with unknown set of parameters $\theta = \{\sigma_g^2, l\}$,

$$K \equiv K_{se}(\mathbf{x}_p, \mathbf{x}_q) = \sigma_g^2 \exp\Big(-\frac{1}{2l}|\mathbf{x}_p - \mathbf{x}_q|^2\Big),$$

where $\mathbf{x}_p$ and $\mathbf{x}_q$ are input vectors. Consequently, $\mathbf{y} = \{y_1, ..., y_n\}^T$ has a normal distribution with zero mean and covariance matrix $C(\theta) = K + \sigma_\epsilon^2 \mathbf{I}$, with $\mathbf{I}$ the identity matrix.

The parameters, θ, need to be inferred from the data; typically this is done by evaluating the log-likelihood function, which is given by

$$L(\theta) = -\frac{1}{2}\log|C(\theta)| - \frac{1}{2}\mathbf{y}^T C(\theta)^{-1}\mathbf{y} - \frac{n}{2}\log 2\pi; \tag{2.13}$$

from this we can, for example, obtain the maximum likelihood estimates of the model parameters, $\hat{\theta}$.

Given the GP prior it is possible to compute the posterior which is also a GP. Under the prior we have for any finite number of input (test) points $x_1^\star, ..., x_r^\star$ the joint (prior) probability distribution

$$[\mathbf{y}, f(x_1^\star), ..., f(x_r^\star)]^T \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} K(\mathbf{x}_p, \mathbf{x}_q) + \sigma_\epsilon^2 \mathbf{I} & K(\mathbf{x}_p, \mathbf{x}_q^\star) \\ K(\mathbf{x}_p^\star, \mathbf{x}_q) & K(\mathbf{x}_p^\star, \mathbf{x}_q^\star) \end{pmatrix}\right).$$

Hence, in order to get the posterior distribution over functions it is necessary to determine a suitable form for the joint prior above. The values $f(x_1^\star), ..., f(x_r^\star)$ of function f conditioned on the outputs $\mathbf{y}$ are also jointly Gaussian distributed according to Rasmussen and Williams (2006),

$$[f(x_1^\star), ..., f(x_r^\star)]^T | \mathbf{y} \sim \mathcal{N}(m_p, K_p), \tag{2.14}$$

where

$$m_p = K(\mathbf{x}_p^\star, \mathbf{x}_q)[K(\mathbf{x}_p, \mathbf{x}_q) + \sigma_\epsilon^2 \mathbf{I}]^{-1}\mathbf{y},$$

and

$$K_p = K(\mathbf{x}_p^\star, \mathbf{x}_q^\star) - K(\mathbf{x}_p^\star, \mathbf{x}_q)[K(\mathbf{x}_p, \mathbf{x}_q) + \sigma_\epsilon^2 \mathbf{I}]K(\mathbf{x}_p, \mathbf{x}_q^\star).$$

As a result, given a GP prior and observations equation (2.14) defines a GP posterior.

2.7 Multiple-output Gaussian process regression

As a flexible nonparametric modelling technique, Gaussian processes are commonly applied in the context of Bayesian regression and classification. Generally, in order to make predictions about a single output variable $\mathbf{y}$ given input data, $\mathbf{x}$, it is necessary to parameterise a covariance function. Learning the parameters guarantees that at a finite number of input points the random function will have a positive definite covariance matrix. However, in some situations it may be desirable to make predictions about multiple output variables simultaneously. One way of dealing with multiple outputs is to model each output variable independently using single GPs; however, this does not capture the dependencies between the output variables and it becomes difficult to specify a valid covariance function that could include cross and auto correlations in a set of related Gaussian processes. An alternative formulation for handling many outputs was introduced by Boyle and Frean (2005), who constructed dependant Gaussian process via multi-output linear filters.

Dealing with linear filters is a central to signal processing where such filters (see Figure 2.6) describe a physical systems that can generate an output signal in response to a given input signal (Haykin and Moher, 2010; Roberts, 2008). Defining the linear filters that satisfy time invariance and linearity requirements, it is easy to see that such filters can be characterised by their kernel function (an impulse response) $h(t)$,

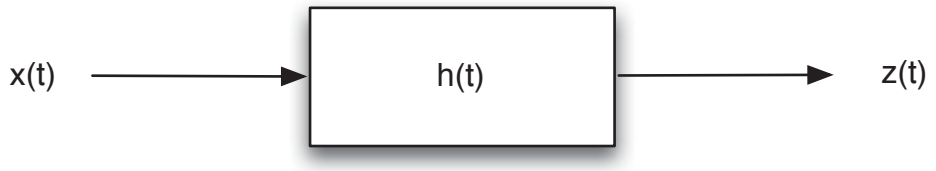


Figure 2.6: Schematic illustration of a single input-output linear filter; here $x(t)$ is an input signal, $z(t)$ - an output signal, and $h(t)$ - a kernel function that is a major characteristic of a filter.

and the output $z(t)$ can be expressed via a convolution integral,

$$z(t) = h(t) \otimes x(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)\mathrm{d}\tau,$$

where the symbol "⊗" denotes the convolution operator. In order to transmit the signal that has a mathematical properties of a GP, the kernel function, $h(t)$ must be absolutely integrable, i.e.

$$\int_{-\infty}^{\infty} |h(t)|\mathrm{d}t < \infty,$$

Then, if the input $X(t)$ is specified to be a Gaussian white noise process, the output process, $Z(t)$, will also be a GP.

Specifying a stable, linear, time invariant filter with M white noise processes as inputs, $X_1(t), \ldots, X_M(t)$, K outputs, $Z_1(t), \ldots, Z_K(t)$, and $M \times K$ impulse responses, results in a dependent GP model (Boyle and Frean, 2004). A multiple-input multiple-output filter can thus be defined as

$$Z_k(t) = \sum_{m=1}^{M} \int_{-\infty}^{\infty} h_{mk}(\tau)X_m(t-\tau)\mathrm{d}\tau,$$

where $h_{mk}(t)$ are kernel functions and $Z_k(t)$ the kth output to the given m imputs. As discussed previously, the observed variables might differ from expected variables due

to the measurement noise and we consider,

$$Y_k(t) = Z_k(t) + W_k(t), \tag{2.15}$$

where $W_k(t)$ is a Gaussian white noise process with variance σ_k^2.

Multiple-input multiple-output filters are able to capture the relationships between several variables $Y_k(t)$; and in the model these kind of dependencies are build in via shared input noise sources that enable the specification of valid covariance functions. For the sake of simplicity, let the impulse response be a Gaussian kernel $h_{mk}(t) = v_{mk}\exp\{-\frac{1}{2}(t-\mu_{mk})^2 A_{mk}\}$. Then evaluating the convolution integral leads to the following cross and auto-covariance function,

$$\begin{aligned} c_{ij}(d) &= \sum_{m=1}^{M}\int_{-\infty}^{\infty} h_{mi}(\tau)h_{mj}(\tau+d)\mathrm{d}\tau \\ &= \sum_{m=1}^{M}\frac{(2\pi)^{\frac{1}{2}}v_{mi}v_{mj}}{\sqrt{A_{mi}+A_{mj}}}\exp\left\{-\frac{1}{2}(d-[\mu_{mi}-\mu_{mj}])^2 S\right\}, \end{aligned} \tag{2.16}$$

where, $S = A_{mi}(A_{mi}+A_{mj})^{-1}A_{mj}$ and $d = t_a - t_b$ is the temporal separation between two input points, (see Boyle and Frean (2004) appendix for derivation and generalisation to multi-dimensions). Constructing intermediate matrices C_{ij}, permits the definition of a positive definite symmetric covariance matrix $\mathbf{C}$ between the K variables,

$$\mathbf{C} = \begin{pmatrix} C_{11}+\sigma_1^2\mathrm{I} & \dots & C_{1K} \\ \dots & \dots & \dots \\ C_{K1} & \dots & C_{KK}+\sigma_K^2\mathrm{I} \end{pmatrix}_{[N\times N]}, \tag{2.17}$$

where $N = \sum_{i=1}^{K} N_i$ is total number of observations, and N_i defines the number of observations of a particular variable i. Having defined such a covariance matrix, we can use the log-likelihood, which has the form (2.13) for the inference of the hyperparameters $\theta = \{v_{mk}, \mu_{mk}, A_{mk}\}$. Again, following a Bayesian framework we can use

the results from the GPR section to evaluate the joint predictive distribution (2.14) for all outputs. Alternatively, for a particular variable i predictions can be made using the appropriate marginal distribution, which is Gaussian, with mean $\mathbf{m}_i(t')$ and variance, $\mathbf{var}_i(t')$, given by

$$\begin{aligned} \mathbf{m}_i(t') &= \mathbf{k}^T\mathbf{C}^{-1}\mathbf{y}, \\ \mathbf{var}_i(t') &= \kappa - \mathbf{k}^T\mathbf{C}^{-1}\mathbf{k}, \end{aligned} \tag{2.18}$$

where

$$\begin{aligned} \kappa &= C_{ii}(0) + \sigma_i^2, \\ \mathbf{k}^T &= [k_1^T, \cdots, k_K^T], \\ k_j^T &= [(C_{ij}(t' - t_{j,1}) \cdots C_{ij}(t' - t_{j,N_j})]. \end{aligned}$$

With multiple-output Gaussian process regression we have introduced the final theoretical concept before turning into applications and results chapters. The subsequent chapter will discuss more thoroughly the applicability of MGPs to model dependencies in gene regulatory systems where we will be imputing the missing experimental measurements.

Chapter 3

Modelling Gene Expression and Imputing the Missing Data with Multiple-output Gaussian Processes

3.1 Introduction

Systems Biology has the ambitious goal of providing a total description of complex biological systems at the cellular level (Karr et al., 2012). By combining mathematical modelling, computational techniques and biological data, it seeks to explain how each individual part contributes to the functioning of the system as a whole. Such systems, even when apparently identical, can exhibit different behaviours among living cells (Spencer and Sorger, 2011; Ingram et al., 2008). Therefore, the ability to identify the origins of such differences between cells can guide us to better understanding of biological processes. This is why gene expression analysis and modelling are of particular importance.

A variety of techniques allow us to analyse cellular data and model biological processes

at a cell level (Aldridge et al., 2009; Kim and Simon, 2014). In order to identify and understand the various regulatory mechanisms that are involved in a particular biological system a range of advanced mathematical and computational methods have been developed in recent decades (Kirk et al., 2012; Yau et al., 2011). An important class of these methods are curve fitting techniques that permit the identification of functional relationships between experimentally observed data. These relationships can be identified by applying parametric or non-parametric techniques, such as parametric regression (Garcia-Etxebarria et al., 2014), splines (Luan and Li, 2003) or Gaussian process regression (Rasmussen and Williams, 2006).

In many cases experimentally obtained data can be incomplete and have missing measurements. This typically would challenge the assessment of the data and could lead to the misinterpretation of the underlying biological question. The problem of missing value imputation is far from new. For this reason various methods have been proposed to deal with incomplete data and current approaches to missing value estimation range from very simple procedures (e.g. excluding samples with missing measurements from further analysis, providing estimates by taking the mean of the observed values for that variable) to complex algorithms and probabilistic models (Weerasinghe, 2010; Rubin, 2004; Oba et al., 2003; Troyanskaya et al., 2001; Hastie et al., 1999). Although literature on missing value imputation is rich (Celton et al., 2010), it is mostly focussed on addressing missing value estimation in gene expression microarray experiments and in data clustering problems. In this chapter we will explore how an advanced statistical modelling technique called Multiple-output Gaussian processes (Boyle and Frean, 2004) can be applied for modelling systems biology data, in particular, gene expression. Moreover, we will inspect the potential applicability of MGPs in order to model the functional relationships in the data and estimate the missing entries. In particular, the applicability of the theory will be demonstrated on popular gene expression examples and experimental gene expression data of the *Hes1* regulatory system.

3.2 Applications

In this section we demonstrate how Gaussian processes can be applied for modelling gene expression data. We consider a generalised scheme for gene expression, and model the relationship between *m*RNA and protein expression levels using dependent GPs. We test this approach on a simplified gene expression example; show how this approach can be used in order to impute the missing data in *p53–Mdm2* system; finally, we apply a dependent GPs to model *Hes1* regulatory system where *m*RNA expression levels were measured experimentally.

3.2.1 Standard model for gene expression

In the standard gene expression model (Ptashne, 1985; Ingram et al., 2008) a transcription factor activates the gene and permits the production of *m*RNA molecules. At this point protein is produced from the *m*RNA; when the transcription factor unbinds from the upstream region of the gene, *m*RNA production stops. Protein production terminates when all *m*RNA molecules have been degraded. This simple *m*RNA and protein production scheme is illustrated in Figure 3.1**A**.

Let us consider three ordinary differential equation (ODE) models that describe the *m*RNA and protein dynamics over time (3.1). The models differ in terms of protein production that affects the overall protein concentration.

$$\text{a)} \begin{cases} \frac{\mathrm{d}m}{\mathrm{d}t} = \alpha m(1-m); \\ \frac{\mathrm{d}p}{\mathrm{d}t} = \beta m - \gamma p; \end{cases} \quad \text{b)} \begin{cases} \frac{\mathrm{d}m}{\mathrm{d}t} = \alpha m(1-m); \\ \frac{\mathrm{d}p}{\mathrm{d}t} = \frac{\beta m}{1+\beta m} - \gamma p; \end{cases} \quad \text{c)} \begin{cases} \frac{\mathrm{d}m}{\mathrm{d}t} = \alpha m(1-m); \\ \frac{\mathrm{d}p}{\mathrm{d}t} = \frac{1}{1+\beta m} - \gamma p. \end{cases} \tag{3.1}$$

Here, m and p denote the concentration levels of *m*RNA and protein respectively; and parameters α, β, and γ represent the production/degradation rate of *m*RNA, protein production rate and protein degradation rate respectively.

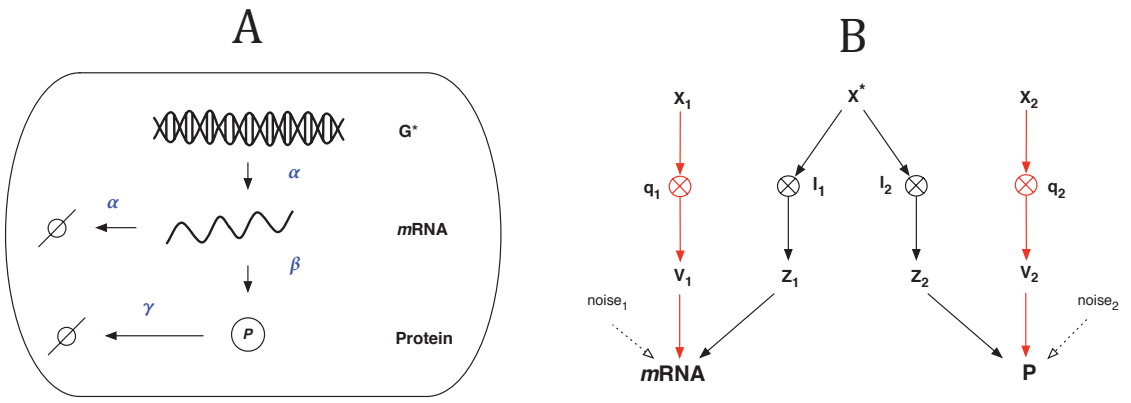


Figure 3.1: A simplified scheme for gene expression. **(A)** $G^\star$ denotes an activated gene; α is a production and degradation rate for mRNA; β is protein P production rate; and γ denotes the degradation rate for protein P. **(B)** Illustration of a graphical representation of a multiple-output Gaussian process model that corresponds to the simplified gene expression scheme; $X_1, X^\star, X_2$ denote the input processes; $q_{1,2}$ and $l_{1,2}$ are Gaussian kernel functions; $V_{1,2}$ and $Z_{1,2}$ are intermediate Gaussian processes, and the symbol $\otimes$ denotes a convolution operation.

We select the parameters $(\alpha, \beta, \gamma) = (0.5, 0.2, 0.25)$ and initial mRNA and protein concentrations to be equal for all three models, $m(0) = 0.1$, $p(0) = 0$. The data sets are generated by simulating each model in (3.1) and recording mRNA and protein concentrations at several time points: $t = (0, 2, 4, 6, 8)$. Furthermore, we add random noise $\epsilon_1 \sim \mathcal{N}(0, 0.1^2)$ and $\epsilon_2 \sim \mathcal{N}(0, 0.05^2)$ to the simulated mRNA and protein data to represent experimentally observable versions of mRNA and protein concentrations (see Figure 3.2 for data trajectories; dashed lines are noise free data; light blue/green dots correspond to the data with added noise term).

Due to the existing biological relationship between mRNA and protein, this relationship should exist and between their time corse data. This means that if a protein is "direct product" of mRNA, their measurable concentrations should be dependent. For this reason, in order to model mRNA and protein data in a nonparametric fashion, we can apply multiple-output GP framework (2.15) (with K=2). Figure 3.1**B** illustrates a graphical representation of dependencies in the mRNA and protein model; here each output can be expressed as a superposition of three Gaussian processes. More explic-

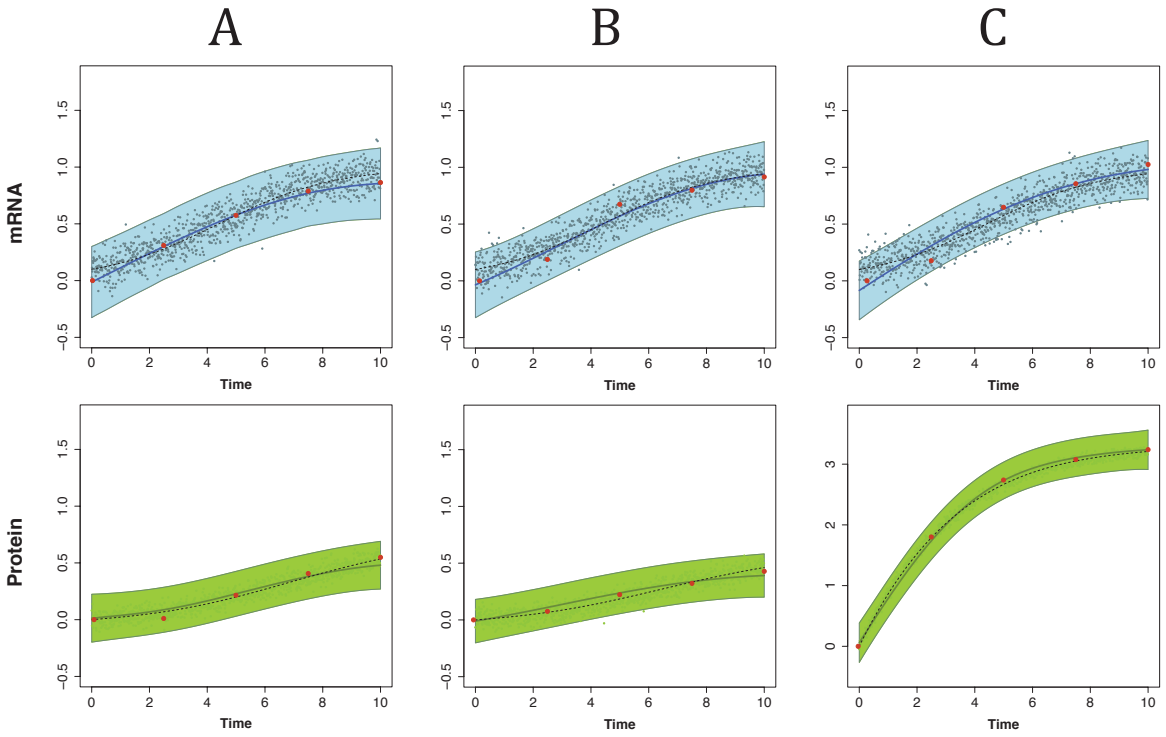


Figure 3.2: Predictions with MGPs model for gene expression data. Red dots are *m*RNA/protein data used to fit MGP model; light blue/green dots are ODE output with added noise. Solid blue and green lines correspond to the mean behaviour of MGP model for *m*RNA and protein respectively. Blue/green shaded areas correspond to the confidence region. **(A)** Predictions for Model (a). **(B)** Predictions for Model (b). **(C)** Predictions for Model (c).

itly, in the model *m*RNA behaviour is captured via two GPs (V_1 and Z_1) and additive noise ($noise_1$). Both GPs are derived via convolution between noise sources, X_1 and $X^\star$, and Gaussian kernel functions, q_1 and l_1, respectively. Similarly, we model protein P, where the expression levels are captured via GPs, V_2 and Z_2, and additive noise ($noise_2$). It is worth noting that dependencies between *m*RNA and protein are included in this model via shared input noise $X^\star$. Next, we set parameters A_i of each Gaussian kernel to be $\exp(f_i)$ and noise variances to $\sigma_1 = \exp(\eta_1)$, $\sigma_2 = \exp(\eta_2)$ leading to a set of hyper-parameters $\theta = (v_i, f_i, \eta_1, \eta_2)$, $i = 1..4$. In order to make predictions using our MGP model we choose the following priors, $v_i \sim \mathcal{N}(0, 2^2)$, $f_{1,2} \sim \mathcal{N}(-4, 2^2)$, $f_{3,4} \sim \mathcal{N}(1, 2^2)$, $\eta_{1,2} \sim \mathcal{N}(-2, 2^2)$, and calculate the maximum a-posteriori estimate using a multi-starting Nelder-Mead optimisation algorithm (Nelder

Parameter	Description
β_x	*p53* production rate
β_y	*p53*–dependent *Mdm2* production rate
α_x	*Mdm2*–independent *p53* degradation rate
α_y	*Mdm2* degradation rate
α_0	*Mdm2* maturation rate
α_k	saturating *p53* degradation rate
k	*p53* threshold for degradation by *Mdm2*

Table 3.1: This table provides the description for all parameters used in *p53–Mdm2* ODE system. For more details see (Geva-Zatorsky et al., 2009).

and Mead, 1965). The resulting predictions are summarised in Figure 3.2. As expected, it can be observed that multiple-output GP approach can provide a good fit to the data.

3.2.2 Imputing the missing data in *p53–Mdm2*

There has been growing interest in studying biological systems that are capable of producing oscillatory behaviour (Silk et al., 2011; Kholodenko, 2006; Kruse and Jülicher, 2005), and *p53–Mdm2* is one of the exemplar systems that has been studied extensively (Purvis et al., 2012; Sun et al., 2011; Geva-Zatorsky et al., 2009; Levine, 1997). In the *p53–Mdm2* system *p53* is known as a tumour suppressor protein and acts as a transcription factor that controls genes associated with the cell cycle. For example, in case of DNA damage *p53* becomes activated and can trigger cell death, cell cycle arrest or DNA repair mechanisms (Maroto and Monk, 2008). The regulation of *p53* levels is achieved via interactions with *Mdm2* protein, where *Mdm2* negatively regulates *p53* by increasing its degradation. Schematically this negative feedback is illustrated in Figure 3.3**A**.

Let us consider a standard ODE model that describes the oscillations in *p53-Mdm2* protein levels (Geva-Zatorsky et al., 2009),

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \beta_x - \alpha_x x - \alpha_k y \frac{x}{x+k}, \\ \frac{\mathrm{d}y_0}{\mathrm{d}t} &= \beta_y x - \alpha_0 y_0, \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= \alpha_0 y_0 - \alpha_y y. \end{aligned} \tag{3.2}$$

In the model x, y_0 and y are nuclear levels of *p53* protein, *Mdm2* precursor and *Mdm2* protein, respectively. For the sake of completeness Table 3.1 contains the description of all parameters used in the model. We simulate the above ODE model with a set of parameter values $(\beta_x, \alpha_x, \alpha_k, \beta_y, \alpha_0, \alpha_y, k) = (0.9, 0, 1.7, 1.1, 0.8, 0.8, 0.0001)$ and initial conditions $(x, y_0, y) = (0.0, 0.1, 0.8)$ (Geva-Zatorsky et al., 2009). Again, to the simulated *p53* and *Mdm2* protein data we add random noise, $\mathcal{N}(0, 0.01^2)$ to imitate experimentally observable versions of both proteins.

In order to demonstrate the applicability and benefits of MGPs model, we select $N_1 = 20$ data points for *p53* and $N_2 = 30$ for *Mdm2* protein. Further, we assume that the data for *p53* is distributed at regular time steps over the interval of time $t = [0, 30]$; and the data for *Mdm2* is recorded at regular time steps over $t = [0, 15] \cup [25, 30]$. The latter assumption in this example can represent the situations, when due to unknown reasons we do not have full access to the information about the subject that we are modelling (e.g. no information about *Mdm2* protein measurements in $t = (15, 25)$). This situation is called a missing data problem (see Figure 3.3**B**,**C**).

Given these noisy measurements we would like to reconstruct the true functional behaviour of *p53* and *Mdm2* proteins over time. For this reason we can employ a MGPs framework for two outputs (2.15) (K=2) (Figure 3.1**B** shows a general scheme for two outputs). By repeating the modelling steps that were described in the previous section, and setting the following priors $v_i \sim \mathcal{N}(0, 2^2)$, $f_i \sim \mathcal{N}(1, 2^2)$, $\mu \sim \mathcal{N}(0.5, 1^2)$, $\eta_j \sim$

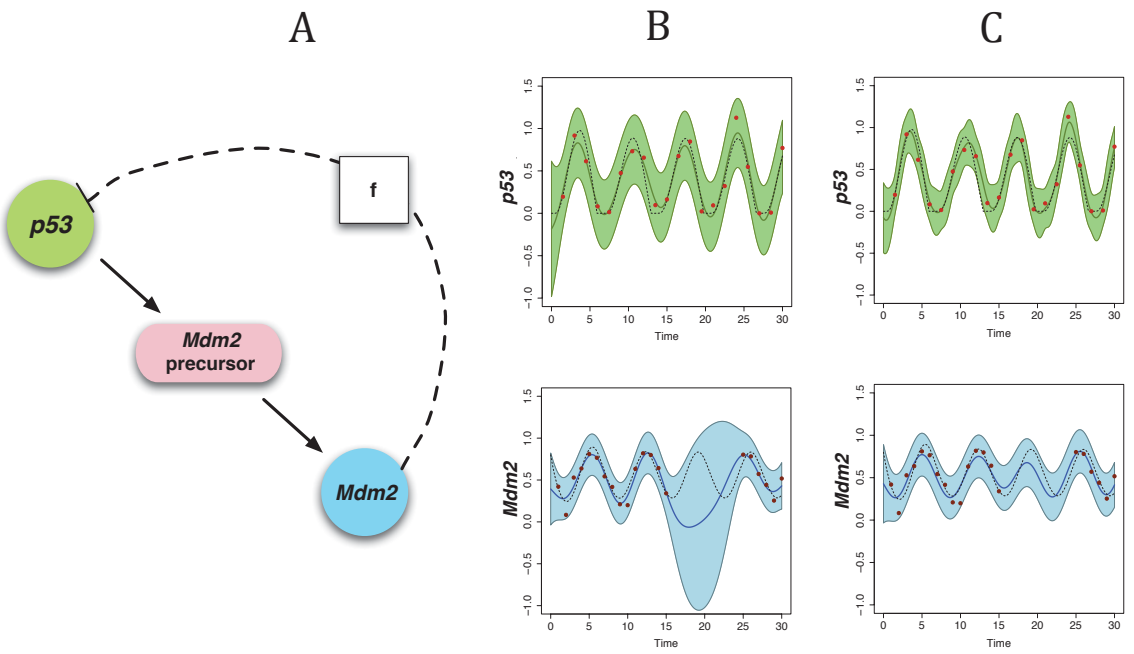


Figure 3.3: *p53–Mdm2* system. **(A)** Schematic illustration of *p53–Mdm2* system: *p53* activates *Mdm2* that in turn negatively regulates *p53* by suspending its activity as a transcription factor and by increasing its degradation rate. The dashed line illustrates a negative regulation. (see Geva-Zatorsky et al., 2009). **(B)** Fitted two independent Gaussian process models for *p53* and *Mdm2* protein data. **(C)** Fitted one multiple-output Gaussian process models for *p53* and *Mdm2* protein data. This model takes in to account dependencies that may exist.

$\mathcal{N}(-2, 2^2)$, we calculate the maximum a-posteriori estimate using a multi-starting Nelder-Mead optimisation algorithm (Nelder and Mead, 1965). The predictions with the MGP model are summarised in Figure 3.3**C**. The solid green/blue lines illustrate the mean behaviour of the MGP model - this is a reconstructed *p53/Mdm2* behaviour that agrees with the true, noise-free protein trajectories (black dashed lines); shaded green/blue areas represent two standard deviations at each prediction point.

To further illustrate the main advantages of using MGPs as a modelling tool let us consider two independent GP models fitted to the same noisy protein data. Figure 3.3**B** summarises the predictions made using the independent Gaussian process models. It is clear that here single Gaussian process models fail in correctly predicting the oscillations in the areas where the data are missing. For this reason modelling the

relationship between *p53* and *Mdm2* can strengthen our predictions in the areas where we have insufficient information.

3.2.3 Applications to *Hes1* regulatory system

In this section we consider the applications of multiple-output Gaussian processes to experimental data. Here we are studying a dynamical system that can exhibit oscillations in the expression levels of the transcription factor *Hes1* (Hirata et al., 2002). The protein *Hes1* is a member of the *Hes* gene family (Kageyama et al., 2007); it contributes to the segmentation of vertebrate embryos and is involved in regulation of cell proliferation and differentiation (Kageyama et al., 2007; Silk et al., 2011). In the *Hes1* system the regulation occurs via a negative feedback loop: the protein shuttles from the cytoplasm into the nucleus and activates the transcription of *m*RNA, which in turn starts the production of *Hes1* protein to be exported to the cytoplasm. A schematic behaviour of this regulatory system is given in Figure 3.4. A simple ODE model describing *Hes1 m*RNA, nuclear and cytoplasmic *Hes1* protein dynamics is summarised below (Silk et al., 2011; Filippi et al., 2013),

$$\begin{aligned}
\frac{\mathrm{d}m}{\mathrm{d}t} &= -k_{deg}m + \frac{1}{1+(\frac{p_2}{k_0})^h}, \\
\frac{\mathrm{d}p_1}{\mathrm{d}t} &= -k_{deg}p_1 + \nu m - k_1 p_1, \\
\frac{\mathrm{d}p_2}{\mathrm{d}t} &= -k_{deg}p_2 + k_1 p_1;
\end{aligned} \tag{3.3}$$

here m denotes the concentration of *m*RNA, p_1 is the concentration of the cytoplasmic protein, and p_2 – the concentration of the nuclear protein. We set the degradation rate k_{deg} for both proteins and *m*RNA to be the same. The remaining ODE parameters are defined as follows: k_0, the quantity of *Hes1* protein in nucleus, k_1, a shuttling rate of *Hes1* protein into the nucleus, h is a Hill coefficient and ν, the *Hes1 m*RNA translation

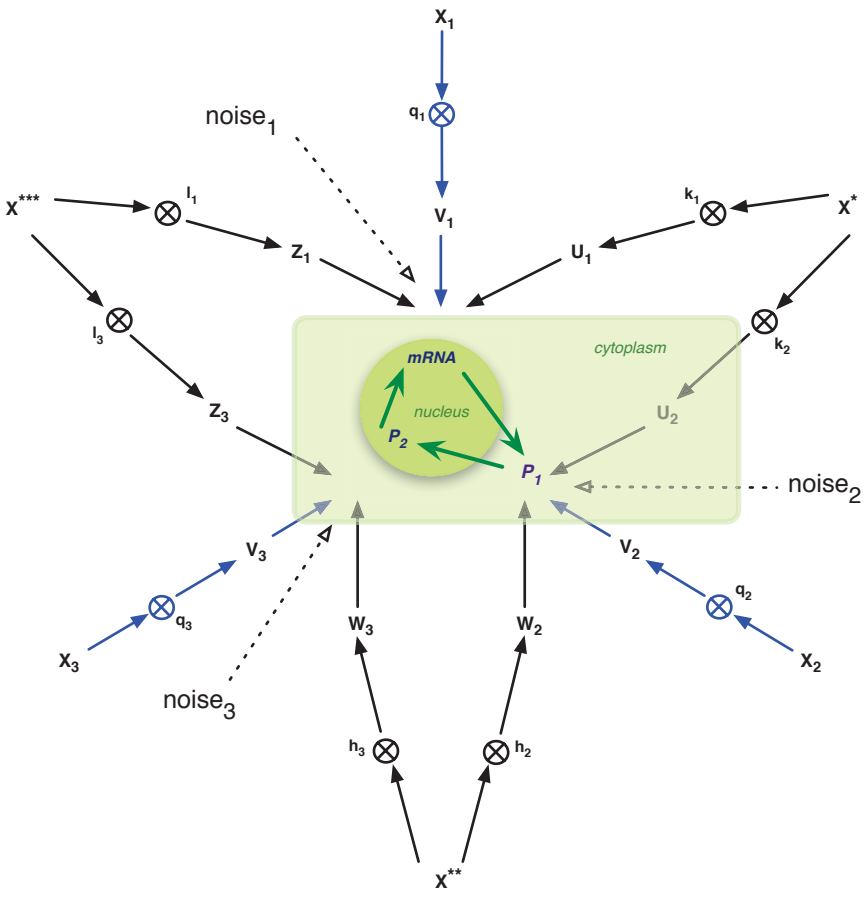


Figure 3.4: *Hes1* regulatory system and corresponding MGPs model. Light green colours illustrate the cell and nucleus; dark green arrows show the regulatory behaviour. On top of a standard systems biology representation of regulatory *Hes1* system a dependent Gaussian process model is constructed to represent the same *Hes1* regulation. Here blue colour illustrates the GPs that are derived from a noise source, and black colour the GPs derived from a noise source that is shared between outputs.

rate.

In order to build a dependent Gaussian process model and test its applicability, we can use the data from Silk et al. (2011). The data consist of *m*RNA measurements taken every half an hour over a 3-hour period. This provides a set of observations for the first output, $D_1 = \{\mathbf{Y}_1, \mathbf{T}\}$, where $\mathbf{Y}_1 = [2, 1.20, 5.90, 4.58, 2.64, 5.38, 6.42, 5.60, 4.48]$ (see Figure 3.5). In order to obtain the cytoplasmic and nuclear protein measurements, we use the parameter values from the literature: $k_{deg} = 0.03$ (Hirata et al., 2002), and $(k_0, \nu, k_1, h) = (2.4, 0.025, 0.1, 1)$ (Filippi et al., 2013), and simulate the ODE

model (3.2) with the following initial conditions $m(0), p_1(0), p_2(0) = (2, 5, 3)$. The trajectories are summarised in Figure (3.5). We record p_1 and p_2 concentrations at regular time points, and add random (i.i.d.) noise $\epsilon_1, \epsilon_2 \sim \mathcal{N}(0, 0.1^2)$ that represents the experimentally observable abundances of cytoplasmic and nuclear protein.

As before, we suppose that there exists a relationship between protein and *m*RNA time courses; this negative feedback regulation can be seen as a cyclic control, where all biological species are interconnected. For this reason there exist dependencies between *m*RNA and nuclear/cytoplasmic protein levels. At this point we can employ the Multiple-output Gaussian process model that describes three interconnected species (2.15)(K=3). In the model we assume that relationships between p_1, p_2 and *m*RNA expression patterns can be captured via shared GPs and these dependencies should be considered in order to reconstruct the original trajectories for protein and *m*RNA. Figure (3.4) shows the graphical representation of MGPs model for the *Hes1* system, where black arrows illustrate model dependencies that exist between species. Precisely, in the model each output can be expressed as a superposition of four Gaussian processes. This means that each output consist of one GP that is derived from an input source that is unique to *m*RNA/p_1/p_2 and an other two GPs, which are derived from shared input sources. The latter two GPs are capturing all possible dependencies between the outputs. The fourth GP is an additive noise. Again for the sake of simplicity, we choose convolution kernels to be Gaussian functions, $v \exp\left(-\frac{1}{2}t^2 A\right)$, $A = \exp(f)$; and noise variances to be $\sigma = \exp(\eta)$. Calculating the maximum a-posterior estimate we can make predictions about original p_1, p_2 and *m*RNA trajectories. The resulting predictions are summarised in Figure 3.5.

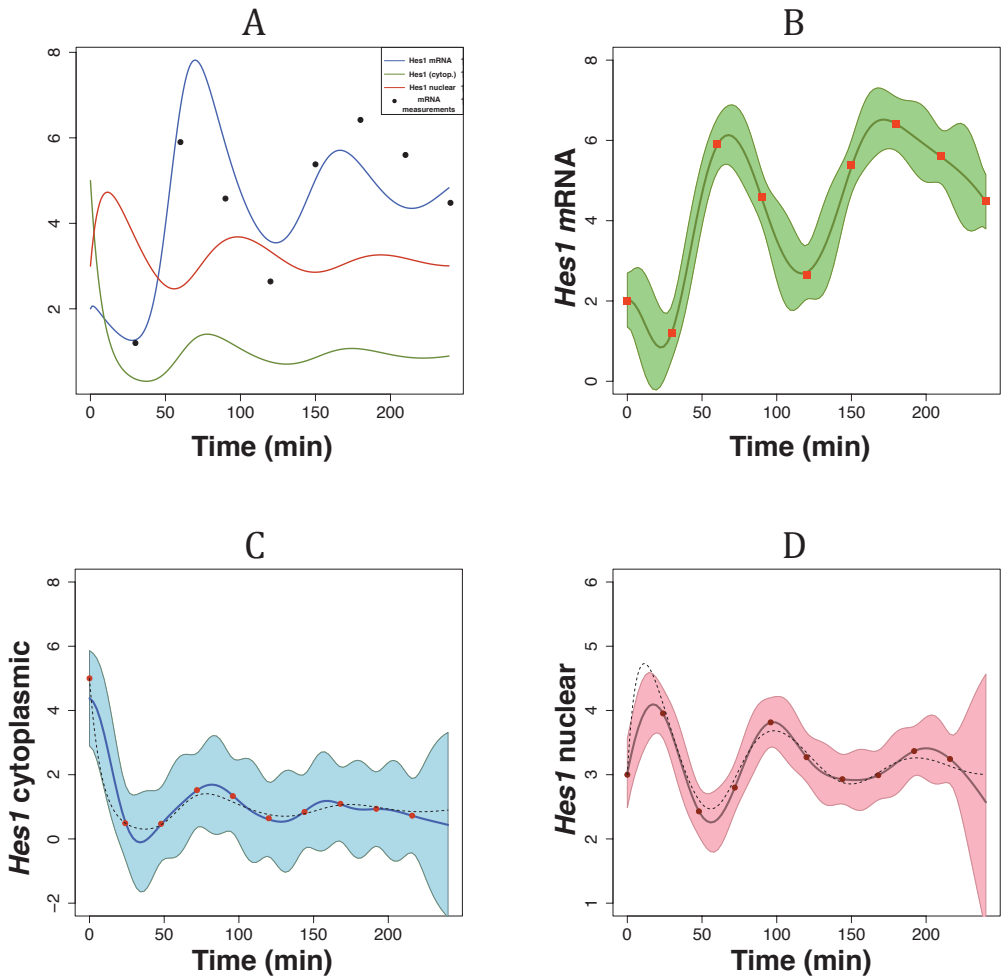


Figure 3.5: **(A)** Simulated trajectories from (3.2) ODE model and experimental measurements of *Hes1* *m*RNA. **(B)-(D)** Predictions with MGPs model: solid lines correspond to the mean of the model, shaded areas are two standard deviations at each prediction point; black dashed lines are noiseless trajectories from the ODE model, red dots are noisy observations of p_1 and p_2, and red squares correspond to the experimental measurements of *m*RNA.

3.3 Conclusions and Discussion

In this chapter we have discussed the applicability of MGPs to model gene expression data. We have explored the approach via linear filters that enable the construction of overall Gaussian processes that takes into account the dependencies and significant features between considered species. This technique is based on convolution integral and parameterises a kernel function $h(t)$; this permits an easier construction of a

positive definite covariance matrix between several outputs. Compared to standard, single GP approach, the ability to consider and model the covariance structure between the species is particularly useful for learning the functional relationship in sparsely sampled areas; and for making a joint predictions about several variables at once. In a biological context sparsely sampled regions occur frequently when dealing with experimentally obtained data that are usually incomplete and have missing entries. Here we have tested this approach on a simple gene expression model and on the *p53-Mdm2* system, as well as on a combination of simulated and experimental data from *Hes1* system.

As with all methods, there exist certain difficulties when applying MGPs approach to model biological systems. For example, the complexity of the model increases together with the number of species it considers. This in turn might lead to a complex log-likelihood model (multimodal) for hyper-parameters, and the inference of them can become a difficult task. For this reason, MGPs might be a less attractive tool for modelling dependencies and dynamics of larger biological systems. Although MGPs are hard to apply for studying for example regulatory networks, they still could be useful when analysing small recurring regulatory patterns called network motifes (Alon, 2007; Milo et al., 2002). Equally, it was previously demonstrated that dependent GPs can be successfully employed to infer the transcription factor activities from *m*RNA concentrations (Lawrence et al., 2006; Gao et al., 2008). In this approach the TF levels were related to *m*RNA levels via linear operator and dependencies (cross/auto covariances) between biological species were also introduced via a convolution integral.

At this point it is a good idea to further explore the applicability of Multiple-output Gaussian processes in different biological contexts, and next chapter will show and discuss how this technique can be extended and employed for predicting metabolic fluxes.

Chapter 4

Derivative Processes for Modelling Metabolic Fluxes

This chapter is based on previously published work – *Derivative processes for modelling metabolic fluxes*. Žurauskienė J., Kirk P., Thorne T., Pinney J., Stumpf M. Bioinformatics. 2014 Jul 1;30(13):1892-8. doi: 10.1093/bioinformatics/btu069. Epub 2014 Feb 26.

4.1 Introduction

All living cells require energy and resources in order to perform vital processes such as repair, replication or movement, and metabolism is a key to cell functioning (Klipp et al., 2005). As a highly organised process metabolism provides all necessary materials by constructing or breaking down complex compounds. One of the most important subjects in metabolic analyses are metabolic fluxes, $v = (v_1, \ldots, v_m)^T$, that correspond to the rates at which molecules, $\mathbf{x} = (x_1, \ldots, x_n)^T$, are turned over by the m reactions. In simple terms this means that the concentration level of a metabolite is a result of

flux activity (Nielsen, 2003).

In general fluxes can be measured experimentally; and while it is possible to obtain experimental data about external fluxes, which tell us how cells are absorbing/secreting various metabolites, it is usually much more interesting to study fluxes that operate inside the cells. For this reason experimental estimates for intracellular fluxes can be obtained by tracking products from isotope-labeled (^{13}C-Metabolic flux analysis) metabolites through the metabolic network (Zamboni, 2011; Blank and Ebert, 2012). Yet, there are a number of difficulties associated with these experiments, e.g. it is hard to design such experiments, the read-out is noisy, and most importantly this approach is restricted to a metabolically steady state analyses and is not appropriate for capturing dynamical flux variations (Noack et al., 2011). Instead theoretical analysis has often progressed by assuming stationarity of the metabolic processes, which in turn allows for characterising the sets of steady-state fluxes under a set of suitable assumptions (Klamt and Stelling, 2003; Voit and Almeida, 2004; Schwartz and Kanehisa, 2006). Flux-balance analysis (Orth et al., 2010) is the most popular example of this strategy, but it becomes questionable once the steady-state assumption can no longer be upheld. Although FBA framework can be augmented using e.g. ODEs and Boolean logic concepts in order to model several biological processes at the same time (Covert et al., 2008), the modelling of time-varying metabolic fluxes is still a challenging task.

Recently proposed methods for the analysis of metabolic pathways use metabolite measurements instead. One such technique is dynamic flux estimation (Goel et al., 2008) that provides estimates of the underlying fluxes in a point-wise fashion at discrete time-points. However, it fails to capture the complete temporal behaviour and in addition it becomes necessary to consider parametric models to explain time variations. Equally, it becomes unclear how to address the noise that might be present in a set of experimentally measured metabolites. Because parametric approaches can offer potentially incorrect representations of the underlying fluxes (Voit, 2013) this

chapter provides a new framework for modelling metabolic fluxes and their dynamics using MGPs. The strengths of GP models arise from their nonparametric nature, which enables us to put priors directly on a function rather than on the parameters of a parametric function. As it was demonstrated in previous chapter, with multiple-output GPs, the single GP framework can be extended to handle many outputs, enabling us to model the relationships between metabolic species. Here, we develop a more general framework that employs so-called derivative GPs (Solak et al., 2003), which allow us to link metabolite abundances, $\mathbf{x}$ (or concentrations) and fluxes v. This in turn enables us to treat also time-course data on metabolites and monitor the changes that occur in fluxes, e.g. over the course of physiological responses, such as to changes in the environment (Bryant et al., 2013).

4.2 Derivative processes

In this section we propose a novel approach to modelling metabolic fluxes: derivative processes that are based on Multiple-output Gaussian processes, which are a flexible nonparametric Bayesian modelling technique. For a Gaussian process that is derived from a linear filter approach, $Y(t) = h(t) \otimes X(t) + W(t)$, where $X(t)$ is a Gaussian process, $h(t)$ is a kernel function and $W(t)$ is an additive noise, it is easy to formulate the expression of a derivative process. Taking a derivative of Y with respect to t it is possible to obtain a new process U that is a Gaussian process as well (Boyle, 2007),

$$U(t) \equiv \frac{\mathrm{d}}{\mathrm{d}t} Y(t) = \int_{-\infty}^{\infty} \Big\{ \frac{\mathrm{d}}{\mathrm{d}t} h(t-\tau) \Big\} X(\tau) \mathrm{d}\tau = g(t) \otimes X(t),$$

This means, that it is possible to construct a derivative process by convolving a white noise Gaussian process $X(t)$ with a derivative kernel function $g(t)$. The above definition enables to consider the derivative processes and the corresponding original

processes as a collection of dependent GPs. This is true because both, the derivative processes and original processes, are derived from exactly the same input $X(t)$.

Specifically, with the aim of constructing a dependent model for a several related variables $\mathrm{Y} = (Y_1, ..., Y_K)$ and their derivatives $\mathrm{U} = (U_1, ..., U_K)$ it is necessary to define a genuine covariance structure, which in principal arises from the initial covariance function (2.16). For example, for a set of four dependent outputs (two original and two derivative processes) the following equations can be applied to compute the covariances (Girard, 2004; Solak et al., 2003; Kirk, 2011),

- Auto-Covariance function of derivative process U_i

$$DDC_{ii}(d) \equiv \mathrm{cov}\left(\left.\frac{\mathrm{d}Y_i}{\mathrm{d}t}\right|_{t=t_a}, \left.\frac{\mathrm{d}Y_i}{\mathrm{d}t}\right|_{t=t_b} \right) = \frac{\mathrm{d}^2}{\mathrm{d}t_a \mathrm{d}t_b} c_{ii}(d);$$

- Cross-Covariance function between two derivative processes U_i and U_j

$$DDC_{ij}(d) \equiv \mathrm{cov}\left(\left.\frac{\mathrm{d}Y_i}{\mathrm{d}t}\right|_{t=t_a}, \left.\frac{\mathrm{d}Y_j}{\mathrm{d}t}\right|_{t=t_b} \right) = \frac{\mathrm{d}^2}{\mathrm{d}t_a \mathrm{d}t_b} c_{ij}(d);$$

- Covariance between original process Y_i and corresponding derivative process U_i

$$DC_{ii}(d) \equiv \mathrm{cov}\left(Y_i, \left.\frac{\mathrm{d}Y_i}{\mathrm{d}t}\right|_{t=t_b} \right) = \frac{\mathrm{d}}{\mathrm{d}t_b} c_{ii}(d);$$

- Covariance between original process Y_i and derivative process U_j

$$DC_{ij}(d) \equiv \mathrm{cov}\left(Y_i, \left.\frac{\mathrm{d}Y_j}{\mathrm{d}t}\right|_{t=t_b} \right) = \frac{\mathrm{d}}{\mathrm{d}t_b} c_{ij}(d).$$

Let $\mathbf{Z} = (Z_1, Z_2)$ denote function values that correspond to test input points for both

outputs Y_1 and Y_2; in addition let $\mathbf{R}$ denote a block matrix,

$$\mathbf{R} = \begin{pmatrix} C_{11} & C_{12} & DC_{11} & DC_{12} \\ C_{21} & C_{22} & DC_{21} & DC_{22} \end{pmatrix}, \quad \mathbf{L} = \mathbf{R}^T,$$

which describes the correlations between observations $\mathbf{Y} = (Y_1, Y_2)$ and their "function" values, $\mathbf{Z} = (Z_1, Z_2)$, and corresponding derivative variables $\mathbf{U} = (U_1, U_2)$ evaluated at any finite number of test points $t_1, ..., t_r$. In a similar fashion, we define another matrix $\mathbf{H}$,

$$\mathbf{H} = \begin{pmatrix} \tilde{C}_{11} & \tilde{C}_{12} & \tilde{DC}_{11} & \tilde{DC}_{12} \\ \tilde{C}_{21} & \tilde{C}_{22} & \tilde{DC}_{21} & \tilde{DC}_{22} \\ \tilde{DC}_{11} & \tilde{DC}_{12} & DDC_{11} & DDC_{12} \\ \tilde{DC}_{21} & \tilde{DC}_{22} & DDC_{12} & DDC_{21}, \end{pmatrix},$$

where the $\tilde{C}_{ij}$ matrices contain the correlations between functions Z_1 and Z_2 evaluated at any finite number of test points $t_1, ..., t_r$; $\tilde{DC}_{ij}$, the correlations between functions $\mathbf{Z} = (Z_1, Z_2)$ and derivative variables $\mathbf{U} = (U_1, U_2)$ evaluated at the same input test points; and finally DDC_{ij} matrices that consist of auto/cross correlations between derivative variables U_1 and U_2. The above $\mathbf{R}$, $\mathbf{L}$ and $\mathbf{H}$ matrices are building components of overall covariance matrix $\mathbf{K}$, which is symmetric and positive definite,

$$\mathbf{K} = \begin{pmatrix} \mathbf{C} + \sigma^2 \mathrm{I} & \mathbf{R} \\ \mathbf{L} & \mathbf{H} \end{pmatrix}.$$

At a finite number of input points $t_1, ..., t_r$, such matrix $\mathbf{K}$ allows us to place a joint prior over observations $\mathbf{Y}$, functions $\mathbf{Z}$ and derivatives $\mathbf{U}$,

$$[\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Z}_1, \mathbf{Z}_2, \mathbf{U}_1, \mathbf{U}_2] \sim \mathcal{N}(\mathbf{0}, \mathbf{K}).$$

Evaluating a GP posterior

$$[\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{U}_1, \mathbf{U}_2] \Big| [\mathbf{Y}_1, \mathbf{Y}_2] \sim \mathcal{N}(\mathbf{m}_{post}, \mathbf{K}_{post}), \tag{4.1}$$

where

$$\mathbf{m}_{post} = \mathbf{L}\left[C + \sigma^2 \mathrm{I}\right]^{-1} \mathbf{R}, \quad \text{and} \quad \mathbf{K}_{post} = \mathbf{H} - \mathbf{L}\left[C + \sigma^2 \mathrm{I}\right]^{-1} \mathbf{Y},$$

enables us to make joint predictions about original processes and derivative processes at the same time. Alternatively, if there is no necessity to sample from the posterior process, we can use marginal Gaussian distributions to make the predictions about each output variable, and in turn facilitate the computational evaluations. Therefore, the following set of equations are marginal distributions for output i and its derivative process i at any input point $t^\star$, with $\mathbf{m}_{Y_i}$ being a mean of the original process, $\mathbf{m}_{U_i}$ - mean of the derivative process, $\mathbf{var}_{Y_i}$ a variance of the original process and $\mathbf{var}_{U_i}$ a variance of the derivative process,

$$\begin{aligned}
\mathbf{m}_{Y_i}(t^\star) &= \mathbf{k}_{Y_i}\left[C + \sigma^2 \mathrm{I}\right]^{-1} \mathbf{Y}, \\
\mathbf{m}_{U_i}(t^\star) &= \mathbf{k}_{Z_i}\left[C + \sigma^2 \mathrm{I}\right]^{-1} \mathbf{Y}, \\
\mathbf{var}_{Y_i}(t^\star) &= \kappa - \mathbf{k}_{Y_i}\left[C + \sigma^2 \mathrm{I}\right]^{-1} \mathbf{k}_{Y_i}^T, \\
\mathbf{var}_{U_i}(t^\star) &= \eta - \mathbf{k}_{Z_i}\left[C + \sigma^2 \mathrm{I}\right]^{-1} \mathbf{k}_{Z_i}^T,
\end{aligned} \tag{4.2}$$

where

$$\kappa = c_{ii}(0) + \sigma_i^2, \qquad \eta = DDC_{ii}(0),$$

$$\mathbf{k}_{Y_i} = \begin{pmatrix} C_{i1}(t^\star - t_{1,1}) \\ \cdots \\ C_{i1}(t^\star - t_{1,N_1}) \\ C_{i2}(t^\star - t_{2,1}) \\ \cdots \\ C_{i2}(t^\star - t_{2,N_2}) \end{pmatrix}, \quad \mathbf{k}_{U_i} = \begin{pmatrix} DC_{i1}(t^\star - t_{1,1}) \\ \cdots \\ DC_{i1}(t^\star - t_{1,N_1}) \\ DC_{i2}(t^\star - t_{2,1}) \\ \cdots \\ DC_{i2}(t^\star - t_{2,N_2}) \end{pmatrix}.$$

Here we described how predictions with posterior process can be done for two original processes and corresponding two derivative processes, but equations (4.1), (4.2) can be easily extended in order to make prediction about K variables instead.

4.3 Applications and results

In order to demonstrate the performance of derivative processes, we consider simple examples: a system of two oscillating signals, Lotka–Volterra *predator–prey* model, model of linear and branched metabolic pathways; and finally we demonstrate the applicability of derivative processes on experimental data of nitrogen metabolism in *Escherichia coli*.

4.3.1 Oscillating signals

A simple oscillating signal can be expressed in the following way, $z(t) = A\sin(\omega t + \phi)$, where A is an amplitude, $\omega = 2\pi f$ an angular frequency, and ϕ the phase. It is a particularly useful illustrative model because it is easy to evaluate the performance of derivative processes as derivative signals have a known functional form. Let us

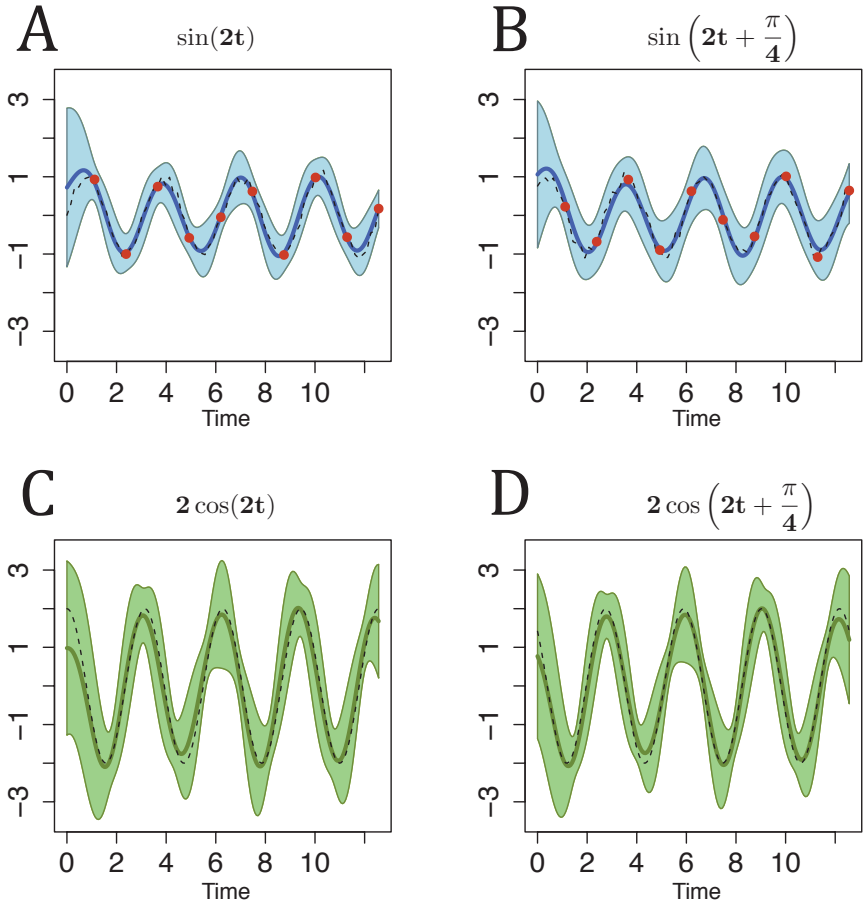


Figure 4.1: This figure is adapted from Žurauskienė et al. (2014a). Predictions with MGPs model for two oscillating signals. **(A)-(B)** Dashed lines represent true behaviour of noiseless $sin(\cdot)$ trajectories; red dots correspond to the noisy observations for both signals (data); solid blue lines are the mean behaviour of the MGPs model (predictions with original GPs); light blue areas correspond to two standard deviations at each prediction point. **(C)–(D)** Dashed lines represent true behaviour of noiseless $cos(\cdot)$ trajectories; solid green lines are the mean behaviour of the MGPs model (predictions with derivative processes); light green areas correspond to two standard deviations at each prediction point.

consider a simple system that consist of two oscillating signals, $z_1(t)$ and $z_2(t)$,

$$\begin{aligned} z_1(t) &= sin(2t), \quad \Longrightarrow \quad z_1'(t) = 2cos(2t), \\ z_2(t) &= sin\Big(2t + \frac{\pi}{4}\Big), \quad \Longrightarrow \quad z_2'(t) = 2cos\Big(2t + \frac{\pi}{4}\Big), \end{aligned}$$

where $t \in [0, 4\pi]$. To imitate a real experimental measurements we added a random noise to the simulated trajectories, $Y_1(t) = z_1(t) + \epsilon_1$, $Y_2(t) = z_2(t) + \epsilon_2$, where $\epsilon_i \sim \mathcal{N}(0, 0.1^2)$; and made some observations of both signals at regular time intervals - $D_1 = \{t_{1,i}, Y_{1,i}\}_{i=1}^{N1=10}$ and $D_2 = \{t_{2,j}, Y_{2,j}\}_{j=1}^{N2=10}$. In order to build a single model that would capture a relationship between both signals, we applied two dependent GPs framework (2.15) (K=2) on a combined dataset $\mathbf{D} = \{D_1, D_2\}$, where each signal can be expressed as a superposition of three Gaussian processes — two of which are being constructed via convolution between a noise source and a Gaussian kernel and the third one is an additive noise. We set parameters A_i of each Gaussian kernel to be $\exp(f_i)$ and noise levels to $\sigma_1 = \exp(\eta_1)$, $\sigma_2 = \exp(\eta_2)$, leading to a set of hyper-parameters $\theta = (\nu_i, f_i, \mu_1, \mu_2, \eta_1, \eta_2)$, $i = 1, \ldots, 4$. To build the model the following priors were chosen: $\nu_i, f_i \sim \mathcal{N}(1, 2^2)$, $\eta_j \sim \mathcal{N}(-2, 2^2)$ and $\mu_j \sim \mathcal{N}(0.5, 1^2)$, $j = 1, 2$; and the maximum a posteriori (MAP) estimate $\hat{\theta}$ was calculated using a multi-starting Nelder-Mead optimisation algorithm (Nelder and Mead, 1965). Dependent GPs posteriors (4.1) allow us to make joint predictions about both signals and their derivative processes at any finite number of input points, and the resulting posterior processes are summarised in Figures (4.1). From these posterior processes it can be seen that the mean behaviour of our model agrees with trajectories of underlying noiseless signals, and in order to make predictions about derivative processes it is enough to consider only samples from the original sinusoidal trajectories.

4.3.2 Lotka–Volterra *predator–prey* model

Below we consider a general Lotka–Volterra model. The following ordinary differential equations describe the dynamics of two interacting species, namely prey, x, and

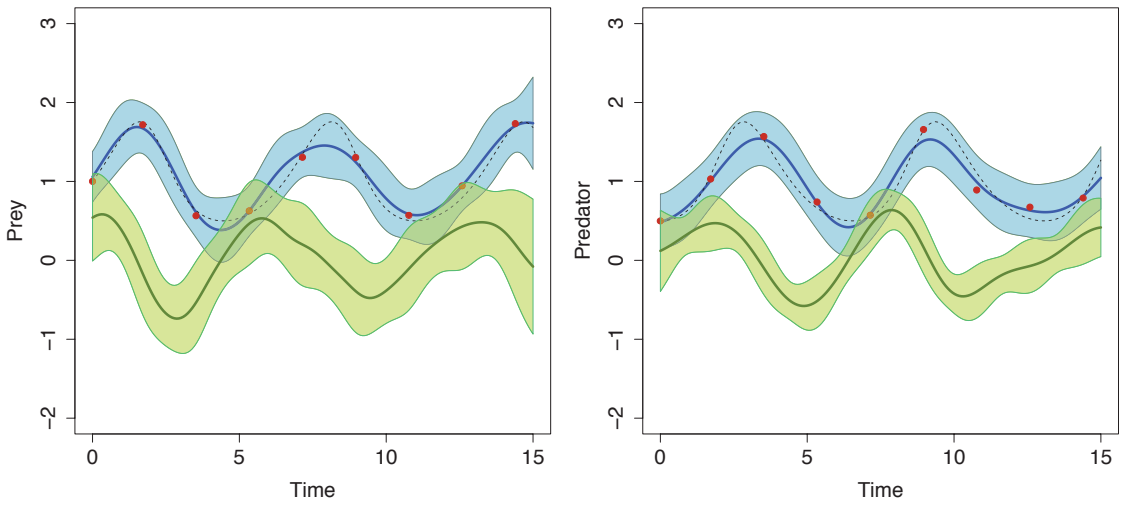


Figure 4.2: Dependent Gaussian processes for Lotka–Volterra predator–prey model. Dashed lines represent a single simulation from ODE model. Red points illustrates noisy observations of each species - prey and predator. Solid blue lines correspond to the mean behaviour of dependent GPs model, solid green lines – to the derivative behaviour of both species and shaded areas correspond to two standard deviations at each prediction point.

predator, y,

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \alpha x - xy, \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= \beta xy - y. \end{aligned}$$

In order to apply the derivative process approach, we simulated the following model with parameter values $\alpha = \beta = 1$ and initial conditions $x_0 = 1,\ y_0 = 0.5$. The dataset consists of selected data points from simulated trajectories with added Gaussian noise $\mathcal{N}(0, 0.1^2)$. Again, we combine the “noisy" measurements, and fit previously discussed two dependent GPs model in order to make predictions about original trajectories and their derivatives. Figure (4.2) illustrates the predictions with posterior processes, solid blue lines correspond to the mean behaviour of the model, dashed lines to the original prey and predator trajectories and solid green lines – to their derivatives respectively.

As was shown above, the derivative processes can be applied in order to make predictions about derivative behaviour of the observed variables. In the subsequent section we will show how this technique can be beneficial for modelling several related fluxes from experimentally obtained measurements of metabolites. The derivative processes can be employed to address a flux estimation problem from time course data. Here, the original GPs describe the dynamics of metabolites and derivative processes capture the functional forms of corresponding fluxes.

4.3.3 Linear pathway

Now let us consider a linear metabolic pathway with two regulatory signals (see Goel et al. (2008) supplementary material for details), which is summarised in Figure (4.3**A**). Here the flow from x_1 to x_2 is negatively regulated by metabolite x_3; and x_3 positively results the transformations of x_2 into x_3. A set of ordinary differential equations can be used to describe the dynamics of these two metabolites, x_2 and x_3, (x_1 is the constant external input),

$$\begin{aligned} \frac{\mathrm{d}x_2}{\mathrm{d}t} &= \frac{x_1 V_{max}}{K_m(1+\frac{x_3}{K_i})+x_1} - x_2^{0.5}x_3, \\ \frac{\mathrm{d}x_3}{\mathrm{d}t} &= x_2^{0.5}x_3 - x_3^{0.5}. \end{aligned} \tag{4.3}$$

In order to apply the derivative process approach, we simulated the ODE model with the following parameter values $(V_{max}, K_m, K_i) = (18.6819, 9.7821, 0.5992)$ and initial conditions $x_2(0) = 1$, $x_3(0) = 1$. In this model the concentration of x_1 is assumed to be constant over time and equal to 2. The dataset consists of selected points from simulated trajectories with added Gaussian noise, $\mathcal{N}(0, 0.05^2)$. Again we combine the "noisy" measurements and fit the dependent GP model in order to make predictions about the original trajectories and their derivatives. In order to obtain functional expressions for fluxes v_1 and v_2 we need to estimate dynamical variations of metabolic,

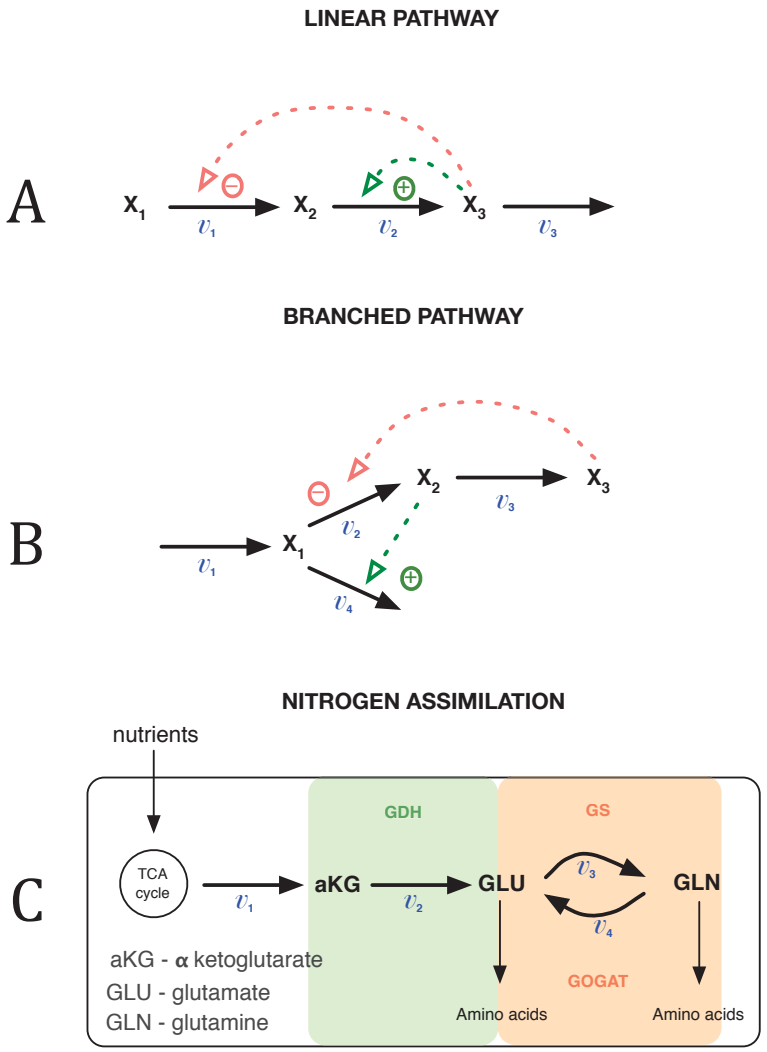


Figure 4.3: This figure is adapted from Žurauskienė et al. (2014a). Pathway information. **(A)** A simple linear metabolic pathway; red and green dashed lines correspond to the inhibition and activation signals. **(B)** Illustrates a branched pathway with positive (green) and negative (red) regulatory signals. **(C)** Illustrates a metabolic pathway in *E. coli*, here v_i, $i = 1...4$ denote the fluxes; αKG, GLU, and GLN correspond to the metabolites; TCA is a short notation for the citrate cycle in *E. coli*. In green it is depicted glutamate dehydrogenase (GDH) part, and in orange - glutamine synthetase-glutamate synthase(GS-GOGAT) part.

x_2, x_3, derivatives. The derivative processes provide the predictions for the left side of equation (4.3) at any finite number of time points, whereas the original GPs describe the solution on the same ODE (4.3). This enable us to link the metabolite measurements to metabolic fluxes. Figure (4.4) illustrates the predictions with posterior processes, where solid blue lines correspond to the mean behaviour of the model,

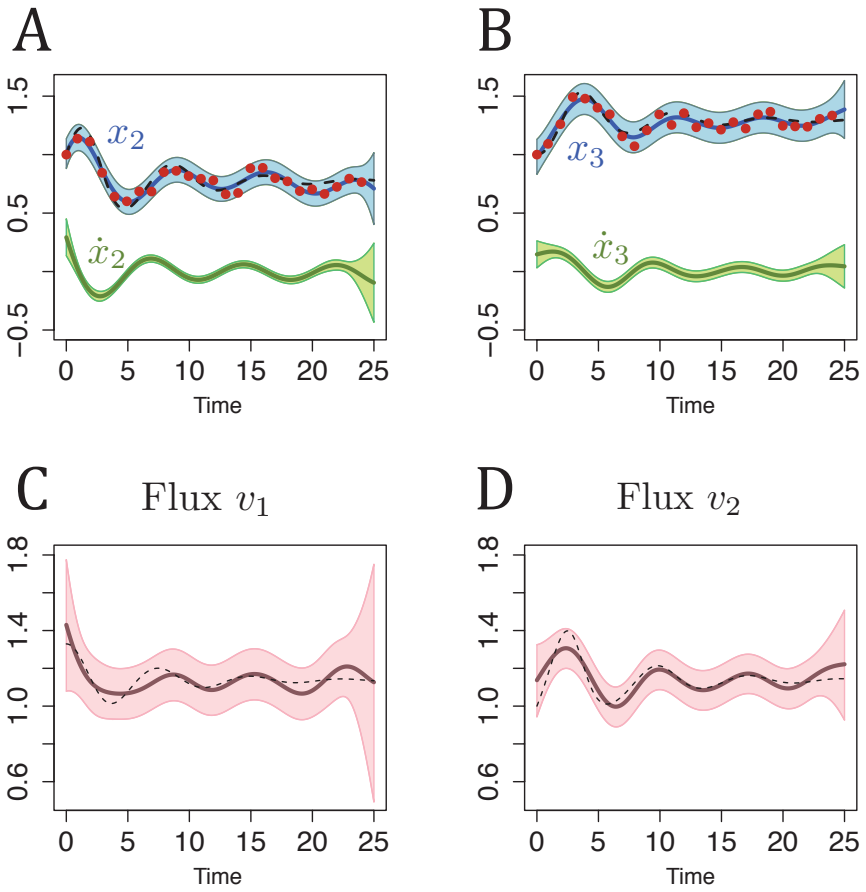


Figure 4.4: This figure is adapted from Žurauskienė et al. (2014a). Predictions with MGPs model for linear metabolic pathway. **(A)–(B)** Dashed lines represent a simulated x_2 and x_3 trajectories from ODE model; red dots correspond to the sparse noisy observations for x_2 and x_3 (data); solid blue/green lines are the mean behaviour of the MGPs model (blue – predictions with original GPs; green – predictions with derivative process); light blue/green areas correspond to two standard deviations at each prediction point. **(C)–(D)** Dark pink lines are predicted fluxes; light pink areas correspond to the confidence region; and dashed lines represent true behaviour of noise-free fluxes v_1 and v_2 (calculated from ODE system).

dashed lines to the original x_2 and x_3 trajectories and solid green lines to their derivatives. In addition, if we assume that we are able to measure flux $v_3 = x_3^{0.5}$, we can obtain the functional expressions for fluxes v_1 and v_2 that are summarised in Figure (4.4**C**,**D**). The dark pink lines illustrate predicted fluxes from noisy metabolite measurements; dashed lines are real fluxes (calculated from ODEs (4.3)); and light pink area corresponds to the confidence region.

4.3.4 Branched pathway

We now turn to an example of metabolic pathway that was originally proposed by Voit, 2013 (see *Example of actual characterization*); Figure (4.3**B**) illustrates a schematic representation of a branched pathway with two regulatory responses, where x_3 inhibits the conversions of x_1 into x_2; and x_2 positively regulates reaction v_4. The following ODE model describes the dynamics of the metabolites that are involved in this pathway,

$$\begin{aligned}
\frac{dx_1}{dt} &= 0.05 - 1.1x_1^{0.5}x_3^{-0.75} - 2.8x_1^{0.8}x_2^{0.4}, \\
\frac{dx_2}{dt} &= 1.1x_1^{0.5}x_3^{-0.75} - 1.1x_2^{0.6}, \\
\frac{dx_3}{dt} &= 1.1x_2^{0.6},
\end{aligned} \tag{4.4}$$

where x_1, x_2, x_3 denote the metabolites. For a given pathway (Figure 4.3**B**), the change in metabolite concentration can be described by the differences between incoming and outgoing fluxes. For this reason we are able to obtain the following expressions for fluxes v_1, v_2, v_3 and v_4,

$$\begin{aligned}
&\frac{dx_1}{dt} = v_1 - v_2 - v_4, && v_1 - v_4 = \frac{dx_1}{dt} + v_2, \\
&\frac{dx_2}{dt} = v_2 - v_3, && \Longrightarrow \quad v_2 = \frac{dx_2}{dt} + \frac{dx_3}{dt}, \\
&\frac{dx_3}{dt} = v_3, && v_3 = \frac{dx_3}{dt}.
\end{aligned} \tag{4.5}$$

These expressions define a system of linear equations that is under-determined as we have more fluxes to estimate than available equations, and it cannot be solved using standard Gaussian elimination techniques. For this reason additional information is required in order to uniquely determine fluxes v_1 and v_4. In this example we will focus only on estimation of fluxes v_2 and v_3 from available data rather than try to address a

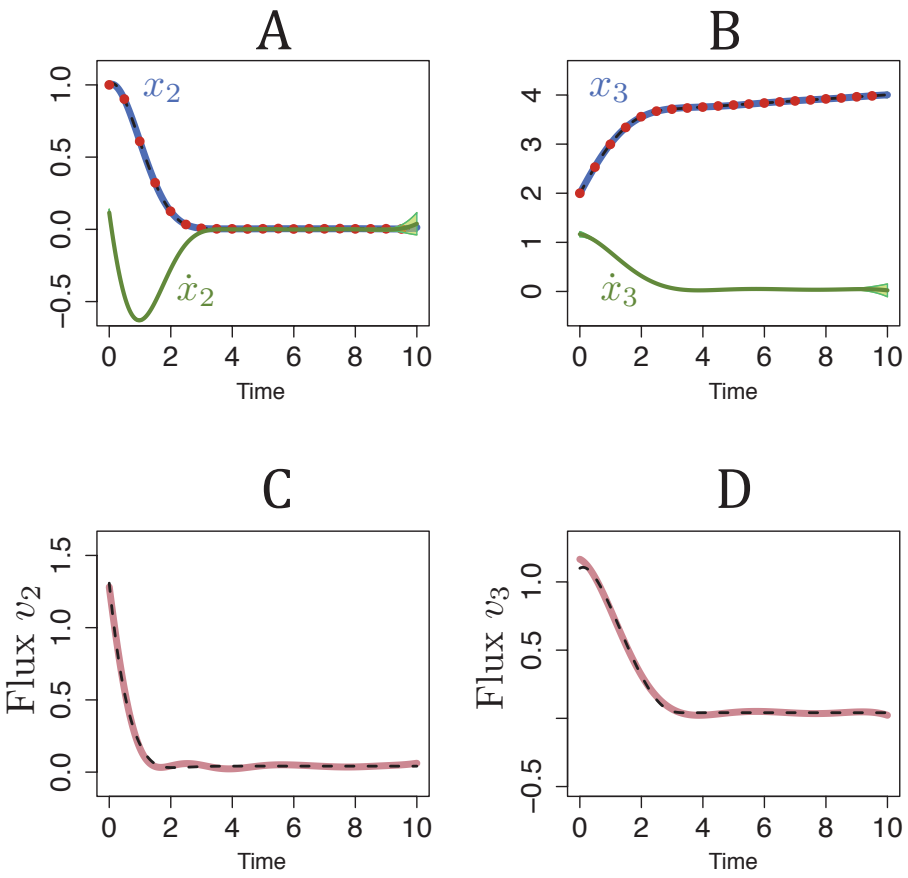


Figure 4.5: This figure is adapted from Žurauskienė et al. (2014a). Derivative processes for branched pathway. **(A-B)** Solid blue and green lines illustrate the mean behaviour of a dependent GPs model; red dots represent the measurements of metabolites x_2 and x_3, and pink lines in the bottom two figures **(C-D)** illustrate the estimated fluxes v_2 and v_3. Here dashed lines correspond to the output generated from ODE model (4.4).

uniqueness problem of v_1 and v_4.

The above ODE model enables us to generate simulated time course data using the initial conditions $x_1(0) = 4$, $x_2(0) = 1$, $x_3(0) = 2$. Next, we apply the dependent GP framework (2.15) (K=2) on the combined dataset $\mathbf{D} = \{D_1, D_2\}$, where $D_1 = \{t_{2,i}, x_{2,i}\}_{i=1}^{N1=20}$ and $D_2 = \{t_{3,i}, x_{3,i}\}_{i=1}^{N1=20}$ contains the measurements of metabolites x_2 and x_3 with added random Gaussian noise $\mathcal{N}(0, 0.01^2)$ (we chose a low noise level so that predictions with derivative processes could be easily compared to the

original fluxes in Voit, 2013 example). For a set of model hyper-parameters $\theta = (v_i, f_i, \eta_1, \eta_2, \mu)$, $i = 1, \ldots, 4$ we use the following priors, $v_i \sim (2, 2^2)$, $f_i \sim (-3, 2^2)$, $\eta_j \sim \mathcal{N}(-2, 2^2)$, $j = 1, 2$ and $\mu \sim \mathcal{N}(0.5, 1^2)$, and calculate the MAP estimate $\hat{\theta}$ as before. Figure 4.5 illustrates the predictions with posterior processes using equations (4.2); **A**-**B** graphs summarise metabolite data. The dark blue lines correspond to the mean behaviour of the original GPs and agrees well with simulated x_2 and x_3 dynamics; the green lines describe the derivatives of the same metabolites and can be understood as a slope estimates. In Figure 4.5**C**-**D** dark pink lines illustrate the predicted metabolic fluxes v_2 and v_3 under consideration of pathway Figure 4.3**B**. From ODE model (4.4) we can calculate original fluxes over the time (in real situations this would not be possible). Figure 4.5**C**-**D** shows good agreement between predicted and original fluxes.

4.3.5 *Escherichia coli* nitrogen assimilation

Finally we apply our technique to experimental data from *E. coli*, where we have measurements of the abundances of several key metabolites involved in the nitrogen-assimilation network. Nitrogen is a key chemical element that acts as a nutrient for the cells; and ammonium is the preferred source of nitrogen for *E. coli* growth (Schumacher et al., 2013; van Heeswijk et al., 2013). In *E. coli*, ammonium can be absorbed via two pathways: GDH (glutamate dehydrogenase) – that operates during cell growth in rich-ammonium environment; and GS-GOGAT(glutamine synthetase-glutamate synthase) – that operates during cell growth in low-ammonium conditions (van Heeswijk et al., 2013). Here, we are focussing on experimental conditions where after a period of nitrogen starvation the bacterial cultures are spiked with the ammonium (Schumacher et al., 2013); Figure 4.6**A** shows experimentally obtained measurements (red dots) for *α-ketoglutarade* (αKG), *glutamate* (GLU) and *glutamine* (GLN) metabolites over the time after ammonium spike; left figure corresponds to a wild

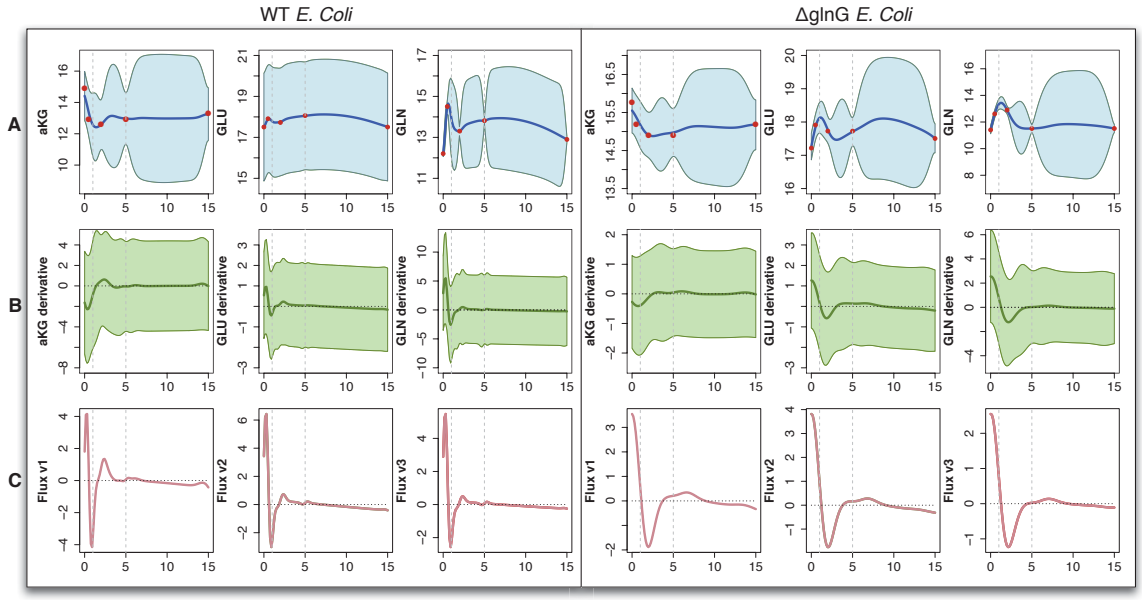


Figure 4.6: Predictions with MGPs model for *E. coli* (WT and ΔglnG). **(A)** The red colour indicates experimentally measured concentrations of αKG, GLU and GLN metabolites for WT and ΔglnG. Solid blue lines correspond to the mean behaviour of dependent GPs model and shaded area is a confidence region. **(B)** Predicted derivative behaviour for αKG, GLU and GLN metabolites, where green solid lines correspond to the mean behaviour of dependent derivative processes, and shaded area is the confidence region. **(C)** Predicted fluxes v_1, v_2 and v_3 for WT and ΔglnG *E. coli*; for convenience, dotted line illustrates horizontal 0-axis.

type (WT) *E. coli* metabolic measurements, and right – to the isogenic glnG deletion (ΔglnG) measurements. Below we will focus on the pathway summarised in Figure 4.3**C**, which is a joint version of both GDH and GS-GOGAT parts. For modelling purposes we assume that fluxes v_3 and v_4 can be summarised by the overall flux v_3 that describes the flow from GLU to GLN as there is not enough information to discriminate between them. From the pathway we can construct a system of linear equations that describe the dependence between fluxes and metabolites,

$$\begin{aligned} \frac{\mathrm{d}\alpha KG}{\mathrm{d}t} &= v_1 - v_2, & v_1 &= \frac{\mathrm{d}\alpha KG}{\mathrm{d}t} + \frac{\mathrm{d}GLU}{\mathrm{d}t} + \frac{\mathrm{d}GLN}{\mathrm{d}t}, \\ \frac{\mathrm{d}GLU}{\mathrm{d}t} &= v_2 - v_3, \quad \Longrightarrow & v_2 &= \frac{\mathrm{d}GLU}{\mathrm{d}t} + \frac{\mathrm{d}GLN}{\mathrm{d}t}, \\ \frac{\mathrm{d}GLN}{\mathrm{d}t} &= v_3, & v_3 &= \frac{\mathrm{d}GLN}{\mathrm{d}t}. \end{aligned} \tag{4.6}$$

We fit a dependent GP model (2.15) (K=3) to WT and then to ΔglnG datasets. In the model, αKG can be expressed as a sum of three GPs: the first GP describes αKG, the second expresses the relationship between αKG and GLU, and the third one describes additive noise; GLN is modelled similarly. GLU, however, is modelled as the sum of four GPs, where the first three describe, GLU; the dependence between GLU and αKG; the dependence between GLU and GLN; and the fourth is an additive noise. Choosing kernel functions to be Gaussian $h_k(t) = v_k \exp\{-\frac{1}{2}t^2 A_k\}$ we obtain the MAP estimate for all hyper-parameters (17 in total). The predictions with posterior process (4.2) are summarised in Figure 4.6, where solid blue lines describe predictions with dependent GP model for WT *E. coli*, and green lines for - ΔglnG. Using the relationship (4.6) we can estimate fluxes v_1, v_2 and v_3 (Figure 4.6**C**). In this example the multiple-output GPs model leads to large confidence intervals, this is potentially due to the fact that metabolic observations are very sparse. The covariance function in the model expresses our beliefs about the similarity between the observations. For this reason the input points that are closer to each other tend to have more similar output values. Therefore this indicates greater similarity between observations and in turn potentially more informative predictions using GPs. However, this suggests as well that the more further away from each other points are, the less similarity they bear, and consequently less instructive prediction might become. For this reason it may cause confidence intervals between the observations that are further away to be considerably larger.

To provide more support for our predictions we can compare flux v_3 and GS protein levels in WT and ΔglnG *E. coli* (see Figure 4.7). In *E. coli*, glnG gene produces a transcription factor (NtrC) (nitrogen regulator) that is controlling protein GS levels; and in its active form GS catalyses the synthesis of glutamine (van Heeswijk et al., 2013). Experimentally it was observed that in ΔglnG case protein GS levels were significantly lower compared to the GS levels in WT *E. coli* (see Figure 4.7**C**-**D**). Because there is less enzyme available to catalyse the reaction in ΔglnG, the flux v_3 should have a notably lower amplitude compared to WT flux v_3 (see Figure 4.7**A**-**B**).

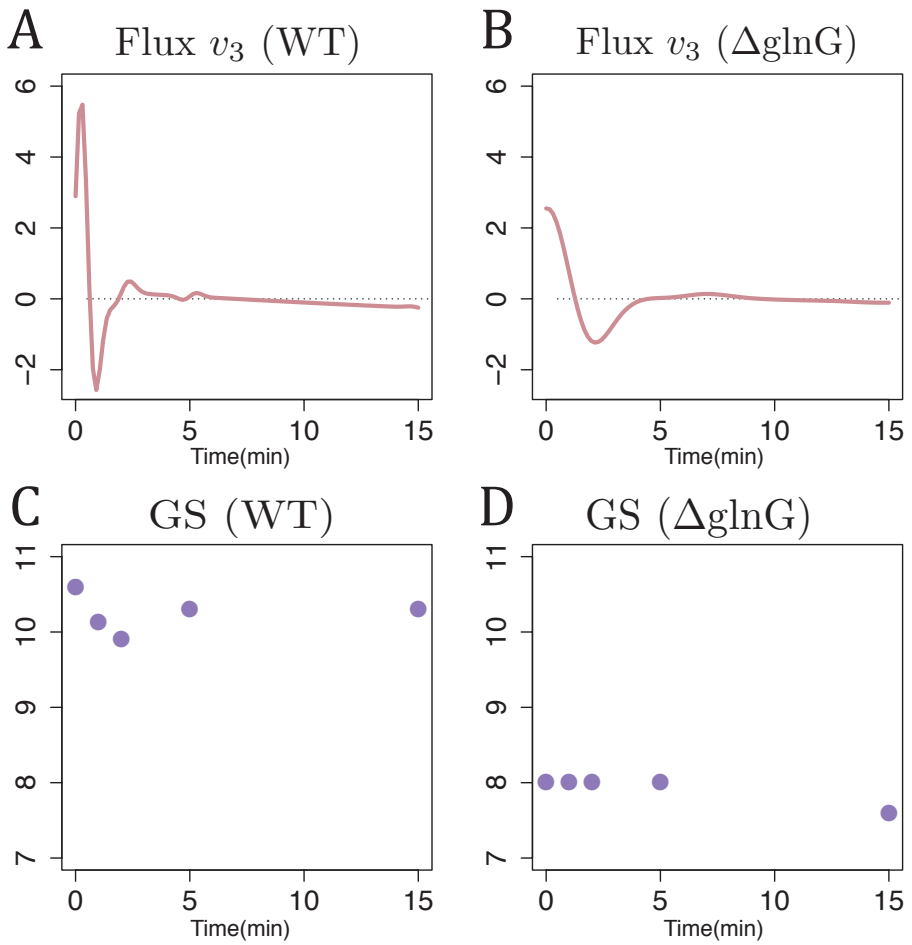


Figure 4.7: This figure is adapted from Žurauskienė et al. (2014a). Flux and glutamine synthetase (GS) comparison. **(A)** Predicted flux v_3 in wild type *E. coli*; dotted line illustrates the horizontal 0-axis. **(B)** Predicted flux v_3 in ΔglnG *E. coli*; dotted line illustrates the horizontal 0-axis. **(C)** Measurements of active form (un-adenylylated form) GS protein in wilde type *E. coli*. **(D)** Measurements of active form (un-adenylylated form) GS protein in ΔglnG *E. coli*.

4.4 Discussion and conclusions

In order to investigate how metabolism functions in cells it is a common practice to consider flux estimation problems. Typically the estimates for a set of fluxes are obtained in a point-wise fashion at discrete time-points. It is clear that this fails to capture the temporal behaviour of the fluxes and additional consideration of parametric models is compulsory in order to fully explain the fluxes; further this approach is sus-

ceptible to noise that is present in experimentally measured metabolite data.

Here, we have addressed these problems that are generally encountered in metabolic modelling/analysis, and proposed a novel nonparametric Bayesian approach to modelling metabolic fluxes. This is based on multiple-output Gaussian processes that enable the construction of derivative processes. Because the derivative processes and original processes share the same input source we can complement the dependent GP model and make joint predictions about original and derivative processes at any finite number of input points. Such derivative processes can be applied to characterise the temporal behaviour of metabolic fluxes from a time course data and here we have demonstrated the applicability on simple models and a real-world example.

GPs, including our approach, propagate uncertainty in line with the assumed covariance structures. This can lead to large confidence intervals, especially if the dependencies among different observations are not considered explicitly. Equally, with increasing number of metabolic species within the pathway, the derivative process approach might become computationally costly due to the inference of a large number of hyper-parameters and a matrix inversion step; however, this limitation potentially might be addressed by considering a sparse approximation for the full covariance matrix of all metabolic species (Alvarez and Lawrence, 2009; Alvarez et al., 2010).

To characterise the functional forms of the metabolites and fluxes is a challenging task. As it was demonstrated in this chapter, it requires the development of a new modelling tools; especially that can deal with a small number of measurements or a sparsely sampled regions within the data. How to accurately reconstruct the underlying function (e.g. protein, flux, metabolite, signal trajectories...) is not only an important question in metabolic analyses, it is more a general problem across different scientific fields. For this reason in the following chapter we discuss under what circumstances the multiple-output Gaussian processes are able to carefully reconstruct the underlying function and when they fail in this task.

Chapter 5

Bayesian Nonparametric Approaches to Oscillatory Systems and the Nyquist Limit

This chapter is based on previously published work – *Bayesian non-parametric approaches to reconstructing oscillatory systems and the Nyquist limit*. Žurauskienė J., Kirk P., Thorne T., Stumpf M.P.H. Physica A: Statistical Mechanics and its Applications, Vol. 407, p.33 - 42, 2014.

5.1 Introduction

The reconstruction of dynamical processes in nature and technology from experimental observations has been central to many scientific fields. Regression approaches, for example, take experimental data and model the empirically found relationships between free, x, and dependent variables, y, in order to capture or predict the be-

haviour of the system (Hastie et al., 2003). Here the choice of the model,

$$y \sim f(x;\theta)$$

is generally chosen in light of prior knowledge or beliefs about the correct relationship between y and x; the choice of the functional form for $f(\ldots)$ is thus of crucial importance and a wealth of statistical approaches (Silvey, 1975; Gelman et al., 2003) have been developed to choose the best models as well as sets of predictor variables, x, that allow us to explain the change in y.

But even if we have settled on the correct form for $f(\ldots)$ the dynamics captured by the regression framework may still differ substantially from the true relationship. This is perhaps less apparent in conventional linear regression frameworks, but becomes readily apparent in more complicated contexts such as dynamical systems (Birke et al., 2010; Silk et al., 2011). Here we are particularly interested in oscillatory systems; these have been characterized comprehensively in physics and many engineering contexts, and they continue to intrigue in biological contexts that range from ecosystems down to molecular networks that control, for example, circadian clocks (Aitken and Akman, 2013) and the cell cycle (Cho et al., 1998). Our aim here is to explore how we can capture such oscillatory behaviour from observing sets of random variables $X_t, Y_t, \ldots$ that depend on time t and are produced by a vector-valued source model.

In order to reconstruct the essential aspects of an oscillatory process, in particular its frequency, temporal sampling of the output has to be sufficiently dense. The theoretical framework is due to Nyquist and Shannon, and for scalar signals very straightforward; in particular the so-called Nyquist rate (Landau, 1967; Haykin and Moher, 2010), reviewed in detail below, sets the minimum frequency at which a signal needs to be sampled so that the frequency of the original signal can be reliably inferred. Here we investigate the extent to which temporal sampling affects an important class of Bayesian nonparametric approaches. Gaussian processes have seen widespread ap-

plication in signal processing (Ridley and Jakeman, 1999), machine learning (Opper and Winther, 2000; Friedman and Nachman, 2000; Seeger, 2004), and modelling of dynamical systems (Kirk and Stumpf, 2009; Roberts et al., 2013). GPs define priors over the space of differentiable functions. They are outlined and reviewed extensively in Rasmussen and Williams (2006). Typically, they proceed by considering the output of a scalar-valued function over time, and can be used to define posterior distributions that capture the temporal change in system output (including an assessment of the uncertainty). GPs for scalar inputs are, of course, subject to the same limitations imposed by the Nyquist-Shannon theorem, and inappropriate (i.e. too sparse) sampling, will result in incorrect inferences about the system dynamics: while aspects of the qualitative dynamics — oscillatory vs. non-oscillatory — may be recovered, the frequency cannot be inferred adequately below the Nyquist rate.

In many important instances, the source of the information does not, however, only produce one output, but generates vector-valued outputs. Traditionally, in the GP framework these have been treated as independent and separate GPs have been fitted to each output separately. Multi-output GPs (Boyle and Frean, 2004; Alvarez and Lawrence, 2009) allow us to detect correlated behaviour between different outputs of a system; this in turn opens up the ability of “borrowing" information between outputs if these are correlated or mutually informative in some discernible way. MGPs have to infer such dependencies from the available data, and here we investigate whether this is indeed a worthwhile pursuit. More specifically, we investigate in illustrative examples and applications to the *p53* protein signalling system, whether MGPs are superior to conventional GP procedures in correctly inferring properties of oscillatory behaviour. Below, we briefly discuss the concept of Nyquist ratio; the explicit review on GPs and MGPs was introduced before in chapter 2 (Theoretical Background). Below we are focusing on GPs and MGPs behaviour and their use in reconstructing oscillatory dynamics.

5.2 Methodology

5.2.1 Nyquist ratio for oscillatory systems

In information theory, and especially in the theory of signal processing, the signal sampling rate often determines reliability of signal transmission and recovery. Usually it is analysed in the time domain where a sequence of samples is often spaced uniformly, but in order to describe adequately or recover the signal using a finite (typically small) set of samples, it is critical to choose the correct sampling rate. The Nyquist–Shannon sampling theorem (Landau, 1967; Roberts, 2008) sets out a lower bound on this rate, below of which recovery is impossible.

Let $y(t)$ be a continuous-time periodic signal and f_m be supremum of all frequencies that constitute the signal. The theorem states that the original continuous time signal can be accurately reconstructed from the series of discrete samples only if the signal is sampled at a frequency $f_s > 2f_m$, where $2f_m$ is called the *Nyquist rate*. Thus the Nyquist rate is a minimum rate at which it is necessary to sample a signal, so that its frequency information can be recovered. A signal sampled at less than its Nyquist rate will be referred to as an *undersampled* signal; a signal sampled at greater than its Nyquist rate is accordingly referred to as an *oversampled* signal. In order to identify the Nyquist rate, it is of course better to move into the frequency domain. The Fourier transform (FT) of a signal $y(t)$ is,

$$G(F) \equiv \mathscr{F}\{y(t)\} \quad = \quad \int_{-\infty}^{\infty} y(t) \exp(-i2\pi f t) \mathrm{d}f,$$

where f is a frequency. The FT tells us which frequencies constitute the signal and the Nyquist sampling rate is readily identifiable from $G(F)$ for a given signal, $y(t)$.

In many real world situations it is necessary to deal with signals that are not contin-

uous, but are observed as discrete samples at a regular intervals. For this reason a fast Fourier transform algorithm (that performs a discrete Fourier transform) is usually applied in order to visualise the data in a frequency domain and enables one to analyse the properties of the signal. The fast Fourier transform typically produces the amount of bins, which is equal to number of the samples analysed (e.g. N). Each bin corresponds to a certain frequency that can be found using the following relationship, $\frac{nf_s}{N}$, where n is a bin number and f_s is a sampling rate in Hz. The first bin represents a DC component, which is the mean value of the waveform; and each subsequent bin represents a distinct frequency.

5.2.2 The role of multiple-output Gaussian process regression

Gaussian processes (see section 2.6) can be employed in order to obtain statistical descriptions (including an assessment of their uncertainty) of functions that describe sets of points, i.e. we can use them as a curve fitting technique. If we have sets of dependent variables for each x, i.e. we want to make predictions about several variables simultaneously, it is wise to consider a model that could capture the correlations between these variables. For example, previously co-kriging (Cressie, 1993) was used in geostatistical literature to modelling relationships between several variables and make joint predictions. An alternative way for modelling many outputs was introduced by Boyle and Frean (2004), who constructed dependant Gaussian process via multiple-input multiple-output linear filters, which where discussed in section 2.7. In this study multiple–output Gaussian processes will be applied in order to capture correlations in oscillating signals and to describe the relationship between phase and frequency.

5.3 Phase and frequency dependence

Here we investigate the performance of single and multiple–output Gaussian processes by testing them on simple simulated oscillating systems with different phase and frequency values. Additionally, we explore the impact of sparse sampling of the data on the GP performance quality.

5.3.1 Variations in phase

Here consider a simple form of sinusoidal signal, $f(t) = A\sin(\omega t + \phi)$, where A is an amplitude, $\omega = 2\pi f$ an angular frequency, and ϕ the phase. Our first example consists of two sinusoidal signals, where f_2 is simply a shifted version of signal f_1. This means that both signals have identical amplitudes and frequencies but are phase shifted, $f_1(t) = sin(2t)$ and $f_2(t) = sin(2t + \frac{\pi}{4})$, on an interval $t \in [0, 4\pi]$. To mimic real experimental measurements we added random noise to the simulated trajectories, $Y_1(t) = f_1(t) + \epsilon_1$, $Y_1(t) = f_2(t) + \epsilon_2$, where $\epsilon_i \sim \mathcal{N}(0, 0.1^2)$. Figure 5.1**A**, **B** illustrates the simulated sinusoidal trajectories with different shifts in time, here $\phi_1 = 0, \phi_2 = \frac{\pi}{4}$ and $\phi_1 = 0, \phi_2 = \pi$ respectively. From these noisy measurements we recorded a dataset of $N = 20$ data points. Selected points, representing 10 measurements per output signal, were spaced at regular intervals. In order to build a single model that would capture a relationship between both signals, we applied two dependent GPs framework (2.15) (K=2), where each signal can be expressed as a superposition of three Gaussian processes — two of which are being constructed via convolution between a noise source and a Gaussian kernel and the third one is an additive noise. We set parameters A_i of each Gaussian kernel to be $\exp(f_i)$ and noise levels to $\sigma_1 = \exp(\eta_1)$, $\sigma_2 = \exp(\eta_2)$, leading to a set of hyper-parameters $\theta = (v_i, f_i, \mu_1, \mu_2, \eta_1, \eta_2)$, $i = 1, \ldots, 4$. To build the model the following priors were chosen: $v_i, f_i \sim \mathcal{N}(1, 2^2)$, $\eta_j \sim \mathcal{N}(-2, 2^2)$ and $\mu_j \sim \mathcal{N}(0.5, 1^2)$, $j = 1, 2$; and the

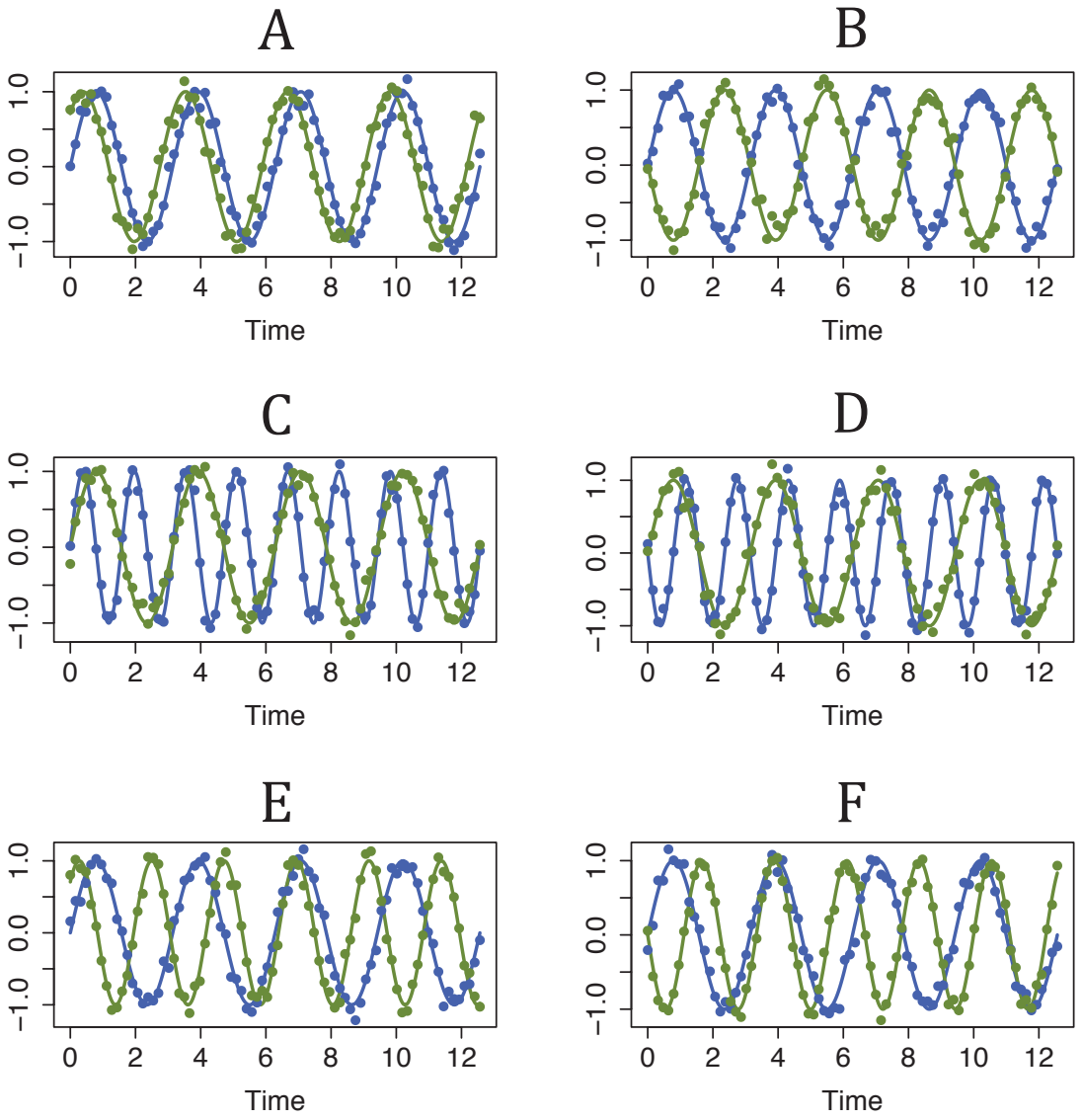


Figure 5.1: This figure is adapted from Žurauskienė et al. (2014b). Simulated datasets. Solid blue lines correspond to noiseless trajectories $f_1(t) = A\sin(\omega_1 t + \phi_1)$, and solid green lines to - $f_2(t) = A\sin(\omega_2 t + \phi_2)$; dots represent 80 measurements with added Gaussian noise $\mathcal{N}(0, 0.1^2)$. **(A)**. Dataset with parameters $A = 1$, $\omega_1 = \omega_2 = 2$, $\phi_1 = 0$, $\phi_2 = \frac{\pi}{4}$. **(B)**. Dataset with parameters $A = 1$, $\omega_1 = \omega_2 = 2$, $\phi_1 = 0$, $\phi_2 = \pi$. **(C)**. Dataset with parameters $A = 1$, $\omega_1 = 4$, $\omega_2 = 2$, $\phi_1 = \phi_2 = 0$. **(D)**. Dataset with parameters $A = 1$, $\omega_1 = 4$, $\omega_2 = 2$, $\phi_1 = \pi$, $\phi_2 = 0$. **(E)**. Dataset with parameters $A = 1$, $\omega_1 = 2$, $\omega_2 = 2\sqrt{2}$, $\phi_1 = 0$, $\phi_2 = \frac{\pi}{4}$. **(F)**. Dataset with parameters $A = 1$, $\omega_1 = 2$, $\omega_2 = 2\sqrt{2}$, $\phi_1 = 0$, $\phi_2 = \pi$.

maximum a posteriori (MAP) estimate $\hat{\theta}$ was calculated using a multi-starting Nelder-Mead optimisation algorithm (Nelder and Mead, 1965). Dependent GPs posteriors

(2.18) allow us to make predictions about both signals at any finite number of input points, and the resulting posterior processes are given in Figures 5.2**A** and **D**. To fit MGP we used 200 equally spaced input points in the considered interval $[0, 4\pi]$. Next, samples evaluated from MGP models at these input points were used in order to obtain $\omega_{1,2}$ estimates $\hat{\omega}_{1,2}$. In both examples we obtained $\hat{\omega}_{1,2} = 2$, which agrees with $\omega_{1,2} = 2$ that were used to simulate initial datasets. For comparison, we fitted two independent GP models to signals that are given in the dataset used in Figure 5.2**A**. From the posterior processes (see Figure (5.3)) it can be seen that in order to correctly capture the oscillations either it is necessary to consider the relationship between the signals or increase the number of observations per signal above the Nyquist sampling rate. However, it is worth noting that fitting independent GP models allowed us to accurately recover both angular frequencies, and we were able to obtain $\hat{\omega}_{1,2} = 2$ estimates that agree with original $\omega_{1,2} = 2$ values. In general, the performance of dependent GPs and independent GPs are in good agreement for signals that are sampled at sufficiently high frequencies; at low frequencies, however, the dependency structure implemented here allows us to reconstruct signal frequencies — and signal shapes more generally — at a rate below the Nyquist sampling rate.

Based on the previous example (see Figure 5.1**A**), we next assume that we have a dataset with $N = 15$ observations: $N_1 = 10$ observations of signal f_1 and $N_2 = 5$ of signal f_2. Repeating the above modelling procedure we obtained dependent posterior processes for both signals. It can be seen in Figure 5.2**B** that the dependent GP model can provide an excellent estimation of both signals in circumstances where one signal is undersampled. Here, we were able to accurately reconstruct angular frequency values for both signals, which are $\hat{\omega}_{1,2} = 2$. Both signals can be accurately estimated because of the strong relationship between the signals which can be captured by the (constant in time) covariance matrix. By contrast, an independent GP model fitted to signal f_2 exhibits the so-called “aliasing" phenomenon and which applied to realistic experimental signal would lead to a serious misinterpretation of the underlying pro-

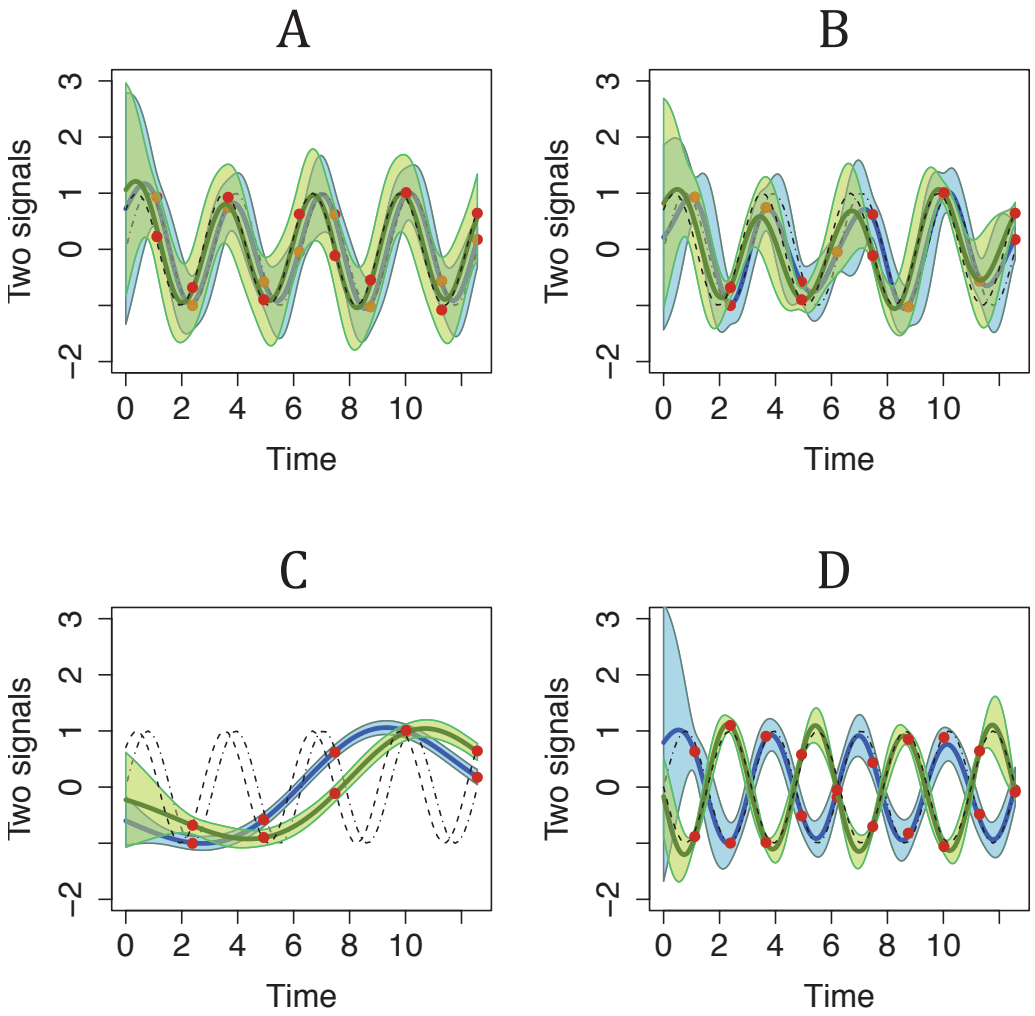


Figure 5.2: This figure is adapted from Žurauskienė et al. (2014b). MGPs. Solid lines represent the mean of the model; blue and green areas correspond to two standard deviations at each prediction point for outputs one and two respectively; red points are noisy observations and dashed lines correspond to the true sinusoidal signals. **(A)** MGP model for $N = 20$ dataset with parameters $A = 1$, $\omega_1 = \omega_2 = 2$, $\phi_1 = 0$, $\phi_2 = \frac{\pi}{4}$. **(B)** MGP model for $N = 15$ dataset with parameters $A = 1$, $\omega_1 = \omega_2 = 2$, $\phi_1 = 0$, $\phi_2 = \frac{\pi}{4}$. **(C)** MGP model for $N = 10$ dataset with parameters $A = 1$, $\omega_1 = \omega_2 = 2$, $\phi_1 = 0$, $\phi_2 = \frac{\pi}{4}$. **(D)** MGP model for $N = 20$ dataset with parameters $A = 1$, $\omega_1 = \omega_2 = 2$, $\phi_1 = 0$, $\phi_2 = \pi$.

cess. However, if we further reduce the number of observations and consider a dataset of $N = 10$ (5 per signal) observations, even the MGP model behaves poorly and is unable to correctly capture the original trajectories and results in aliasing of both signals (see Figure 5.2**C**, in this example samples from original signals were generated using

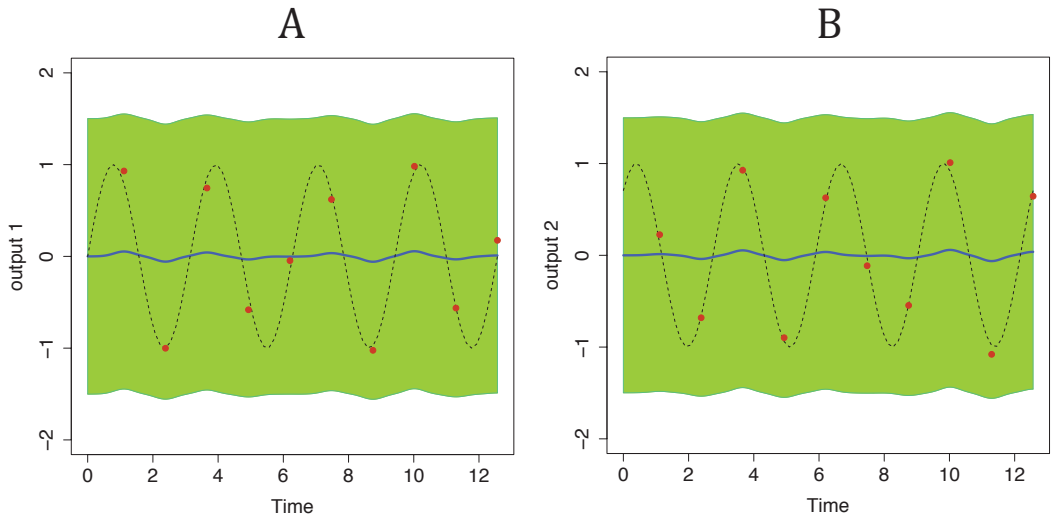


Figure 5.3: This figure is adapted from Žurauskienė et al. (2014b). Independent GP models for $N = 20$ dataset with parameters $A = 1$, $\omega_1 = \omega_2 = 2$, $\phi_1 = 0$, $\phi_2 = \frac{\pi}{4}$. Solid lines represent the mean of the model; green areas correspond to two standard deviations at each prediction point.

$\omega_{1,2} = 2$; however, from MGP model we obtain estimates $\hat{\omega}_{1,2} = 0.5$. This shows that we have recovered signals that have different properties). Note, however, that we inferred all aspects of the MGP from the provided data, in particular the covariances.

5.3.2 Variations in frequency and phase

Next we consider a system with two oscillating signals, fast and slow and with different phase shifts (see Figure 5.1**C** and **D**). The data consist of $N_1 = 25$ noisy observations of the fast signal - $f_1(t) = sin(4t)$ and $N_2 = 15$ noisy observations of the slow signal - $f_2(t) = sin(2t)$, $t = [0, 4\pi]$ resulting in a joint dataset of size $N = 40$ observations. Again, we applied the MGP framework (2.15) with Gaussian kernels and $\theta = (v_i, f_i, \mu_1, \mu_2, \eta_1, \eta_2)$, $i = 1, \ldots, 4$. We kept priors for hyper-parameters of the model as described in previous section, and used multi-starting Nelder-Mead algorithm to obtain the estimated values $\hat{\theta}$. The resulting GPs are summarised in Figure (5.4) where solid blue and green lines represents the mean behaviour of the posterior process and shaded areas corresponds to two standard deviations at each prediction

point for f_1 and f_2, respectively. **A** and **B** illustrate the MGPs where $\phi_1 = \phi_2 = 0$ and $\phi_1 = \pi, \phi_2 = 0$; in both examples we were able to accurately estimate $\omega_{1,2}$ values, which are $\hat{\omega}_1 = 4$ and $\hat{\omega}_2 = 2$. In this case reducing the number of observations so that the fast signal would be undersampled results in aliasing; and the fast signal is influenced to adapt the frequency of the slow signal. By contrast, in situations where the slow signal is undersampled and the fast signal is oversampled the aliasing occurs in the undersampled signal in such a way that the inferred slow signal is forced to adopt the behaviour of the fast signal. Fitting MGP models for oscillating signals with relationship between frequencies $\omega_2 = a\omega_1$, where a is a constant, and different values of phase parameter (for example $\frac{\pi}{2}, \frac{3\pi}{8}, \frac{\pi}{4}, \frac{\pi}{8}$), leads to reasonably good predictions about the true signals. This is especially, and trivially, true if both signals are oversampled. This can be explained by the fact that covariance structure between the signals is no longer constant but varies over time, resulting in weaker dependence between the signals, which in turn complicates the inference.

In Figure 5.4**A** and **B** the Nyquist rate for fast signal is $\frac{4}{\pi}$ and $\frac{2}{\pi}$ for a slow signal, meaning that we require to sample the signals at rates which are greater than 16 and 8 samples per signal in 4π interval of time. Similarly, Figure 5.4**C** and **D** illustrates the MGP fit to the data where signals are related via $\omega_2 = \sqrt{2}\omega_1$ frequencies, here angular frequency estimates are, $\hat{\omega}_1 = 2$ and $\hat{\omega}_2 = 3$; and the Nyquist rate for the fast signal is $\frac{2\sqrt{2}}{\pi}$ and for the slow, $\frac{2}{\pi}$; For an accurate reconstruction of both signals it is therefore necessary to have more than 12 samples of the fast and more than 8 samples of the slow signal in a considered interval of time.

5.4 Applications to systems biology data

Many of the problems in the analysis of biological systems involve processes that show regularly repeating patterns in both time and space. Cell cycle, diurnal cycles and

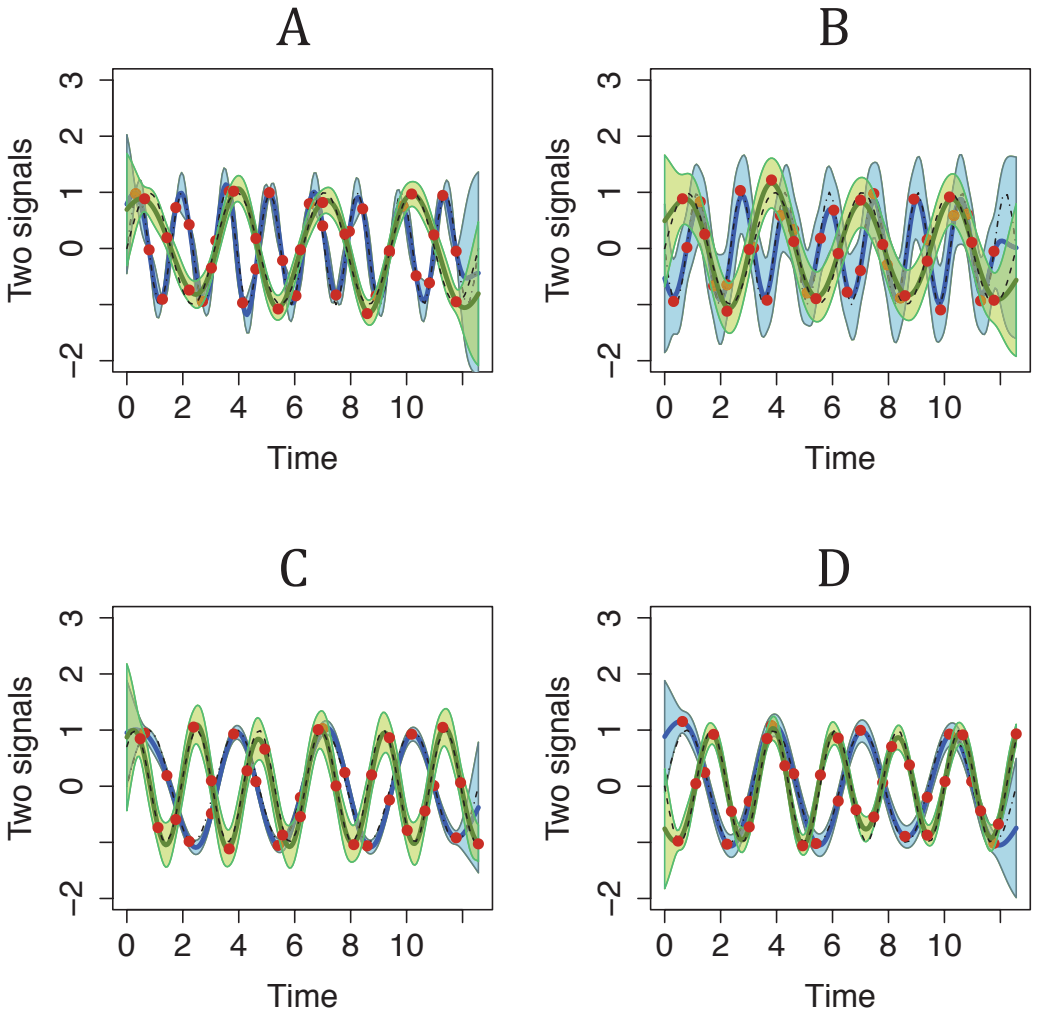


Figure 5.4: This figure is adapted from Žurauskienė et al. (2014b). MGPs. Solid lines represent the mean of the model; blue and green areas correspond to two standard deviations at each prediction point for outputs one and two respectively; red points are noisy observations and dashed lines correspond to the underlying sinusoidal signals. **(A)** MGP model for $N = 40$ dataset with parameters $A = 1$, $\omega_1 = 4$, $\omega_2 = 2$, $\phi_1 = \phi_2 = 0$. **(B)** MGP model for $N = 40$ dataset with parameters $A = 1$, $\omega_1 = 4$, $\omega_2 = 2$, $\phi_1 = \pi$, $\phi_2 = 0$. **(C)** MGP model for $N = 35$ dataset with parameters $A = 1$, $\omega_1 = 2$, $\omega_2 = 2\sqrt{2}$, $\phi_1 = 0$, $\phi_2 = \frac{\pi}{4}$. **(D)** MGP model for $N = 35$ dataset with parameters $A = 1$, $\omega_1 = 2$, $\omega_2 = 2\sqrt{2}$, $\phi_1 = 0$, $\phi_2 = \pi$.

clocks are the canonical examples for such regularly recurring temporal patterns. A host of other systems have been reported to oscillate over physiological time-scales and here we illustrate how MGPs perform on one such system, the *p53-Mdm2* signalling system, under different data sampling schedules.

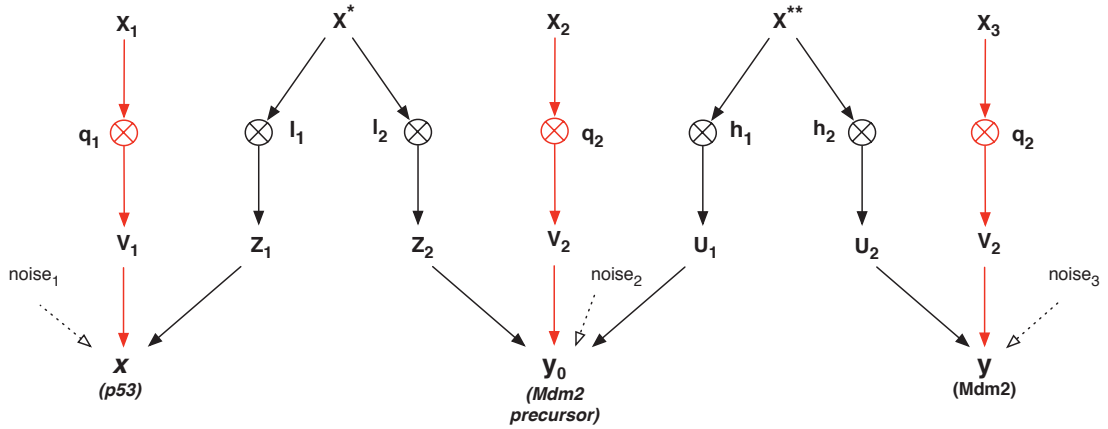


Figure 5.5: Graphical illustration of multiple-output Gaussian process model for *Mdm2-p53* system.

5.4.1 *p53-Mdm2* system

Proteins are biomolecules that are responsible for many cellular activities such as providing structural molecules, catalysing biochemical reactions or participating in cell signalling and signal transduction. The protein *p53* stands out due to its ability to participate in regulation of cell cycle, response to DNA damage and tumour suppression. Under stress conditions, *p53* concentration levels increase within the cell and physical interactions with *Mdm2* stabilise *p53* levels. This is done by inhibiting *p53* transcriptional activity and increasing its degradation rate; this can then lead to oscillation in both protein species.

A widely used model for the *p53-Mdm2* system was first proposed by Geva-Zatorsky et al. (2009); here the influence of *Mdm2* on *p53* occurs in a nonlinear fashion via Michaelis–Menten dynamics,

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \beta_x - \alpha_x x - \alpha_k y \frac{x}{x+k}, \\ \frac{\mathrm{d}y_0}{\mathrm{d}t} &= \beta_y x - \alpha_0 y_0, \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= \alpha_0 y_0 - \alpha_y y. \end{aligned} \tag{5.1}$$

Here x, y_0 and y corresponds to the nuclear levels of *p53*, *Mdm2* precursor and *Mdm2*, respectively (see (Geva-Zatorsky et al., 2009) for detailed explanation of the model and parameters). Dashed lines given in Figure 5.6 illustrate the simulated trajectories from the ordinary differential equation model with a set of parameters $(\beta_x, \alpha_x, \alpha_k, \beta_y, \alpha_0, \alpha_y, k) = (0.9,\ 0,\ 1.7,\ 1.1,\ 0.8,\ 0.8,\ 0.0001)$ and initial conditions $(x, y_0, y) = (0.0, 0.1, 0.8)$.

To evaluate how we can benefit from the MGP framework for reconstruction of the concentration levels of sparsely sampled protein species we investigate four differently sampled datasets. The first dataset in Figure 5.6**A** is a control case where all protein species are well oversampled and dependent GPs model (2.15) (K=3) accurately describe noisy observations of all proteins. To build our model we described each protein by a linear sum of four Gaussian processes, where dependence between all proteins is introduced via two shared input noise sources under convolutions with different Gaussian kernels (Figure 5.5). We applied such model to datasets given in Figures 5.6**B** and **C**, where two proteins are oversampled ($N_1 = N_2 = 20$) and one is undersampled ($N_3 = 6$). It can be seen that predictions with posterior GPs allow us to fairly well reconstruct the concentration levels of all proteins. However, in cases where dataset consists of any two proteins that are undersampled ($N_1 = N_2 = 6$) and only one is oversampled ($N_3 = 20$) the dependent GPs posterior can capture correct oscillation only for oversampled protein and leads to “aliasing" phenomenon of undersmpled proteins (see Figure 5.6**D**).

5.5 Discussion

Constructing dependent Gaussian processes via convolution involving sets of Gaussian white noise processes and appropriate kernel functions offers considerable advantages compared to traditional methods. In particular we are able to account for covariances

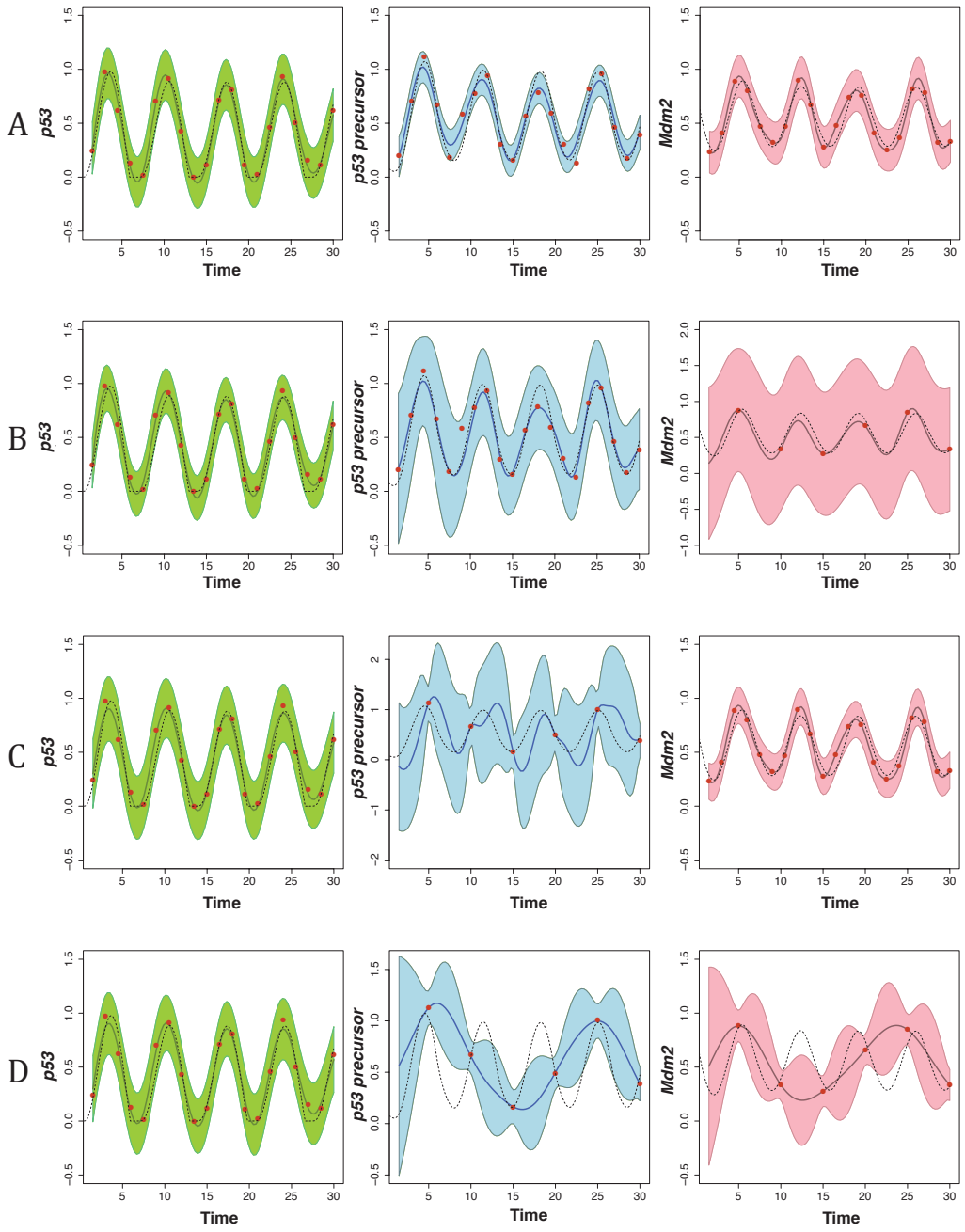


Figure 5.6: This figure is adapted from Žurauskienė et al. (2014b). MGPs for the *p53-Mdm2* system. Solid lines represent the mean of the model; blue, green and pink areas correspond to two standard deviations at each prediction point for all outputs; red points are noisy observations and dashed lines correspond to the underlying *p53-Mdm2* ODE model behaviour. **(A)** MGPs for all species being oversampled. **(B)** MGPs for oversampled *p53* and *Mdm2* precursor, and undersampled *Mdm2*. **(C)** MGPs for oversampled *p53* and *Mdm2*, and undersampled *Mdm2* precursor. **D** MGPs for oversampled *p53* and undersampled *Mdm2* precursor and *Mdm2*.

between outputs and use this information to improve the predictive power substantially. Here we have used computationally affordable linear filters to construct a single MGP rather than several single-output GPs.

Experimental measurements are frequently hard to come by and we have explored the use of MGPs in the analysis of oscillatory systems as a potential means of accounting for potential under-sampling of such oscillatory systems. For oscillating systems with scalar output the Nyquist sampling rate sets the limit below which recovery of the correct oscillatory pattern — here we are particularly concerned with the frequency of oscillations — is no longer possible. For vector-valued output such hard and fast rules are harder to come by and we resorted to simulations to explore the use of MGPs in reconstruction of oscillatory (vector-valued) outputs. This problem reduces in a sense to inferring an appropriate covariance matrix between the signals/system outputs.

This turns out to be straightforward for systems where the different states oscillate at the same frequency (but with a phase-shift). Here undersampling one output (below the Nyquist rate appropriate for a single output) can be compensated for by sampling the other signal sufficiently densely (above the Nyquist rate), whence the MGP provides an adequate description of the whole output (as opposed to conventional single output GPs). But MGP performance on small datasets strongly depends on the nature of both signals and undersampling all outputs leads to aliasing as the covariance matrix cannot be inferred sufficiently well. For cases where different outputs oscillate at different frequencies, the covariance is no longer constant in time and all outputs need to be sampled at high enough rate for MGPs to result in reliable predictive distributions over the system outputs.

MGPs thus offer advantages in cases where correlations between different outputs exist and, crucially, can be learned from sparse input, compared to conventional GPs, which treat each output independently.

5.6 Conclusions

Multipe-output Gaussian process models provide a way to link data driven and mechanistic modelling. In this chapter we explored the performance and applicability of MGP regression approach on carefully chosen examples with the aim to emphasize various modelling outcomes that might depend on the data size. However, we believe that in the model the covariance structure plays an important role, especially in conveying our beliefes about the underlying similarities between all variables and between data points of each variable (Rasmussen and Williams, 2006). For this reason the applicability of MGPs to model biological systems can be made more attractive by further exploring the effects of alternative kernel functions, which potentially could provide better fits to the data in certain situations. Although this approach enables us to capture the existing dependencies between several variables, it is not applicable for generating hypotheses and predicting the underlying mechanistic structure (or predicting dependencies between several variables). For this reason, the next chapter will be focusing on development of new tools that will enable us to generate hypotheses about underlying biological regulatory processes. In addition we will discuss it in a Bayesian nonparametric framework that employs Dirichlet processes.

Chapter 6

A Graph Theoretical Approach to Data Fusion

6.1 Introduction

Due to the fast development of experimental technologies, high-throughput genomic measurements become cost-effective and increasingly available. This also accounts for the diversity of data types and structures. With the accessibility of such immense quantities of data new demands and challenges are coming to light. One of which is the development of novel statistical and computational tools that can jointly analyse the data arising from distinct genomic sources and generate unified hypotheses about the underlying biological processes. The development of such integrative modelling tools is particularly important as they should provide us with ways to better understand the regulatory molecular mechanisms that are driving various diseases (for example cancer).

Currently the necessity for data integration is rapidly evolving and is moving towards potential clinical applications (Altman, 2013; Chen et al., 2012). For example, recently

proposed methods try to group cancer patients into subtypes using fused similarity networks based on a combination of DNA methylation, *m*RNA expression and *m*iRNA expression datasets (Wang et al., 2014). Equally, it has been suggested to integrate the information from several datasets in order to subtype cancers by allowing the model to learn both – the overall and data specific clusterings (Lock and Dunson, 2013). Furthermore, other studies are focusing on integrative modelling in order to identify potential regulatory mechanisms. For example, Zhang et al. (2011) proposed a new technique that allows one to integrate the information from several genomic data sources and identify *m*iRNA-gene regulatory comodules involved in cancer. In addition, Kirk et al. (2012) are focusing on modelling pairwise similarities between datasets, which allows the identification of protein complexes whose genes undergo transcriptional co-regulation in yeast. Overall, the existing techniques can be broadly classified either as being Bayesian methods (Kirk et al., 2012; Lock and Dunson, 2013; Savage et al., 2013), which are commonly considered to be more accurate and reliable but might be computationally challenging to apply on a full genomic datasets; or non-Bayesian (Wang et al., 2014; Shen et al., 2009; Zhang et al., 2011) approaches, which can operate on the genome-scale data but the clustering outcomes might be noisy and difficult to interpret.

In this work we introduce a new methodology for dealing with integrative modelling of genomic datasets. Our technique can be employed in order to fuse information from several data sources and extract cluster structures that are shared across all datasets. Further, we present a collection of data integration approaches that can be compatible with Bayesian and non-Bayesian clustering methods. Our main integrative technique consist of two basic steps: clustering the datasets (or data types) with the most relevant Bayesian non-parametric technique(s) (see section 2.2) and based on the outcomes, integrating the results with one of the approaches developed here (see section 2.3). For the purpose of cluster identification we employ a Dirichlet process mixture (DPM) models with either a Gaussian process (GP) or a multinomial likelihood function. We

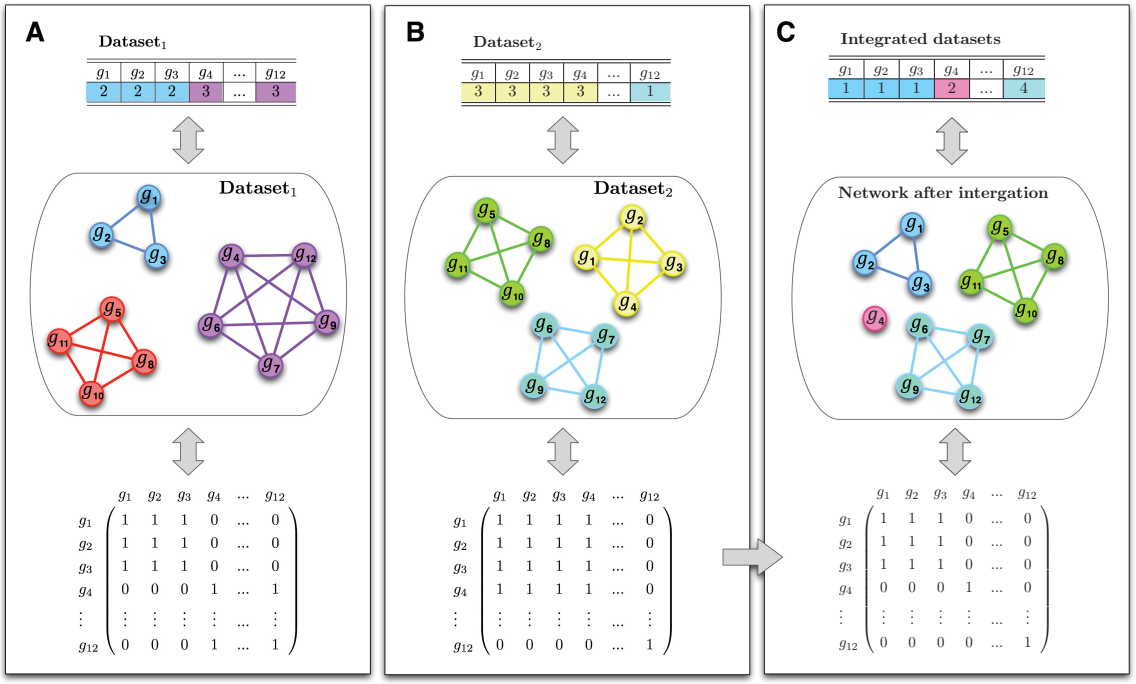


Figure 6.1: Method illustration. It is common to visualise the clustering outcome in a “table-like" fashion, by listing all genes next to their associated cluster labels. To visualise this, we construct corresponding graphs; here, each node in the network represents a gene and a line indicates that two genes cluster together. We use different colour schemes to represent cluster labels. By adopting a graph-theoretical approach we can represent each network as an adjacency matrix, which in turn can be used for data integration. **(A)** Illustrates an artificial example of 12 genes that are assigned into three clusters (e.g. control case) from the first dataset. **(B)** Illustrates the same list of genes assigned into different three clusters (e.g. disease case) from a second dataset. **(C)** Illustrates the corresponding network (and cluster assignment) after performing the data integration step.

favour the DPM models due to their natural ability to determine the number of clusters within the datasets. In our approach the actual data integration is performed by constructing the connectivity networks that represent each data source, and then by preserving those edges that are shared across all datasets (Figure 6.1). The final network can be viewed as a shared network, which is the same across all data types and provides the information about which “genes" are clustering similarly in all datasets. This integration step is somewhat similar to the consensus clustering approach (Monti et al., 2003), which was originally developed for the purpose of assessing the cluster stability, and for providing the consensus across multiple evaluations of the same clus-

tering approach. Compared to other existing techniques (e.g. *Multiple data integration* (MDI) by Kirk et al. (2012), or *Bayesian consensus clustering* (BCC) by Lock and Dunson (2013)) our approach clusters all datasets independently. Despite the fact that such independent pre-processing of each dataset cannot fully and explicitly take into account the potential similarities between data sources, it enables us to perform computations in parallel. This particular feature might be favourable in cases where it is necessary to rerun analyses in order to consider additional datasets. Equally, because our graph theoretical approach is compatible with Bayesian and non-Bayesian clustering methodology, the analysis can be easily extended to the full genome-scale datasets.

6.1.1 Bayesian integration of multiple datasets

In this section we provide a summary of Bayesian integration of multiple datasets, MDI (Kirk et al., 2012). MDI is an integrative clustering tool that seeks to model diverse datasets and capture dependencies by considering a correlation structure between allocation variables. It exploits Dirichlet-multinomial allocation (DMA) mixture model approach to model each dataset (Green and Richardson, 2001). Using sufficiently large number of components (e.g. equal to the number of observations, half this number etc.), DMA model provides the approximation to a standard Dirichlet process mixture model. Below, N component DMA mixture model can be summarised as,

$$p(x) = \sum_{c=1}^{N} \pi_c f(x|\theta_c),$$

where $f(x|\theta_c)$ is density model that is parameterised with a set of parameters θ_c and π_c are mixing proportions. Given K datasets and introducing the component allocation variables $c_j \in \{1, ..., N\}$ for each of them, MDI connects all K DMA models via a

conditional prior that is placed on component allocation variables,

$$p(c_{i1}, c_{i2}, ..., c_{iK}|\phi) \propto \prod_{k=1}^{K} \pi_{c_{ik}k} \prod_{k=1}^{K-1} \prod_{l=k+1}^{K} (1 + \phi_{kl} \mathbb{1}(c_{ik} = c_{il})),$$

here $\pi_{c_{ik}k}$ denotes a mixing proportion for component c_{ik} in model k, and parameter $\phi_{kl} \in \mathbb{R}_{\geq 0}$ explains how strong the association between datasets k and l is. The above prior is a major model component that links all datasets, for this reason it enables one to consider dependencies between several datasets and perform integrative clustering.

6.1.2 Overview of Bayesian consensus clustering

In this section we provide an overview to another Bayesian modelling technique, called Bayesian consensus clustering, that can be applied to perform integrative clustering. When applied to several datasets, it allows us to simultaneously estimate data specific clustering and equally provides one overall estimate – a consensus clustering. BCC is based on a finite Dirichlet mixture model that is adjusted to operate on a number of datasets, $\mathbf{X}_1, ..., \mathbf{X}_M$; in the same manner it allows each dataset to posses a unique probability model $f_m(X_{mn}|\theta_m)$, where X_{mn} denotes the data point n that arise from dataset m and θ_m is a set of parameters associated with probabilistic model f_m. BCC main assumption is that each dataset can have a unique clustering structure, however these clusterings contribute to the overall consensus with a probability $\alpha = (\alpha_1, ..., \alpha_m)$. If $\mathbf{L} = (L_{m1}, ..., L_{mn})$, where $L_{mn} \in \{1, ..., K\}$, denotes the clustering for dataset m and $\mathbf{C} = (C_{m1}, ..., C_{mn})$, where $C_{mn} \in \{1, ..., K\}$, denotes the consensus, then the conditional model can be expressed as,

$$P(L_{mn} = k|X_{mn}, \theta_{mk}, C_n) = \nu(k, C_n, \alpha_m) f_m(X_{mn}|\theta_{mk})$$

here v is called dependence function, and has the following format,

$$v(k, C_n, \alpha_m) = \begin{cases} \alpha_m, & \text{if} \quad C_n = L_{mn} \\ \frac{1-\alpha_m}{K-1}, & \text{otherwise} \end{cases},$$

here $\alpha \in [1/K, 1]$ can be understood as a probability that $L_{mn} = C_n$. If further $\pi_k = P(C_n = k)$ denotes the probability of a data point belonging to the consensus cluster k, then the probability for data point to belong to a data specific cluster is $P(L_{mn} = k|\pi_1, ..., \pi_k) = \pi_k\alpha + (1 - \pi_k)\frac{1-\alpha_m}{K-1}$. From here follows the conditional distribution of $\mathbf{C}$,

$$P(C_n = k|\mathbf{L}, \pi_1, ..., \pi_k, \alpha) \propto \pi_k \prod_{m=1}^{M} v(L_{mn}, k, \alpha_k).$$

The estimation of this model can be achieved via general MCMC procedure.

6.2 Methodology

In this section we review Bayesian nonparametric approaches for modelling heterogeneity in genomic datasets, and propose a novel methodology for integrating clustering outcomes across several datasets. The proposed technique is not limited to Bayesian clustering approaches and a modified version of methodology is discussed later in this chapter.

6.2.1 Dirichlet process mixtures

The methodology for modelling heterogeneity in genomic datasets bears similarity to the structure of an infinite Gaussian mixture model (Rasmussen, 2000; Neal, 2000), which was introduced in chapter 2. A Dirichlet process mixture model can be derived

as a limit of a finite mixture model when the number of mixture components grows to infinity and below for convenience we recall it with all necessary alterations.

Let us consider a dataset $D = \{x_1, .., x_N\}$ that we intend to model by the following mixture model,

$$p(x_1, \ldots, x_N|\pi, \theta) \sim \sum_{k=1}^{K} \pi_k F(D|\theta_k), \quad \pi_k > 0, \quad \sum_{k=1}^{K} \pi_k = 1, \tag{6.1}$$

where K is the number of components, π_k are the mixing proportions, and $F(D|\theta_k)$ are component density functions parameterised with a set of parameters θ_k. Furthermore, we associate each data point, x_i, with a component indicator variable $c_i \in \{1, \ldots, K\}$. This allows us to track which mixture component generated a data point x_i. We can allocate a symmetric Dirichlet prior to the mixing proportions,

$$p(\pi_1, ..., \pi_K|\alpha) = \Gamma(\alpha)/\Gamma(\alpha/K)^K \prod_{k=1}^{K} \pi_k^{\alpha/K-1},$$

where $\frac{\alpha}{K}$ is a concentration parameter; and a multinomial prior,

$$p(c_1, ..., c_K|\pi) = \prod_{k=1}^{K} \pi_k^{n_k},$$

to the indicator variables with n_k indicating the number of times $c_i = k$ (the number of observations that have the same indicator value). Then a conditional prior for a single indicator (all others being given) is obtained by integrating over the mixing proportions

$$p(c_i = k|c_{-i}, \alpha) = \frac{n_{-i,k} + \alpha/K}{N - 1 + \alpha},$$

where the subscript, "$-i$", is a short notation for all indicators excluding i; and $n_{-i,k}$ denotes the number of observations within cluster k not including observation x_i. Now, by taking the limit as K goes to infinity the conditional prior has the following

limits,

$$p(c_i = k|c_{-i}, \alpha) = \frac{n_{-i,k}}{N-1+\alpha}, \tag{6.2a}$$

$$p(c_i \neq c_{i'}, i' \neq i|c_{-i}, \alpha) = \frac{\alpha}{N-1+\alpha}. \tag{6.2b}$$

Combining conditional priors (6.2) with a likelihood function, $F(x_i|\theta_k)$, will result in conditional posteriors,

$$p(c_i = k|c_{-i}, \alpha) \propto \frac{n_{-i,k}}{N-1+\alpha} F(x_i|\theta_k), \tag{6.3a}$$

$$p(c_i \neq c_{i'}, i' \neq i|c_{-i}, \alpha) \propto \frac{\alpha}{N-1+\alpha} \int F(x_i|\theta_j) H(\theta_j) \mathrm{d}\theta_j, \tag{6.3b}$$

that are necessary to perform the inference of all parameters associated with model (6.1). This can be achieved via Markov chain Monte Carlo methods (Neal, 2000). In equations (6.3) H denotes a prior for parameters θ_k, which might be a conjugate prior and depends on the likelihood model.

Above we have introduced a DPM model as a general framework, and for this reason we will be able to focus on specific likelihood models that can capture all necessary data properties. Specifically, we will employ DPM of Gaussian process regression models to cluster gene expression time series and DPM of multinomial models to model categorical/discrete functional genomics data.

Likelihood for a Gaussian process model.

For convenience, below we will use the following notation, $\mathbf{x_i} \equiv x_i$. Instead of specifying a parametric (e.g. a multivariate-Gaussian) likelihood function, in this work we capture the time course observations $\mathbf{x_i} = \{x_i(t_1), \ldots, x_i(t_p)\}$, where $x_i(t_j)$ denotes the measurement taken on gene i at time point t_j, with a regression model. In a

regression approach, each gene x_i can be expressed as

$$x_i(t_j) = f_i(t_j) + \epsilon_{ij},$$

where f_i is a regression function, and $\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$ is added to express the potential uncertainty in measurements. In our case we are modelling observations (genes) that tend to cluster together. This means that each cluster can be described by the same "data generating" function $f_i \equiv f_k$ and noise $\sigma_i^2 \equiv \sigma_k^2$ model; here, $k = 1, ..., K$. In order to identify the function $f_k = [f_k(t_1), \ldots, f_k(t_p)]$ for each cluster, we adopt a Bayesian nonparametric approach by specifying a Gaussian process prior for the function f_k. In order to specify a GP prior we need to define two main characteristics: a mean, m, and covariance, cov, functions. Such a GP prior allows us to describe the Gaussian distributions that are associated with unique gene clusters. In simple terms this means that the function f_k evaluated at a finite number of input points $t_1, \ldots, t_p$ will have a multivariate Gaussian distribution with zero mean and there exists a covariance function, cov, such that,

$$[f_k(t_1), ..., f_k(t_p)]^T \sim \mathcal{N}\left(0, \text{cov}(t_i, t_j)\right); \quad t_i, t_j \text{ – are any two inputs.}$$

Here, for simplicity, we adopt a zero mean function ($m(t) = 0$, for all t) and squared exponential function,

$$\text{cov}(t_i, t_j) = a_k^2 \exp\left(-\frac{(t_i - t_j)^2}{2l_k}\right), \tag{6.4}$$

where $a_k, l_k > 0$ are the hyper-parameters. Then, the genes (observations) within each cluster, k,

$$\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{N_k}^{(k)} | f_k, \sigma_k^2 \sim \mathcal{N}(f_k, \sigma_k^2 I_p).$$

Here, N_k is the number of observations in cluster k. For convenience we can rewrite the above in an expanded form,

$$\Big[x_1^{(k)}(t_1),\ldots,x_{N_k}^{(k)}(t_1),\ldots,x_1^{(k)}(t_p),\ldots,x_{N_k}^{(k)}(t_p)\Big]\Big|f_k,\sigma_k^2$$
$$\sim\mathcal{N}(\big[f_k(t_1),\ldots,f_k(t_1),\ldots,f_k(t_p),\ldots,f_k(t_p)\big],\sigma_k^2 I_{N_k p}),$$

where $\big[f_s(t_1),\ldots,f_k(t_1),\ldots,f_k(t_p),\ldots,f_k(t_p)\big]^T$ contains N_k replicates of each $f_k(t_j)$.

Now, we can define a Gaussian process prior,

$$\big[f_k(t_1),\ldots,f_k(t_1),\ldots,f_k(t_p),\ldots,f_k(t_p)\big]^T|a_k,l_k\sim\mathcal{N}(\mathbf{0},\mathrm{cov}^{(k)}).$$

Here $\mathrm{cov}^{(k)}$ is an $N_k p\times N_k p$ matrix that is composed of smaller block matrices,

$$\begin{pmatrix}[\mathrm{cov}(t_1,t_1)] & \cdots & [\mathrm{cov}(t_1,t_p)]\\ \vdots & \ddots & \vdots\\ [\mathrm{cov}(t_p,t_1)] & \cdots & [\mathrm{cov}(t_p,t_p)]\end{pmatrix},$$

where $\big[\mathrm{cov}(t_i,t_j)\big]$ denotes i,j-th a smaller matrix structure. Here each matrix is symmetric and positive definite. This enable us to specify the following likelihood function within each cluster k,

$$\Big[x_1^{(k)}(t_1),\ldots,x_{N_k}^{(k)}(t_1),\ldots,x_1^{(k)}(t_p),\ldots,x_{N_k}^{(k)}(t_p)\Big]\Big|a_k,l_k,\sigma_k\sim\mathcal{N}(\mathbf{0},\mathrm{cov}^{(k)}+\sigma_k^2 I_{N_k p}).\quad(6.5)$$

Likelihood for a Multinomial model.

For convenience in this section we will describe a multinomial model (Kirk et al., 2012) to capture categorical data. Typically, a categorical dataset consists of a list of genes (objects) where measurements, $r\in\{1,\ldots,R\}$, for each gene are taken at Q distinctive attributes (e.g. transcription factors, proteins etc). For genes that tend

to cluster together, x_{rq} denotes the number of times q-th attribute receives a value r. Thus, the multivariate probability mass function for categorical data,

$$p(\mathbf{x}_q|\theta_{1q},\ldots,\theta_{Rq}) \propto \prod_{r=1}^{R} \theta_{rq}^{x_{rq}}, \quad \sum_{r=1}^{R} \theta_{rq} = 1,$$

where $\mathbf{x}_q = \left[x_{1q},\ldots,x_{Rq}\right]$; and θ_{rq} denotes the cluster related probability for attribute q to receive a value r.

Setting a Dirichlet prior, $\mathscr{D}(\beta_{1q},\ldots,\beta_{Rq})$, for $\theta_{1q},\ldots,\theta_{Rq}$, we obtain

$$p(\mathbf{x}_q|\beta_{1q},\ldots,\beta_{Rq}) = \frac{\Gamma(B_q)}{\Gamma(S_q+B_q)} \prod_{r=1}^{R} \frac{\Gamma(x_{rq}+\beta_{rq})}{\Gamma(\beta_{rq})},$$

where $B_q = \beta_{1q},\ldots,\beta_{Rq}$ and $S_q = x_{1q},\ldots,x_{Rq}$. From the independence between the attributes follows the marginal likelihood function,

$$f(\mathbf{x}_1,\ldots,\mathbf{x}_Q|\boldsymbol{\beta}) = \prod_{q=1}^{Q} \frac{\Gamma(B_q)}{\Gamma(S_q+B_q)} \prod_{r=1}^{R} \frac{\Gamma(x_{rq}+\beta_{rq})}{\Gamma(\beta_{rq})},$$

where $\boldsymbol{\beta}_{R\times Q}$ is a matrix composed of hyper-parameters from Dirichlet prior (for further details see supplementary material in (Kirk et al., 2012)).

6.3 Ways of integrating results from different sources

Approach 1

Below we describe the methodology that enables us to integrate the results from Bayesian clustering approaches. Here, Bayesian clustering refers to algorithm that produce a set of samples from a posterior distribution.

Hence, we begin with a collection of R datasets, $D_1,...,D_R$. Each dataset, D_r, contains M_r samples taken from a posterior distribution for component indicator variables, $\mathbf{c}^r =$

$\left(c_1^r, \ldots, c_N^r\right)$. Note, it is not necessary that the number of samples M_r for a dataset D_r should match the number of samples M_l for dataset D_l. However, the only requirement is that all datasets must contain the same data points (genes). Thus,

1. we randomly sample a clustering $\mathbf{c}^r$ from each dataset D_r, $r = 1, \ldots, R$;

2. for each selected clustering we construct an adjacency matrix,

$$(A^{(r)})_{ij} = \begin{cases} 1, & \text{if} \quad c_i^{(r)} = c_j^{(r)}; \\ 0, & \text{otherwise,} \end{cases}$$

   where every non-zero entry represents two data points being associated with the same cluster label;

3. next, we compute a Hadamard product (Fill et al., 2005) (an entry wise product) of all adjacency matrices,

$$H = A^{(1)} \circ \cdots \circ A^{(R)};$$

   Here, the outcome is an adjacency matrix as well; the effects of Hadamard multiplication are illustrated in Figure 6.2**D**;

4. finally, we reconstruct a clustering $\hat{\mathbf{c}}$ that represents the adjacency matrix H.

Dealing with the adjacency matrices (Zhang and Horvath, 2005) is routine in graph theory and biological network analysis. These matrices contain the information about the connections between the data points. For this reason, the above sequence of steps can be also interpreted as comparison of several networks and refinement of those edges that exist across all of the datasets.

Now, repeating the above procedure, (1–4), p times will result in a collection of clusterings $\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_p$. These will represent the “posterior" samples of clusterings that contains

indexes of all genes that cluster together across multiple datasets. Thus, the final clustering $\bar{\mathbf{c}}$ can be identified as described in (Fritsch and Ickstadt, 2009) by constructing a posterior similarity matrix of $\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_p$ and maximising the posterior expected adjusted Rand index (Hubert and Arabie, 1985) (see chapter on theoretical background, section (2.5)).

Approach 2

An alternative way to combine the information from independent clustering outcomes and to identify the genes that cluster across all datasets is the following,

1. for each dataset D_r, $r = 1, \ldots, R$ we identify the overall clustering that maximises the posterior expected adjusted Rand index (Fritsch and Ickstadt, 2009), $\bar{\mathbf{c}}^1, \ldots, \bar{\mathbf{c}}^R$;

2. for each clustering $\bar{\mathbf{c}}^r$ we construct an adjacency matrix

$$(A^{(r)})_{ij} = \begin{cases} 1, & \text{if} \quad \bar{c}_i^{(r)} = \bar{c}_j^{(r)}; \\ 0, & \text{otherwise}; \end{cases}$$

3. next, we compute a Hadamard product between all adjacency matrices $A^{(1)}, \ldots, A^{(R)}$,

$$H = A^{(1)} \circ \cdots \circ A^{(R)};$$

4. and finally, we reconstruct $\hat{\mathbf{c}}$ which corresponds to the adjacency matrix H. The component indicator variables within $\hat{\mathbf{c}}$ will enable us to assign the data points to clusters that are shared across all datasets.

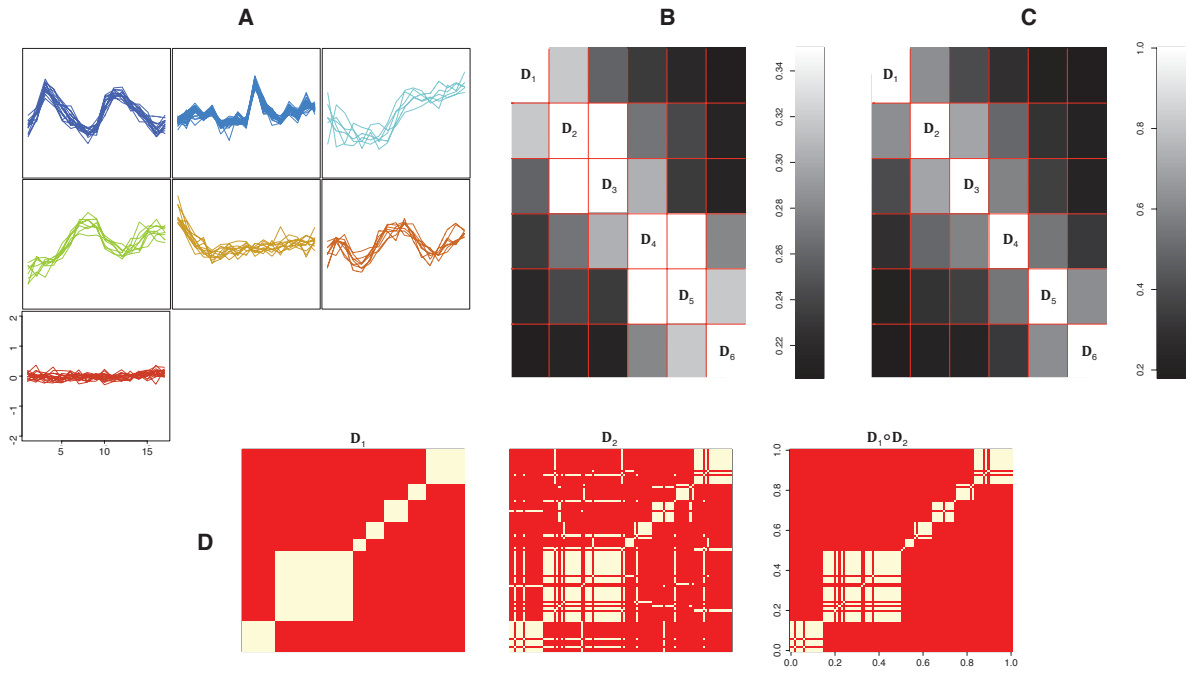


Figure 6.2: Example with six artificial datasets. **(A)** Illustrated genes from Cho et al. (1998) classified into seven clusters with DPM model with GP likelihood. **(B)** A heatmap illustrating the similarity between final clusterings from all 6-datasets using our similarity measure $S(D_r, D_l)$. **(C)** Illustrated the similarity between the same clusterings using ARANDI. **(D)** Illustration of the Hadamard product of two adjacency matrices that correspond to the final cluster assignments for original dataset, D_1, and the first modified dataset, D_2.

6.3.1 Heuristic score of clustering similarity

It might be useful to measure the compatibility between data sources by comparing independent clusterings (before data integration) with a clustering obtained after the integrative modelling. Due to the nature of our technique it is expected to observe more clusters after the data integration process. This is particularly true when studying less related datasets and when integrating more than a few data sources. Therefore, we can define a measure of similarity between any two data sources in terms of number of clusters,

$$S(D_r, D_l) = \frac{(K_{D_r} + K_{D_l})/2}{K_{D_r \circ D_l}},$$

here K_{D_r} and K_{D_l} correspond to the number of clusters before data integration, and $K_{D_r \circ D_l}$, the number of clusters after integration. Here, the measure S can have any value between zero (noninclusive) and one (inclusive). For this reason the closer S is to 0, the more dissimilar datasets are; and hence, it might be worth considering to remove one of the “noisier" datasets from the study (by “noisier dataset" we refer to the data source that strongly influence how many clusters we observe after fusion step; this e.g. can be the effect caused by a certain clustering algorithm or non-informative data itself). On the other hand, the closer S is to 1, the more substantial the similarity is between both datasets, and if $S = 1$ then both datasets are clustered identically and integrative modelling is unnecessary.

6.4 Applications and results

In this section we explore the performance of our data integration methodology and compare both techniques: *Approach 1* (section 6.3) and *Approach 2* (section 6.3). Then, we test the performance on a set of popular examples from the literature and on a new dataset from *sporadic Inclusion Body Myositis*.

6.4.1 *Saccharomyces cerevisiae* time course dataset

To begin with let consider a *S. cerevisiae* dataset from (Cho et al., 1998), that contains *m*RNA transcription levels taken to study the cell cycle. From 416 genes that had previously been identified to have periodic changes in transcript levels we select 100 (as described in (Kirk et al., 2012)), and applied DPM with GPR likelihood model to perform clustering on this dataset. The details regarding MCMC specification and diagnostics are summarised in Appendix section A.2.1. In order to sort genes in to appropriate cluster, we further applied Fritsch and Ickstadt (2009) criterion that pro-

vided us with a single clustering estimate. Figure 6.2**A** illustrates *S. cerevisiae* genes sorted into seven clusters. To demonstrate the performance of our methodology, we further consider a *6–dataset* example from Kirk et al. (2012). The example consists of six data sources, where the first source is the original dataset from Figure 6.2**A**, and the other five were obtained sequentially, by randomly permuting a quarter of gene names with gene time courses. Next, we applied *Approach 1* on pairwise combinations of these datasets to identify the numbers of genes that cluster together and used our compatibility score to determine the similarities across all 6 datasets. The pairwise similarities are summarised in Figure 6.2**B** where columns and rows identify which combination of datasets were considered, and colour illustrates the level of similarity. Alternatively, the similarity between these datasets can be identified from the final clusterings by computing the adjusted Rand index (ARANDI) (Hubert and Arabie, 1985). The ARANDI compares two given partitions of the same list of genes and is based on how often a gene (observation) is associated with the same cluster in both partitions (see Figure 6.2**C**). Similarly, applying methodology (6.3) we can obtain identical data integration results and similarity measures.

We also tested an alternative approach (*Approach 3* discussed in section 6.3), where we compute the Hadamard product between posterior similarity matrices rather than adjacency matrices. Here we used the outcome of product wise entry, H, to maximise the posterior expected adjusted Rand index. We found that both *Approach 1* and *Approach 2* give identical results while this alternative *Approach 3* produce slightly different results (figures are not shown here), and for this reason we will continue using only *Approach 1* and *2* throughout this chapter.

6.4.2 Integrating cell cycle datasets

In this section we compare the results from our approach (6.3) to the results by *Multiple data integration* (MDI) (Kirk et al., 2012). We consider integrating two different

combinations of datasets from yeast cell cycle studies. The first dataset contains gene expression time courses (see Granovskaia et al., 2010), where *m*RNA measurements are taken at 41 time points across 551 genes that exhibit oscillatory expression profiles. The second dataset is ChIP-chip data from Harbison (2004) that contains binary information about proteins binding to DNA.

Applying the independent DPM models with Gaussian process likelihood to the gene expression time courses, and multinomial likelihood to the transcription factor binding data we construct the adjacency matrices from posterior clusterings. Then, computing the Hadamard products allow us to extract the final allocation variable that contains indices of genes that cluster together in both datasets. As before, further details on MCMC specification and diagnostics are given in Appendix section A.2.2.

In this example we are aiming to compare the results from our method to the results from MDI. MDI jointly clusters all datasets by modelling the dependencies between them; the final clustering can be extracted by calculating the probability that any two genes are fused and removing the genes where this probability is less than 0.5. For this reason, in our case we can consider removing genes that lack the evidence of clustering together. This can be achieved by computing the matrix $\mathbf{P} = \frac{1}{p}\sum_{q=1}^{Q} H_q$, where each matrix entry is probability, P_{ij}, for gene i and gene j to be in the same cluster in both gene expression and transcription factor binding datasets; and removing genes i, where $P_{ij} < 0.5$ for every j. The above procedure is somewhat analogous to MDI in terms of looking only at those genes that are fused across both datasets. Then, applying the Fritsch and Ickstadt (2009) methodology on the filtered posterior clusterings we obtain the final clustering that assigns genes into clusters based on both datasets. Figure 6.3 illustrates the performance of our *Approach 1*, where genes are allocated into clusters based not only of their expression profiles but also on which transcription factors binds to DNA. This means, our approach enabled us to elucidate those sets of genes that are regulated by the same transcription factors. Here, we identified 10

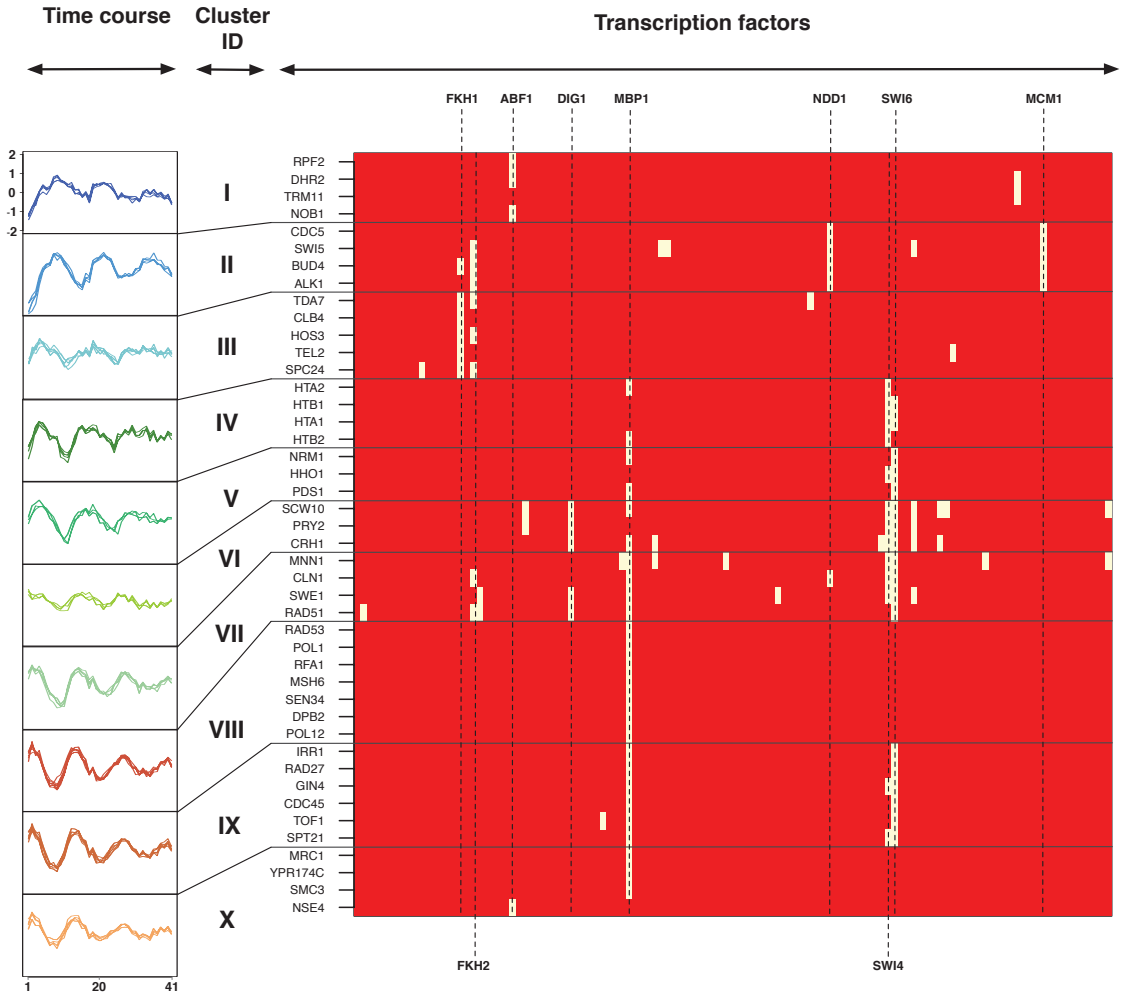


Figure 6.3: Incorporating yeast cell cycle time course and transcription factor (TF) binding datasets with *Approach 1*. On the left – gene expression time courses from Granovskaia et al. (2010) are classified into 10 clusters and then projected on a heatmap of TF binding data from Harbison (2004) on the right. Here yellow colour corresponds to a value 1, and illustrates that gene is regulated by a TF; and red – corresponds to 0 value, which means that there is no regulation/binding. A key TF, which agree with MDI's results, are highlighted with black dashed guidelines. Horizontal black lines mark cluster boundaries.

clusters that consist of a total of 44 genes (MDI: 48 genes); on the left in Figure 6.3 are illustrated the gene expression time courses and on the right – the ChIP-chip dataset. The comparison between *Approach 1* and MDI's output is summarised in Figure 6.4**A** and **B**. Here each clustering is presented as an adjacency matrix, where yellow colour means that two genes are clustering together, and red – that they do not cluster. For

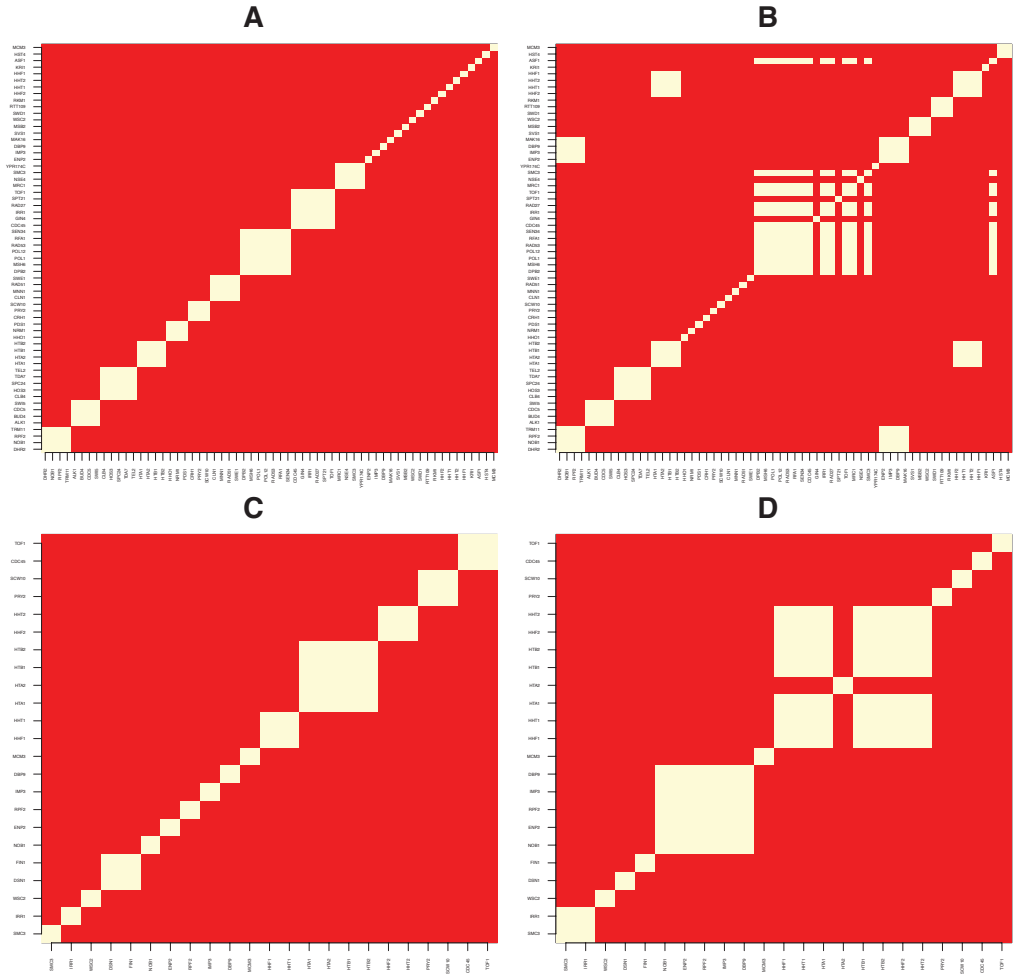


Figure 6.4: All figures show the final clustering as an adjacency matrix, here yellow colour illustrate two genes that appear in the same cluster, and red colour indicate that two genes are assigned in to different clusters. **(A)** Final clustering obtained using *Approach 1* for GE and TF example. **(B)** Final clustering obtained using MDI for GE and TF example, here genes are sorted according to clustering in **(A)**. **(C)** Final clustering obtained using *Approach 1* for GE, TF and PPI example. **(D)** Final clustering obtained using MDI for GE, TF and PPI, here genes are sorted according to clustering in **(C)**.

easier comparison clustering in **B** is sorted according to clustering in **A**.

The second step of our yeast cell cycle example considers integrating three datasets: gene expression, transcription factor binding and protein-protein interaction (PPI). In order to obtain PPI dataset, we select matching genes from BioGRID (Stark et al., 2006) as described by Kirk et al. (2012), and cluster them using the DPM model with

the multinomial likelihood function. Then, applying *Approach 1* we can incorporate the PPI clustering outcome with the gene expression and TF binding results. Again, thinning out the genes that lack evidence for clustering together, we obtain a set of 14 genes that can be assigned into 6 clusters (MDI – 16 genes assigned into 5 clusters). The comparison between *Approach 1* and MDI's output is summarised in Figure 6.4**C** and **D** respectively. Here each clustering is presented as an adjacency matrix, where yellow colour indicates that two genes are clustering together, and red – that they are in different clusters. For easier comparison clustering in **D** is sorted according to clustering in **C**. Figure 6.5 illustrates all genes that clusters together across three datasets. For convenience, appendix A Tables A2 and A3 contain further details regarding gene function.

In this example we have shown that it is not necessary to consider and explicitly model the dependencies that exist between data sources in order to achieve comparable results. This example demonstrates that our data fusion technique is a competitive tool that can be applied for studying underlying regulatory processes at the molecular level, and can be an alternative to MDI. Equally, in our case it was not required to rerun all data pre-processing (clustering) in order to further consider PPI dataset.

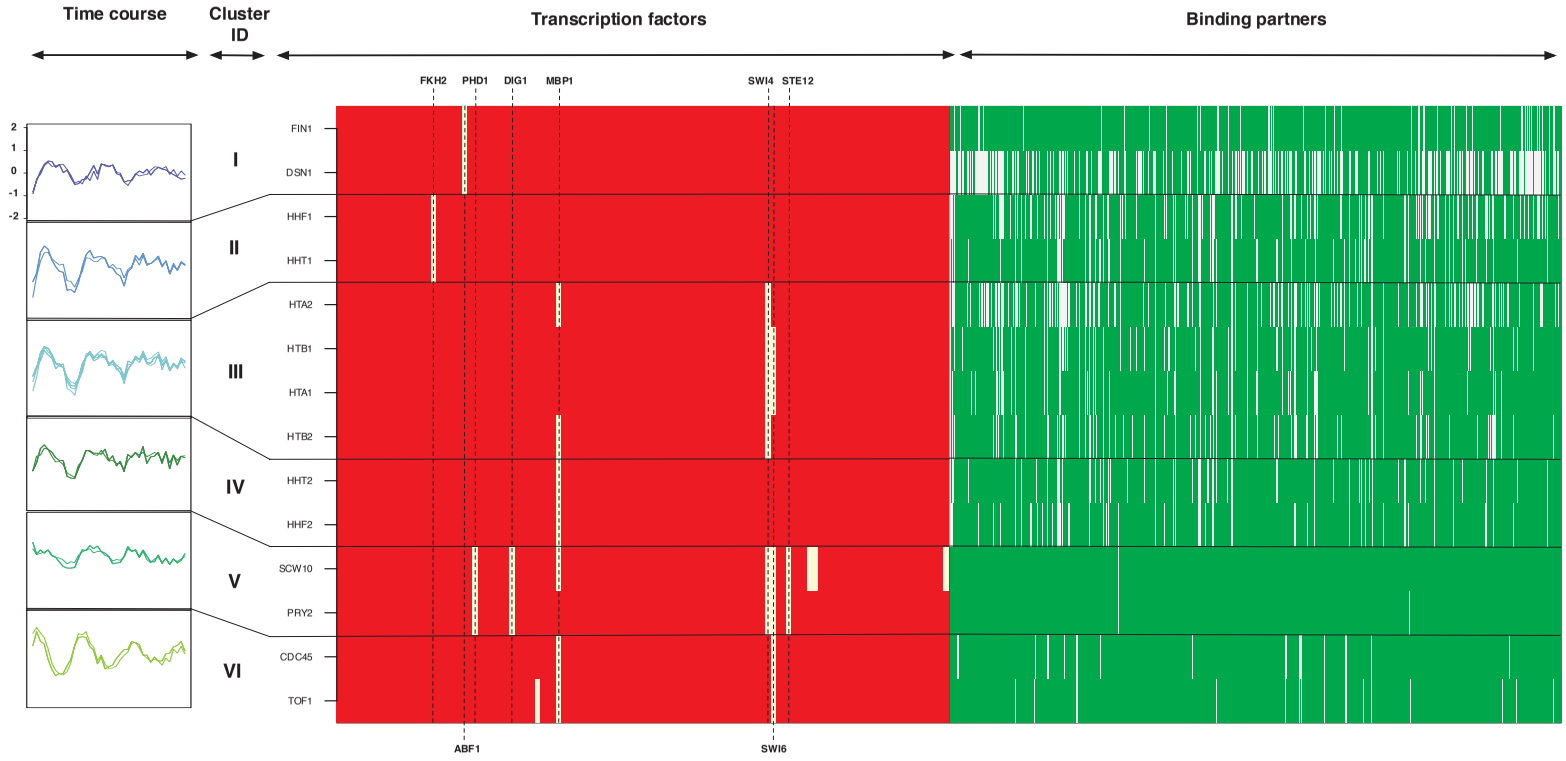


Figure 6.5: Integrating yeast cell cycle datasets with *Approach 1*. On the left, gene expression time courses from Granovskaia et al. (2010) are classified into 10 clusters and projected on a heatmap of TF data from Harbison (2004), and then on PPI data. A key TF, which agree with MDI's results, are highlighted with black dashed guidelines. Horizontal black lines mark cluster boundaries.

6.4.3 Breast cancer data

In this example we explore the performance of our data integration technique on a breast cancer dataset. We aim to integrate four different data sources taken from *The Cancer Genome Atlas* (Cancer Genome Atlas Network and others, 2012). For this reason we will use a dataset that was previously described in (Lock and Dunson, 2013). The preselected data consist of 348 tumour samples taken across four datasets: RNA gene expression (645 genes), DNA methylation (574 probes), *m*iRNA expression (423 *m*iRNAs) and reverse phase protein array (171 proteins). In order to cluster all data sources, we adopt a modified version of Bayesian consensus clustering (Lock and Dunson, 2013). BCC is data integration technique that seeks to simultaneously model data specific and shared features by inferring the overall clustering $\hat{C}$ (that describes all datasets) and by inferring data specific clusterings $\hat{L}_i, i = 1, \ldots, 4$. The source specific clustering is controlled by parameter $\alpha = [\alpha_1, \ldots, \alpha_4]$, which express the probability of how much each L_i contributes to the overall $\hat{C}$. Our goal is to cluster each dataset independently without inferring the overall clustering $\hat{C}$. For this reason we fix the probability $\alpha = 1$ and perform BCC individually on each genomic dataset (using publicly available R code). Next, applying *Approach 1* on posterior samples L_i we can identify the overall clustering $\bar{\mathbf{c}}$.

Breast cancer is a heterogeneous disease, for this reason four biologically distinct molecular subtypes where connected to these data sources (Cancer Genome Atlas Network and others, 2012; Lock and Dunson, 2013). They are known as *Her2*, *Basal*, *Luminal A*, *Luminal B*, and are associated with different clinical prognosis (Dawood et al., 2011; Rakha et al., 2008). In order to assess our results we can identify cancer subtypes that are associated with each cluster and compare these to the BCC clusters. Table 6.1 illustrates the summarised results. In the second column we present the outcome from BCC (a single run of publicly available R code) and in column three are given results from our *Approach 1*. It can be seen that clusters identified by our tech-

Table 6.1: Comparison between BCC overall and *Approach 1* final clusterings. In the table are given numbers of tumour samples per cluster; e.g. our cluster 3 contains 6 samples of *Her2*, 65 samples of *Basal* and 4 samples of *Luminal A*, for this reason, cluster 3 can be summarised as containing mostly *Basal* type tumours.

	BCC			*Approach 1*		
TCGA tumor subtypes	Cl_1	Cl_2	Cl_3	Cl_1	Cl_2	Cl_3
Her2	20	6	13	5	28	6
Basal	4	2	**66**	1	6	**65**
Luminal A	**81**	**76**	4	**59**	**98**	4
Luminal B	**73**	3	0	**59**	17	0

Note: Table contains clusters that are classified by particular cancer subtype using publicly available R code.

nique can be described by similar cancer subtypes when compared to BCC outcome (e.g. Cl_3 in our case contains mostly *Basal* type tumour samples (65 in total), and this corresponds to Cl_3 in BCC analysis; our Cl_2 can be described by *Luminal A* subtype, and in BCC case Cl_2 is a similar cluster. Furthermore, both cluster, Cl_1, from our method and BCC cluster, Cl_1, contains tumour samples from *Luminal A, B* subtypes).

We compared our method to the BCC final outcome. Although our method does not model the relationships between data specific and overall clustering, the final data integration outcome lead to very similar results. For this reason our method can serve as an alternative.

6.4.4 Sporadic inclusion body myositis

In this section we apply our technique on clinical gene expression datasets that include: (i) *sporadic Inclusion Body Myositis* (sIBM), which is an inflammatory muscle disease that progress very slowly, cause muscular weakness and eventually muscle atrophy (Grau and Selva-O'Callaghan, 2008; Machado et al., 2009); (ii) *polymyositis* (PM) which causes chronic inflammation of the muscles; and (iii) a dataset containing human protein-protein interactions (BPPI). Both diseases are associated with ageing but

interestingly sIBM can be frequently misdiagnosed as PM, and the explicit diagnosis can only be confirmed via a muscle biopsy (Dalakas, 2006). Current understanding is that sIBM is driven by two coexisting processes (autoimmune and degenerative); however, the actions by which sIBM occurs are still only poorly understood (Dalakas, 2006; Needham and Mastaglia, 2007).

PM and sIBM case. Here we are focusing on the experimental sIBM and PM datasets that have 5 and 3 data points (clinical cases). In order to apply our technique, we select 424 genes that where previously identified to have the largest variation in their expression across all data points (Thorne et al., 2013). In order to cluster these datasets, we employ "*mclust*" package in R, which fits a Gaussian mixture model and uses the Bayesian information criterion to estimate the number of components. Because we do not have access to the clustering samples from the posterior for each dataset, we use *Approach 2* to fuse single clusterings from both datasets. To validate our results we used a web-based tool called "ToppGene"[1] that performs gene set enrichment analysis. This allowed us to detect functional enrichment for phenotype (disease) for each cluster. When compared to independent clustering of each dataset, the integrative analysis enabled us to identify those clusters (7 out of 28) that are mostly enriched with diseases like rheumatoid arthritis, recurrent bacterial infections or myositis. This suggest that genes in these clusters (especially in the myositis

[1] https://toppgene.cchmc.org

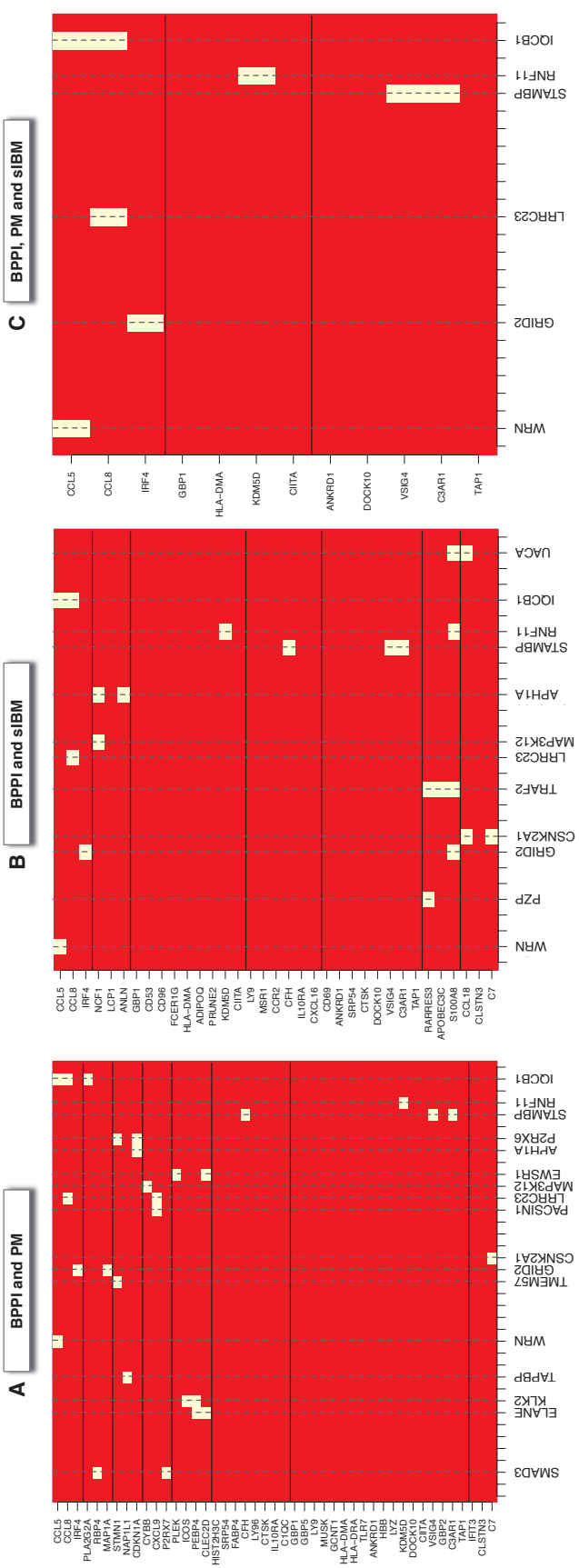


Figure 6.6: Integrating biomedical myositis datasets with *Approach 2*. All heatmaps illustrate the results for PM, sIBM and BPPI. Genes on the left are classified into clusters; black horizontal lines mark cluster boundaries. On the bottom are listed protein names; here, yellow colour indicates binding and horizontal grey lines guides toward the corresponding protein name. **(A)** Integration performed on PM and BPPI datasets. **(B)** Integration performed on sIBM and BPPI datasets. **(C)** Integration performed on all three datasets.

Table 6.2: A list of proteins that interact with human genes used in our study. "+" indicates whether a particular interaction is present across various combinations of PPI, PM and sIBM datasets.

Protein	PPI∘PM	PPI∘sIBM	PPI∘PM∘sIBM	Protein description
SMAD3	+			SMAD family member 3
ELANE	+			elastase, neutrophil expressed
KLK2	+			kallikrein-related peptidase 2
TAPBP	+			TAP binding protein (tapasin)
WRN	+	+	+	Werner syndrome, RecQ helicase-like
PZP		+		pregnancy-zone protein
TMEM57	+			transmembrane protein 57
GRID2	+	+	+	glutamate receptor, ionotropic, delta 2
CSNK2A1	+	+		casein kinase 2, alpha 1 polypeptide
TRAF2		+		TNF receptor-associated factor 2
PACSIN1	+			protein kinase C and casein kinase substrate in neurons 1
LRRC23	+	+	+	leucine rich repeat containing 23
MAP3K12	+	+		mitogen-activated protein kinase kinase kinase 12
EWSR1	+			EWS RNA-binding protein 1
APH1A	+	+		APH1A gamma secretase subunit
P2RX6	+			purinergic receptor P2X, ligand-gated ion channel, 6
STAMBP	+	+	+	STAM binding protein
RNF11	+	+	+	ring finger protein 11
IQCB1	+	+	+	IQ motif containing B1

cluster) might play an important and shared role in both diseases, PM and sIBM. For this reason they could be subject for further analysis.

BPPI, PM and sIBM case. The initial BPPI dataset was downloaded from the *Human Protein Reference Database* (http://www.hprd.org/), which contains all available binary protein-protein interactions in humans (HPRD_Release9_041310.tar.gz on 14 of April 2014). First of all, we symmetrized the BPPI dataset matrix; then, we removed all genes (rows) that do not correspond to the set of genes of interest (424). This preprocessing step resulted in a final list of 118 genes that we used for further analysis. Similarly, we have removed all columns in the BPPI matrix that had less than three non-zero entries; this provided us with a total of 49 columns. The BPPI dataset is a binary matrix, and for this reason it can be clustered using DPM with the multinomial likelihood model and the final clustering can be identified applying the Fritsch and

Ickstadt (2009) approach. In order to integrate all data sources (BPPI, PM and sIBM) we applied *Approach 2*, Figure 6.6 illustrates the results and Table 6.2 summarise which proteins have a binding partners among study genes across integrated datasets. Here the integrative modelling allowed us to identify a set of proteins (*WRN, GRID2, LRRC23, STAMBP, RNF11, IQCB1*) that have similar binding partners across both diseases, PM and sIBM. These interactions could potentially be studied further in order to better understand the underlying similarities between both diseases.

6.5 Discussion and conclusions

In this chapter we have proposed an alternative way to model data generated from different sources. Our main technique relies on the outcomes from Bayesian nonparametric clustering approaches and is based on graph theory concepts. We have demonstrated that both approaches, *Approach 1* and *Approach 2*, can give similar results (see example 6.4.1). Equally, while *Approach 1* can fuse the outcomes from Bayesian clustering algorithms, *Approach 2* can be applied in order to fuse the single clusterings across various datasets. This makes *Approach 2* compatible with non-Bayesian clustering methods.

Here we have demonstrated the applicability of our technique to a variety of biological problems: the identification of potentially underlying regulatory mechanisms in the yeast cell cycle, subtyping tumours in breast cancer data, and exploring similarity patterns across inflammatory muscle diseases. We have compared our technique to MDI and BCC, which are currently used to address similar data integration problems. Unlike Bayesian data integration or Bayesian consensus clustering, our graph theoretical approach to data fusion is an exploratory tool that does not model the relationships between all data sources explicitly, but instead extracts shared structures after a clustering step. By adopting this post-processing strategy this method might loose some

accuracy when dealing with an increasing number of datasets or data sources that are very disparate, for this reason data integration might lead to a large number of clusters. However, the main benefits of our graph-theoretical approach include: (i) the applicability to Bayesian and non-Bayesian type clustering approaches. This means that our methodology can be applied in order to model multiple sources on a genome scale data without facing computational challenges; and (ii) ability to perform clustering in a parallel fashion. Such feature might be favourable in situations where it is necessary to rerun computations in order to consider additional datasets. As part of our modelling approach, we have defined a measure of similarity between two data sources. This, for example, could be used to evaluate the effects of data integration routine and in order to assess the agreement between data sources prior to performing e.g. model based data fusion.

As we have stated before, it is possible to apply our graph-theoretical approach to the outcomes from simple clustering techniques (for example hierarchical or k-means clustering), and here we will discuss it in more details. This could be done by first performing a bootstrapping approach on each gene within a dataset D_r, M times; for further details on bootstrapping see Efron (1981) and Kerr and Churchill (2001). This would produce a set of bootstrapped datasets $D_r^{b_1}, ..., D_r^{b_M}$. The bootstrapped datasets together with the initial dataset can be clustered using a standard hierarchical clustering algorithm. The clustering outcomes can to some extent be viewed as being the analogues to the samples from posterior. Then after bootstrapping all data sources we can use our *Approach 1* to perform integrative modelling. This process can be seen as a frequentist modelling approach to data integration.

All data integration tools that largely rely on clustering might have certain limitations. For example, such techniques might not be able to consider genes whose measurements are missing; equally, we cannot associate the interaction (regulation) probabilities between e.g. gene and TF. For this reason network analysis might potentially be

a more appropriate tool for quantifying or detecting underlying regulatory reactions and establishing probabilistic dependencies between cluster genes/TFs/proteins.

Chapter 7

Conclusion

7.1 Summary

Throughout this thesis we were considering the applicability of modern developments in the field of Bayesian statistics, and how recent theoretical advances can be translated into biological applications.

In chapter 3 we proposed a new theoretical approach to model gene expression data. We were considering a technique that employs recent results from nonparametric Bayesian statistics to capture the relationships between several variables. This approach enabled us to consider and model the dependencies between proteins and *m*RNA. We demonstrated that the importance of this approach becomes evident when considering data that has sparsely sampled regions. Employing multipe-output Gaussian process (MGP) regression as a modelling tool enabled us to improve predictions and impute the regions where measurements were missing.

In chapter 4 we used multipe-output Gaussian processes as a foundation to model time-varying metabolic species. Based on this dependent Gaussian processes model we constructed the derivative process model and proposed a new approach to capture

and predict the temporal behaviour of metabolic fluxes.

In chapter 5 we further explored the capabilities of multipe-output Gaussian process models by applying this approach to oscillatory systems. We tested the performance of the MGPs model on signals that have different frequencies and phase parameters under various sampling schemes. This allowed us to better understand under what circumstances we can accurately reconstruct the underlying function and when the reconstruction is impossible due to the Nyquist limit.

In addition to our research on Gaussian processes, in chapter 6 we were concerned with Dirichlet processes. We demonstrated that the usefulness of this nonparametric approach becomes evident when analysing large functional genomic datasets (e.g. gene expression time courses, transcription factor binding) and performing clustering. Based on the outcomes of Bayesian nonparametric clustering technique we proposed a new graph-theoretical approach to perform integrative modelling across various genomic datasets. We demonstrated that our technique can be applied in order to study different biological problems, for example this technique can be employed in order to study a list of genes and look for a potential regulatory mechanisms; equally, we can subtype tumour samples (patients) based on different genomic measurements (e.g. *mi*RNA, DNA methylation). Further, in parallel we discussed and shown the compatibility of our graph-theoretical approach with other well established clustering techniques.

In this thesis we explored the applicability of Bayesian nonparametric techniques to address a broad range of biological problems; we showed that Bayesian nonparametric techniques can allow us to generate new hypotheses about the underlying regulatory processes. Furthermore, these methods enable us to complement the mechanistic modelling tools and provide further insights into biological data.

7.2 Potential future work

Addressing the uniqueness question in metabolic flux analysis

Flux balance analysis (FBA) is a commonly applied mathematical approach for modelling metabolism. By adding constrains to the stoichiometric analysis of a metabolic system FBA is able to determine the metabolic fluxes at steady state. Although FBA does not require the specification of kinetic parameters, which might be difficult to obtain, and is very simple to implement the main problems, however, are that FBA does not uniquely specify fluxes (but instead has to invoke additional criteria and constraints, such as optimality of biomass production) and that it cannot be used for modelling the dynamical behaviour of fluxes. A new approach based on non-parametric Bayesian techniques potentially could enable the extension of conventional FBA to temporally varying flux data. In contrast to traditional approaches to metabolic flux estimation, which employ uniform sampling of the space (Price et al., 2004), such technique could allow us to capture the temporal evolution of flux dynamics at successive time-points as well (see chapter 4). For this reason future work could make use of previously described multiple–output Gaussian processes (section 2.7, chapter 4, Appendix B) in order to extend the conventional FBA and to further address the flux uniqueness question as well as capture the dynamical flux behaviour.

Appendix A

A Supplementary material for chapter 6

A.1 Introduction

In this supplementary material we provide the additional results that are not included in the main chapter. Section (A.2) explains the inference of hyper–parameters used in Dirichlet process mixture (DPM) model and provides further MCMC running details. Section A.3 contain supplementary tables for *Yeast cell cycle* examples that are omitted in the main chapter.

A.2 Inference of the hyper-parameters

In this section we explain how inference is performed for DPM models. As well, we provide the Markov chain Monte Carlo (MCMC) running details for *S. cerevisiae* and *yeast cell cycle* examples (see sections (6.4.1 and 6.4.2) in chapter 6).

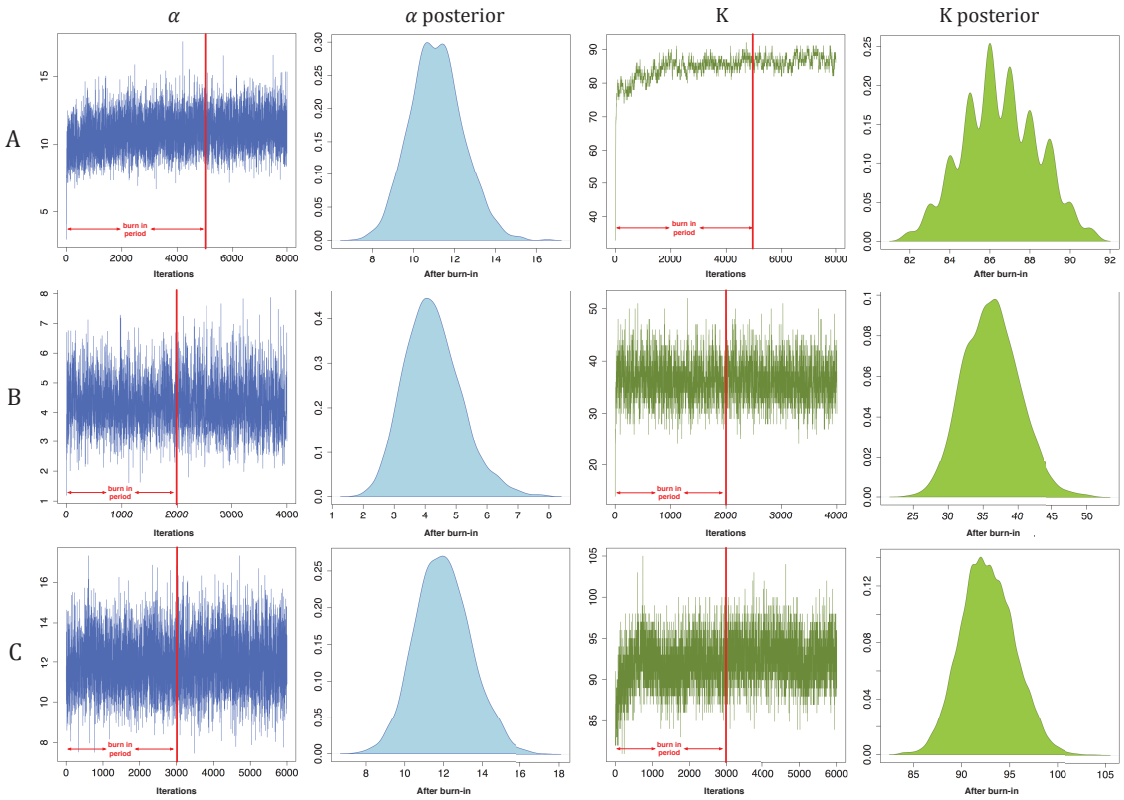


Figure A.1: MCMC diagnostics plots. In the first column figures illustrate a “thinned-out" Markov chains for the concentration parameter, α; in the second column are given posterior distributions for α; in the third column – a “thinned-out" Markov chains for the number of clusters, K; in the fourth – posterior distributions for K. **(A)** Time course datasets from Granovskaia et al., 2010. **(B)** Transcription factor binding dataset from Harbison, 2004. **(C)** Protein-protein interaction dataset from BioGRID (Stark et al., 2006).

In *DPMSysBio* package, the Gaussian process likelihood is controlled by the following three hyper-parameters $\theta_c = \{a_c, l_c, \sigma_c\}$ that are necessary in order to learn the means and covariances of each cluster. We set a Gaussian priors for the logarithmic versions of hyper-parameters ($\log(\theta_c)$) and employ a Gibbs sampling algorithm as described in Neal (2000) (see *section 6* for details). This algorithm can be seen as the most general Gibbs sampling scheme that can deal with a non-conjugate priors. In order to learn the hyper-parameters of DPM model with multinomial likelihood, we use Dirichlet priors.

Finally, to infer the concentration parameter α we set the following *gamma* prior $\alpha \sim \mathcal{G}(2, 4)$, and adopt approach proposed by Escobar and West (1995).

A.2.1 Saccharomyces cerevisiae example

For the original time courses (Cho et al., 1998) and five perturbed datasets, we ran *DPMSysBio*[1] package with GPR likelihood for 50000 iterations recording each 5th sample. This provided us with 10000 “thinned-out" MCMC samples. Further, we discarded the first 5000 samples as a “burn-in" period and for further analysis we used 5000 samples per each dataset.

A.2.2 Yeast cell cycle datasets example

- For time course dataset (Granovskaia), we ran *DPMSysBio* package with GP likelihood for 40000 iterations recording each 5th sample. This provided us with 8000 “thinned-out" MCMC samples. Further, we discarded the first 5000 samples as a “burn-in" period and for further analysis we used 3000 samples.
- For transcription factor binding dataset (Harbison), we ran *DPMSysBio* package with multinomial likelihood for 20000 iterations recording each 5th sample. This provided us with 4000 “thinned-out" MCMC samples. Further, we discarded the first 2000 samples as a “burn-in" period and for further analysis we used 2000 samples.
- For protein-protein interaction dataset (Biogrid), we ran *DPMSysBio* package with multinomial likelihood for 30000 iterations recording each 5th sample. This provided us with 6000 “thinned-out" MCMC samples. Further, we discarded the first 3000 samples as a “burn-in" period and for further analysis we used 3000 samples.

Figure 6.2 illustrates the Markov chains and posterior distributions for the number of clusters, K, and the concentration parameter, α, for Granovskaia (TC), Harbison (TF)

[1] *DPMSysBio* is a Matlab package that performs data clustering. It is available from author Dr. Paul Kirk upon request.

and Biogrid (PPI) datasets.

A.3 *Yeast cell cycle* datasets

Cluster ID	Gene name	Short description
I	FIN1	Spindle pole body-related intermediate filament protein
	DSN1	Essential component of the MIND kinetochore complex
II	HHF1	Histone H4, core histone protein
	HHT1	Histone H3, core histone protein
III	HTA2	Histone H2A, core histone protein
	HTB1	Histone H2B, core histone protein
	HTA1	Histone H2A, core histone protein
	HTB2	Histone H2B, core histone protein
IV	HHT2	Histone H3, core histone protein
	HHF2	Histone H4, core histone protein
V	SCW10	Cell wall protein with similarity to glucanases
	PRY2	Sterol binding protein involved in the export of acetylated sterols
VI	CDC45	DNA replication initiation factor
	TOF1	Subunit of a replication–pausing checkpoint complex (Tof1p-Mrc1p-Csm3p)

Table A.1: List of genes that cluster together across gene expression, ChIP-chip and protein-protein interaction datasets.

Cluster ID	Gene name	Short description
I	RPF2	Involved in the assembly of the 60S ribosomal subunit
	DHR2	Required for 18S rRNA synthesis
	TRM11	Catalytic subunit of an adoMet-dependent tRNA methyltransferase complex
	NOB1	Involved in synthesis of 40S ribosomal subunits
II	CDC5	Polo-like kinase with multiple functions in mitosis and cytokinesis; possible Cdc28p substrate
	SWI5	Transcription factor that activates transcription of genes expresses at the M/G1 phase boundary and in G1 phase; appears to be regulated by phosphorylation by Cdc28p kinase
	BUD4	Involved in bud-site selection; potential Cdc28p substrate
	ALK1	Protein kinase; accumulation and phosphorylation are periodic during the cell cycle
III	TDA7	Cell cycle-regulated gene of unknown function
	CLB4	B-type cyclin involved in cell cycle progression; activates Cdc28p to promote the G2/M transition; may be involved in DNA replication and spindle assembly
	HOS3	Histone deacetylase (HDAC) with specificity in vitro for histones H3, H4, H2A, and H2B
	TEL2	Required for telomere length regulation and telomere position effect
	SPC24	Involved in chromosome segregation, spindle checkpoint activity and kinetochore clustering
IV	HTA2	Histone H2A, core histone protein
	HTB1	Histone H2B, core histone protein
	HTA1	Histone H2A, core histone protein
	HTB2	Histone H2B, core histone protein
V	NRM1	Transcriptional co-repressor of MBF-regulated gene expression
	HHO1	Histone H1, linker histone with roles in meiosis and sporulation
	PDS1	Securin
VI	SCW10	Cell wall protein with similarity to glucanases
	PRY2	Sterol binding protein involved in the export of acetylated sterols
	CRH1	Chitin transglycosylase
VII	MNN1	Alpha-1,3-mannosyltransferase
	CLN1	G1 cyclin involved in regulation of the cell cycle
	SWE1	Protein kinase that regulates the G2/M transition by inhibition of Cd
	RAD51	Strand exchange protein
VIII	RAD53	Protein kinase, required for cell-cycle arrest in response to DNA damage
	POL1	Required for the initiation of DNA replication during mitotic DNA synthesis and pre meiotic DNA synthesis
	RFA1	Subunit of heterotrimeric Replication Protein A (RPA)
	MSH6	Protein required for mismatch repair in mitosis and meiosis
	SEN34	Subunit of the tRNA splicing endonuclease
	DPB2	Second largest subunit of DNA polymerase II (DNA polymerase epsilon)
	POL12	B subunit of DNA polymerase alpha-primase complex
IX	IRR1	Subunit of the cohesin complex
	RAD27	5' to 3' exonuclease, 5' flap endonuclease
	GIN4	Protein kinase involved in bud growth and assembly of the septin ring
	CDC45	DNA replication initiation factor
	TOF1	Subunit of a replication–pausing checkpoint complex (Tof1p-Mrc1p-Csm3p)
	SPT21	Protein with a role in transcriptional silencing
X	MRC1	S-phase checkpoint protein required for DNA replication
	YPR174C	Protein of unknown function
	SMC3	Subunit of the multi protein cohesin complex
	NSE4	Component of the SMC5-SMC6 complex

Table A.2: List of genes that cluster together across gene expression and ChIP-chip datasets.

Appendix B

Flux Balance Analysis and Gaussian Process Constrains

Flux balance analysis

Flux balance analysis (FBA) enable us to make predictions about a production rate of a metabolite of interest and investigates various questions such as what is the maximum growth rate of an organism by calculating the flow of metabolites through the network (Orth et al., 2010). The first step in FBA is to mathematically represent a metabolic system in a matrix form

$$\mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{pmatrix},$$

where m is the number of compounds; and n is the number of reactions. Such matrix S is called a stoichiometry matrix. The stoichiometric coefficients s_{ij} can be positive or negative, meaning that a metabolite was produced or consumed. If a metabolite is not

participating in a particular reaction, then the corresponding coefficient in the matrix **S** is set to zero. The overall flux through all the reactions and metabolite concentrations are represented as vectors

$$\begin{aligned} \mathbf{v} &= \{v_1, ..., v_n\} \quad \text{– overall flux,} \\ \mathbf{x} &= \{x_1, ..., x_m\} \quad \text{– overall concentration.} \end{aligned}$$

Generally, for such a metabolic network the mass balance is given by the following equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{S} \cdot \mathbf{v}.$$

The requirement that the metabolic system is at a steady state,

$$\mathbf{S} \cdot \mathbf{v} = 0, \tag{B.1}$$

defines a system of linear equations. Such system is typically under-determined ($n > m$), and cannot be solved using e.g. a standard Gaussian elimination technique. For this reason additional information is required in order to uniquely determine the fluxes. A collection of all vectors **v** that are solutions to equation (B.1) is called metabolic *genotype*, and mathematically a null space of **S** (Varma and Palsson, 1994). By invoking additional criteria and constraints, such as optimality of biomass production, FBA can identify an optimal solution within a null space. The optimisation is frequently completed by linear programming where it is compulsory to minimise or maximise the objective function $Z = \mathbf{c}^T \mathbf{v}$, here **c** is a vector of weights. This optimal solution is a particular flux distribution under given conditions and is called a metabolic *phenotype* (Varma and Palsson, 1994).

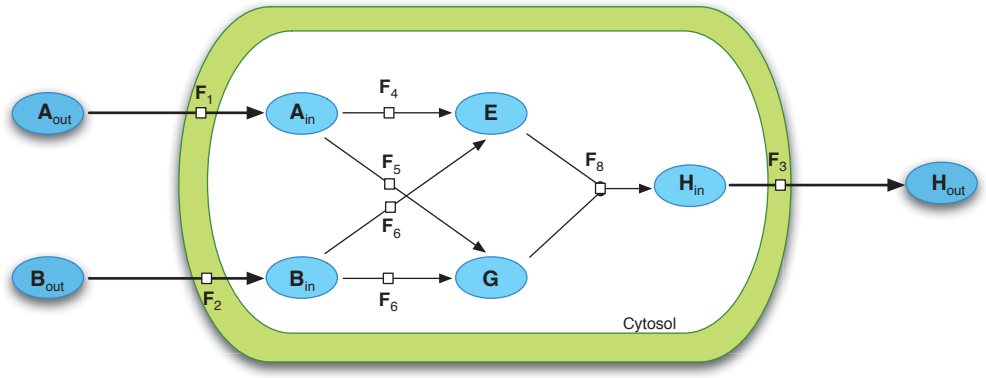


Figure B.1: A schematic illustration of metabolic system/pathway with eight fluxes and five internal compounds.

The role of Gaussian process

Below we show how correlations between fluxes can be captured using a multiple-output Gaussian process, and briefly discuss how predictions about unknown fluxes potentially could be made.

To begin with, let us consider an example of metabolic system, which is summarised in Figure (B.1), consisting of eight fluxes and five internal metabolites. Let further consider that it is possible to obtain only noisy measurements of two incoming fluxes, $F_1(t)$, $F_2(t)$ and one outgoing flux, $F_3(t)$, over some period of time, $T = t_1, .., t_N$; and stoichiometry matrix S is known,

$$S = \begin{array}{c} \\ A \\ B \\ H \\ E \\ G \end{array} \begin{array}{c} \begin{array}{cccccccc} F_1(t) & F_2(t) & F_3(t) & F_4(t) & F_5(t) & F_6(t) & F_7(t) & F_8(t) \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -2 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}.$$

It is clear that the system above is underdetermined and it is not possible to obtain

unique solution in terms of all eight fluxes. The question is whether it is possible to models such system in a multiple–output Gaussian processes framework and to identify all fluxes.

From the stoichiometry matrix S we can identify the relationship that connects all fluxes,

$$\begin{aligned} F_1 &= F_5 + F_4; \\ F_2 &= F_7 + F_6; \\ F_3 &= \frac{1}{3}(F_4 + F_5 + F_6 + F_7). \end{aligned}$$

As shown in Figure (B.2) we can model fluxes $F_1(t)$ and $F_2(t)$ as a linear sum of three stationary Gaussian processes $V_i, U_i, Z_i,\ i = 1, 2$ and additive measurement noise. Each stationary process is obtained from a convolution between Gaussian white noise process and Gaussian kernel. Flux $F_3(t)$ is one third of the linearly combined fluxes $F_1(t)$ and $F_2(t)$, this relationship can be obtained from the stoichiometric matrix. Thus, we obtain the expressions $F_i(t) = V_i(t) + U_i(t) + Z_i(t) + W_i(t)$, $i = 1, 2$, where $W_i(t)$ is a stationary Gaussian white noise process with variance, σ_i^2; and flux three is $F_3(t) = \frac{1}{3}(F_1(t) + F_2(t))$. Intermediate processes in the MGP model are defined as

$$\begin{aligned} V_1(t) &= (h_4 + h_5) \otimes X_1; & V_2(t) &= (h_6 + h_7) \otimes X_2; \\ U_1(t) &= k_4 \otimes X_4; & U_2(t) &= k_6 \otimes X_6; \\ Z_1(t) &= k_5 \otimes X_5; & Z_2(t) &= k_7 \otimes X_7, \end{aligned}$$

where h_i and k_i, $(i = 4, 5, 6, 7)$ are parameterised squared exponential kernels,

$$\begin{aligned} k_i(t) &= v_i \exp\left(-\frac{1}{2}t^2 A_i\right), \quad A_i > 0, \quad i = 4, 5, 6, 7; \\ h_i(t) &= w_i \exp\left(-\frac{1}{2}t^2 B_i\right), \quad B_i > 0, \quad i = 4, 5, 6, 7; \end{aligned}$$

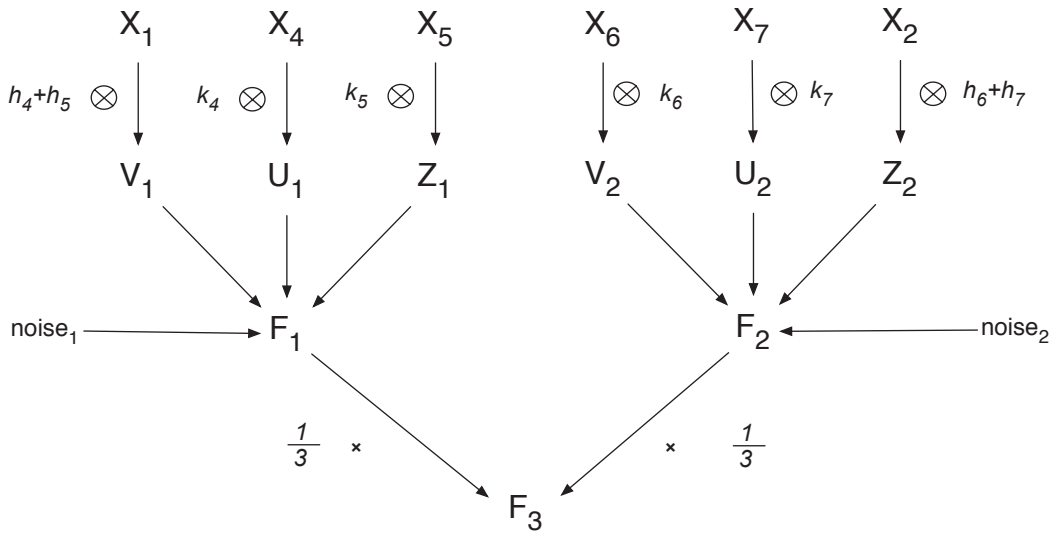


Figure B.2: Multiple-output Gaussian process model for three dependent fluxes F_1, F_2 and F_3. Here, all X_i, $i = 1, 2, 4, 5, 6, 7$ and $\text{noise}_{1,2,3}$ are independent Gaussian white noise processes.

The general expression for the covariance function (see equation (2.16)), that includes auto-covariance ($i = j$) and cross-covariance ($i \neq j$) between outputs i and j and inputs t_a and t_b, was derived in the work of Boyle and Frean (2005) by computing convolution integral. For the metabolic flux model auto and cross covariance expressions are listed below; the first set describes the auto-covariance, the second set describes the cross-covariances,

$$C^F_{11}(d) = C^{V_1}_{11}(d) + C^{U_1}_{11}(d) + C^{Z_1}_{11}(d) + \delta_{ab}\sigma_1^2; \qquad C^F_{12}(d) = C^F_{21}(d) = 0;$$

$$C^F_{22}(d) = C^{V_2}_{22}(d) + C^{U_2}_{22}(d) + C^{Z_2}_{22}(d) + \delta_{ab}\sigma_2^2; \qquad C^F_{13}(d) = \frac{1}{3}C^F_{11}(d), \quad C^F_{31}(d) = C^F_{13}(-d);$$

$$C^F_{33}(d) = \frac{1}{3}\left(C^{F_1}_{11}(d) + C^{F_2}_{22}(d)\right); \qquad C^F_{23}(d) = \frac{1}{3}C^F_{22}(d), \quad C^F_{32}(d) = C^F_{23}(-d),$$

where $d = t_a - t_b$ is a separation between inputs t_a and t_b, and

$$\begin{aligned} C_{ii}^{V_i}(d) &= w_{j_1}^2 \frac{\pi^{\frac{1}{2}}}{\sqrt{B_{j_1}}} \exp\left(-\frac{B_{j_1} d}{4}\right) + w_{j_2}^2 \frac{\pi^{\frac{1}{2}}}{\sqrt{B_{j_2}}} \exp\left(-\frac{B_{j_2} d}{4}\right); \\ C_{ii}^{U_i}(d) &= v_{j_1}^2 \frac{\pi^{\frac{1}{2}}}{\sqrt{A_{j_1}}} \exp\left(-\frac{A_{j_1} d}{4}\right); \\ C_{ii}^{Z_i}(d) &= v_{j_2}^2 \frac{\pi^{\frac{1}{2}}}{\sqrt{A_{j_2}}} \exp\left(-\frac{A_{j_2} d}{4}\right), \end{aligned}$$

(for $i = 1, j_1 = 4, j_2 = 5$ and $i = 2, j_1 = 6, j_2 = 7$). From given covariance functions $C_{ij}^F(d)$ we can obtain the covariance matrices $\mathbf{C}_{ij}$, that all together lead to a positive definite and symmetric covariance matrix $\mathbf{C}$ (see equation (2.17)). For the considered metabolic model we can set $A_i = \exp(f_i)$, $B_i = \exp(g_i)$ and $\sigma_i = \exp(\beta_i)$. Next, setting priors on hyper-parameters, $v_i, w_i, f_i, g_i, \beta_i$, we can compute MAP estimates using log-likelihood (2.13), where $n = 3N$, vector $\mathbf{F} = [F_1\ F_2\ F_3]^T$ contains measurements of fluxes, and $\mathbf{C}$ is the overall covariance matrix. To minimise the negative log-likelihood (2.13) multiplied by the priors we can employ a nonlinear Nelder–Mead optimisation algorithm (Nelder and Mead, 1965) (downhill simplex method).

Above we discussed how measured fluxes, F_1, F_2, F_3, can be modelled in terms of kernels of unknown fluxes, h_4, h_5, h_6, h_7. By doing so, this enabled us to learn all the hyper-parameters that were introduced in the MGPs model. For this reason, such modelling could provide a way to make predictions about all fluxes (measured and unknown) jointly because internal and external fluxes are derived from the same noise sources. This means, that the future work could focus on exploring further the design of MGPs model as well as identify potential types of metabolic pathways where the flux uniqueness question could be addressed.

Bibliography

Aitken, S. and Akman, O. E. (2013). Nested sampling for parameter inference in systems biology: application to an exemplar circadian model. *BMC systems biology*, 7(1):72.

Aldous, D. (1985). *Exchangeability and Related Topics in Lecture Notes in Mathematics*. Springer-Verlag, Berlin.

Aldridge, B. B., Burke, J. M., Sorger, P. K., and Lauffenburger, D. A. (2006). Physico-chemical modelling of cell signalling pathways. *Nature cell biology*, 8.

Aldridge, B. B., Saez-Rodriguez, J., Muhlich, J. L., Sorger, P. K., and Lauffenburger, D. A. (2009). Fuzzy logic analysis of kinase pathway crosstalk in tnf/egf/insulin-induced signalling. *PLOS Computational Biology*, 5(4).

Alon, U. (2007). Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461.

Altman, R. (2013). Personal genomic measurements: The opportunity for information integration. *Clinical Pharmacology & Therapeutics*, 93(1):21–23.

Alvarez, M. A. and Lawrence, N. D. (2009). Sparse convolved multiple output Gaussian processes. *arXiv preprint arXiv:0911.5107*.

Alvarez, M. A., Luengo, D., Titsias, M. K., and Lawrence, N. D. (2010). Efficient multioutput Gaussian processes through variational inducing kernels. *Appearing in*

Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS).

Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J. A., and Blom, J. G. (2009). Systems biology: parameter estimation for biochemical models. *FEBS journal*, 276(4):886–902.

Barndorff-Nielsen, O. E., Cox, D. R., and Kl uppelberg, C. (2000). *Complex Stochastic Systems*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability.

Behrens, A., van Deursen, J. M., Rudolph, K. L., and Schumacher, B. (2014). Impact of genomic damage and ageing on stem cell function. *Nature Cell Biology*, 16(3):201–207.

Bilmes, J. A. et al. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126.

Birke, M., Bissantz, N., and Holzmann, H. (2010). Confidence bands for inverse regression models. *Inverse Problems*, 26(11):115020.

Birtwistle, M., Mager, D., and Gallo, J. (2013). Mechanistic vs. empirical network models of drug action. *Pharmacometrics & Systems Pharmacology*, 2:e72.

Bishop, C. M. (2006). *Pattern recognition and machine learning*, volume 1. springer New York.

Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via pólya urn scheme. *The Annals of Statistics*, 1:353–355.

Blank, L. M. and Ebert, B. E. (2012). From measurement to implementation of metabolic fluxes. *Current Opinion in Biotechnology*, 24:13–21.

Boyle, P. (2007). *Gaussian Processes for Regression and Optimisation*. PhD thesis, Victoria University of Wellington, New Zealand.

Boyle, P. and Frean, M. (2004). Multiple-output Gaussian process regression. *Technical report, Victoria University of Wellington*.

Boyle, P. and Frean, M. (2005). Dependent gaussian processes. *Advances in Neural Information Processing Systems*, 17:217–224.

Bryant, W. A., Faruqi, A. A., and Pinney, J. W. (2013). Analysis of metabolic evolution in bacteria using whole-genome metabolic models. *J.Comp.Biol.*, 20:755–764.

Cairns, R. A., Harris, I., and Mak, T. W. (2011). Regulation of cancer cell metabolism. *Nature Reviews Cancer.*, 11:85–95.

Cancer Genome Atlas Network and others (2012). Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70.

Celton, M., Malpertuy, A., Lelandais, G., and De Brevern, A. G. (2010). Comparative analysis of missing value imputation methods to improve clustering and interpretation of microarray experiments. *BMC genomics*, 11(1):15.

Chen, R., Mias, G. I., Li-Pook-Than, J., Jiang, L., Lam, H. Y. K., Chen, R., Miriami, E., Karczewski, K. J., Hariharan, M., Dewey, F. E., et al. (2012). Personal omics profiling reveals dynamic molecular and medical phenotypes. *Cell*, 148(6):1293–1307.

Chen, R. and Snyder, M. (2012). Systems biology:personalised medicine for the future? *Current Opinion in Pharmacology*, 12:623–628.

Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D., and Davis, R. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, 2:65–70.

Covert, M. W., Xiao, N., Chen, T. J., and Karr, J. R. (2008). Integrating metabolic, transcriptional regulatory and signal transduction models in escherichia coli. *Bioinformatics*, 24(18):2044–2050.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain monte carlo convergence diagnostics: A comparative review. *Journal of the Americal Statistical Association*, 91:883–904.

Cressie, N. (1993). *Statistics for Spatial Data: Wiley Series in Probability and Statistics*. Wiley-Interscience New York.

Dalakas, M. C. (2006). Sporadic inclusion body myositis – diagnosis, pathogenesis and therapeutic strategies. *Nature Clinical Practice Neurology*, 2(8):437–447.

Dawood, S., Hu, R., Homes, M. D., Collins, L. C., Schnitt, S. J., Connolly, J., Colditz, G. A., and Tamimi, R. M. (2011). Defining breast cancer prognosis based on molecular phenotypes: results from a large cohort study. *Breast cancer research and treatment*, 126(1):185–192.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *Journal of the Royal Statistical Society:Series B (Statistical Methodology)*, 68:411–432.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *J. of the Royal Statistical Society, Series B*, 34:1–38.

Efron, B. (1981). Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. *Biometrica*, 68:589–599.

Escobar, M. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230.

Filippi, S., Barnes, C. P., Cornebise, J., and Stumpf, M. (2013). On optimality of kernels for approximate bayesian computation using sequential monte carlo. *Statistical applications in genetics and molecular biology*, 12(1):87–107.

Fill, J. A., Flajolet, P., and Kapur, N. (2005). Singularity analysis, hadamard products, and tree recurrences. *Journal of Computational and Applied Mathematics*, 174(2):271–313.

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

Friedman, N. and Nachman, I. (2000). Gaussian process networks. *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 211–219.

Fritsch, A. and Ickstadt, K. (2009). Improved criteria for clustering based on the posterior similarly matrix. *Bayesian Anal.*, 4:367–391.

Gamerman, D. and Lopes, H. F. (2006). *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press.

Gao, P., Honkela, A., Rattray, M., and Lawrence, N. D. (2008). Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, 24(16):i70–5.

Garcia-Etxebarria, K., Garcia-GarcerĹ, M., and Calafell, F. (2014). Consistency of metagenomic assignment programs in simulated and real data. *BMC Bioinformatics*, 15(90):1–10.

Gehlenborg, N., O'Donoghue, S. I., Baliga, N. S., Goesmann, A., Hibbs, M. A., Kitano, H., Kohlbacher, O., Neuweger, H., Schneider, R., Tenenbaum, D., and Gavin, A. C. (2010). Visualization of omits data for systems biology. *Nature Methods Supplement*, 7(3s):S56–S68.

Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2003). *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd edition.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–511.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions an the bayesian restoration of images. *IEEE Trans. Pattn. Anal. Mach. Intel.*, 6:721–741.

Geva-Zatorsky, N., Rosenfeld, N., Itzkovitz, S., Milo, R., Sigal, A., Dekel, E., Yarnitzky, T., Liron, Y., Polak, P., Lahav, G., and Alon, U. (2009). Oscillations and variability in the p53 system. *Molecular Systems Biology*, 13.

Geweke, J. (1992). *Evaluating the Accuracy of Sampling-Based Approaches to Calculating Posterior Moments, in J. M. Bernardo, J. O. Berger, A. P. Dawiv, and A. F. M. Smith, Bayesian Statistics*, volume 4. Oxford, Clarendon Press.

Geyer, C. (2011). Introduction to markov chain monte carlo. *Handbook of Markov Chain Monte Carlo*, pages 3–48.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London.

Girard, A. (2004). *Approximate methods for propagation of uncertainty with Gaussian process models*. PhD thesis, University of Glasgow, UK.

Goel, G., Chou, I., and Voit, E. O. (2008). System estimation from metabolic time-series data. *Bioinfromatics*, 24:2505–2511.

Granovskaia, M., Jensen, L., Ritchie, M., Toedling, J., Ning, Y., Bork, P., Huber, W., and Steinmetz, L. (2010). High-resolution transcription atlas of the mitotic cell cycle in budding east. *Genome biology*, 11:R24.

Grau, J. M. and Selva-O'Callaghan, A. (2008). *Sporadic Inclusion Body Myositis. in-Diagnostic Criteria in Autoimmune Diseases, from Y. Shoenfeld et al. (eds.)*. Humana Press, NJ.

Green, P. J. and Richardson, S. (2001). Modelling heterogeneity with and without the dirichlet process. *Scandinavian journal of statistics*, 28(2):355–375.

Harbison, C. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431:99–104.

Hastie, T., Tibshirani, R., and Friedman, J. (2003). *The elements of statistical learning*. Springer.

Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P., and Botstein, D. (1999). Imputing missing data for gene expression arrays.

Haykin, S. and Moher, M. (2010). *Communication Systems, 5th ed.* Wiley, Asia.

Hirata, H., Yoshiura, S., Ohtsuka, T., Bessho, Y., Harada, T., Yoshikawa, K., and Kageyama, R. (2002). Oscillatory expression of the bhlh factor hes1 regulated by a negative feedback loop. *Science STKE*, 298:840–843.

Hua, F., Hautaniemi, S., Yokoo, R., and Lauffenburger, D. A. (2006). Integrated mechanistic and data-driven modelling for multivariate analysis of signalling pathways. *Journal of the Royal Society Interface*, 3(9):515–526.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *J. Classif.*, 2:193–218.

Ingram, P., Stumpf, M., and Stark, J. (2008). Nonidentifiability of the source of intrinsinc noise in gene expression from single-burst data. *PLoS Computational Biology*, 4:1–10.

Janes, K. A. and Yaffe, M. B. (2006). Data-driven modelling of signal-transduction networks. *Nature Reviews Molecular Cell Biology*, 7:820–828.

Joyce, A. and Palsson, B. O. (2006). The model organism as a system: integrating 'omics' data sets. *Nature Reviews Molecular Cell Biology*, 7:198–210.

Kageyama, R., Ohtsuka, T., and Kobayashi, T. (2007). The hes gene family: repressors and oscillators that orchestrate embryogenesis. *Development (primer)*, 134(7):1243–1251.

Karr, J., Sanghvi, J., Macklin, D., Gutschow, M., Jacobs, J., Bolival, B. J., Assad-Garcia, N., Glass, J., and Covert, M. (2012). A whole-cell computational model predicts phenotype from genotype. *Cell*, 12:180.

Kerr, M. K. and Churchill, G. A. (2001). Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. *Proceedings of The National Academy of Science of the United States of America*, 98(16):8961–8965.

Kholodenko, B. (2006). Cell-signalling dynamics in time and space. *Nature Reviews Molecular Cell Biology*, 7(3):165–176.

Kim, K. I. and Simon, R. (2014). Using single cll sequencing data to model the evolutionary history of a timor. *BMC Bioinformatics*, 15(27):1–13.

Kirk, P., Griffin, J., Savage, R., Ghahramani, Z., and Wild, D. (2012). Bayesian correlated clustering to integrate multiple datasets. *Bioinformatics*, 28:3290–3297.

Kirk, P. and Stumpf, M. (2009). Gaussian process regression bootstrapping: Exploring the effects of uncertainty in time course data. *Bioinformatics*, 25:1300–1306.

Kirk, P. W. D. (2011). *Inferential stability in systems biology*. PhD thesis, Imperial College London, UK.

Klamt, S. and Stelling, J. (2003). Two approaches for metabolic pathway analysis? *Trends in Biotechnoligy*, 21(2):64–69.

Klipp, E., Herwig, R., Kowald, A., Wierling, C., and Lehrach, H. (2005). *Systems Biology in Practice: Concepts, Implementation and Application*. Wiley-VCH Verlag GmbH & Co. KGaA, Berlin.

Kotte, O., Zaugg, J. B., and Heinemann, M. (2010). Bacterial adaptation through distributed sensing of metabolic fluxes. *Molecular Systems Biology*, 6.

Kruse, K. and Jülicher, F. (2005). Oscillations in cell biology. *Current Opinion in Cell Biology*, 17:20–26.

Landau, H. J. (1967). Sampling, data transmission, and the nyquist rate. In *Proceedings of the IEEE,* pages 1701–1706.

Lawrence, N. D., Sanguinetti, G., and Rattray, M. (2006). Modelling transcriptional regulation using Gassign processes. *Advances in Neural Information and Processing Systems NIPS*.

Levine, A. J. (1997). p53, the cellular gatekeeper for growth and division. *Cell*, 88:323–331.

Lock, E. and Dunson, D. (2013). Bayesian consensus clustering. *Bioinformatics*, 29(20):2610–2616.

Lord, C. J. and Ashworth, A. (2012). The DNA damage response and cancer therapy. *Nature*, 481:287–294.

Luan, Y. and Li, H. (2003). Clustering of time-course gene expression data using a mixed-effects model with b-splines. *Bioinformatics*, 19(4):474–482.

Machado, P., Miller, A., Holton, J., and Hanna, M. (2009). Sporadic inclusion body myositis: an unsolved mystery. *Acta Reumatol Port.*, 34(2A):161–182.

MacKay, D. J. (1997). Gaussian processes-a replacement for supervised neural networks? *In NIPS97 Tutorial*.

MacKay, D. J. C. (1998). Introduction to Gaussian processes. *Neural Networks and Machine Learning*.

Marin, J., Mengersen, K., and Robert, C. P. (2005). *Bayesian modelling and inference on mixtures of distributions*, volume 25. Elsevier.

Maroto, M. and Monk, N. (2008). *Cellular Oscillatory Mechanisms*. Springer, Landes Bioscience and Springer Science+Business Media.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Hohn Wiley Sons, INC., New York.

Medvedovic, M., Yeung, K. Y., and Bumgarner, R. E. (2004). Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8):1222–1232.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52:91–118.

Morris, M. K., Saez-Rodriguez, J., Sorger, P. K., and Lauffenburger, D. A. (2010). Logic-based models for the analysis of cell signaling networks. *Biochemistry*, 49(15):3216–3224.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Neal, R. M. (1997). Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*.

Neal, R. M. (2000). Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265.

Needham, M. and Mastaglia, F. L. (2007). Inclusion body myositis: current pathogenetic concepts and diagnostic and therapeutic approaches. *The Lancet Neurology*, 6(7):620–631.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Computer journal*, 7:308–313.

Nielsen, J. (2003). It is all about metabolic fluxes. *Journal of Bacteriology*, 185(24):7031–7035.

Noack, S., Nöh, K., Moch, M., Oldiges, M., and Wiechert, W. (2011). Stationary versus non-stationary (13)c-mfa: a comparison using a consistent dataset. *Journal of Biotechnology*, 154(2-3):179–190.

Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., and Ishii, S. (2003). A bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096.

O'Hagan, A. and Kingman, J. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–42.

Opper, M. and Winther, O. (2000). Gaussian processes for classification: mean-field algorithms. *Neural computation*, 12(11):2655–2684.

Orbanz, P. and Teh, Y. W. (2010). *Bayesian nonparametric models*. Springer.

Orth, J. D., Thiele, I., and Palsson, B. O. (2010). What is flux balance analysis? *Nature Biotechnology*, 28(3):245–248.

Pan, W., Lin, J., Le, C. T., et al. (2002). Model-based cluster analysis of microarray gene-expression data. *Genome Biology*, 3(2):1–0009.

Price, N., Schellenberger, J., and Palsson, B. O. (2004). Uniform sampling of steady-state flux spaces: Means to design experiments and to interpret enzymopathies. *Biophysical Journal*, 87(4):2172–2186.

Ptashne, M. (1985). Gene regulation by proteins acting nearby and at a distance. *Nature*, 322(6081):697–701.

Purvis, J., Karhohs, W., Mock, C., Batchelor, E., Loewer, A., and Lahav, G. (2012). Science. *p53 Dynamics Control Cell Fate*, 336:1440–1444.

Quo, C. F., Kaddi, C., Phan, J. H., Zollanvari, A., Xu, M., Wang, M. D., and Alterovitz, G. (2012). Reverse engineering biomolecular systems using- omic data: challenges, progress and opportunities. *Briefings in bioinformatics*, 13(4):430–445.

Rakha, E. A., Reis-Filho, J. S., and Ellis, I. O. (2008). Basal-like breast cancer: a critical review. *Journal of Clinical Oncology*, 26(15):2568–2581.

Rasmussen, C. E. (2000). The infinite gaussian mixture model. *Advances in neural information processing systems*, 12:554–560.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press, USA.

Ridley, K. D. and Jakeman, E. (1999). Fm demodulation in the presence of multiplicative and additive noise. *Inverse Problems*, 15(4):989–1002.

Roberts, M. J. (2008). *Fundamentals of Signals and Systems*. Mc Graw Hill, New York.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., and Aigrain, S. (2013). Gaussian processes for time-series modelling. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 371(1984):20110550.

Ross, W. P. and Kaina, B. (2013). Dna damage-induced cell death: From specific dna lesions to the dna damage response and apoptosis. *Cancer Letters*, 332:237–248.

Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons.

Savage, R., Ghahramani, Z., Griffin, J., Kirk, P., and Wild, D. (2013). Identifying cancer subtypes in glioblastoma by combining genomic, transcriptomic and epigenomic data. *International Conference on Machine Learning (ICML):Workshop on Machine Learning in Genetics and Genomics*, arXiv:1304.3577 [q-bio.GN].

Schumacher, J., Behrends, V., Pan, Z., Brown, D. R., Heydenreich, F., Lewis, M. R., Bennett, M. H., Razzaghi, B., Komorowski, M., Barahona, M., et al. (2013). Nitrogen and carbon status are integrated at the transcriptional level by the nitrogen regulator ntrc in vivo. *MBio*, 4(6):e00881–13.

Schwartz, J.-M. and Kanehisa, M. (2006). Quantitative elementary mode analysis of metabolic pathways: the example of yeast glycolysis. *BMC Bioinformatics*, 7(186).

Seeger, M. (2004). Gaussian processes for machine learning. *International journal of neural systems*, 14(2):69–106.

Sethuraman, J. (1994). A constructive definition of dirichlet priors. *Statistica Sinica*, 4:639–650.

Shen, R., Olshen, A., and Ladanyi, M. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, 25:2906–2912.

Silk, D., Kirk, P., Barnes, C. P., Toni, T., Rose, A., Moon, S., Dallman, M. J., and Stumpf, M. (2011). Designing attractive models via automated identification of chaotic and oscillatory dynamical regimes. *Nature communications*, 2:489.

Silvey, S. (1975). *Statistical Inference*. Chapman and Hall.

Solak, E., Murray-Smith, R., Leithead, W., Leith, D., and Rasmussen, C. (2003). Derivative observations in gaussian process models of dynamic systems. *In Advances in Neural Information Processing Systems*, 15:1033–1040.

Spencer, S. L. and Sorger, P. (2011). Measuring and modelling apoptosis in single cells. *Cell*, 144:926–939.

Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006). Biogrid: a general repository for interaction datasets. *Nucleic Acids Research*, 34:D535–D539.

Sun, T., Yang, W., Liu, J., and Shen, P. (2011). Modeling the basal dynamics of p53 system. *PLoS ONE*, 6(11):1–12.

Teh, Y. W. (2010). *Dirichlet process*. Springer.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476).

Thorne, T., Fratta, P., Hanna, M. G., Cortese, A., Plagnol, V., Fisher, E. M., and Stumpf, M. P. (2013). Graphical modelling of molecular networks underlying sporadic inclusion body myositis. *Molecular BioSystems*, 9(7):1736–1742.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525.

van Heeswijk, W. C., Westerhoff, H. V., and F.C., B. (2013). Nitrogen assimilation in escherichia coli: putting molecular data into a systems perspective. *Reliability Engineering System Safety*, 77(4):628–695.

Varma, A. and Palsson, B. O. (1994). Metabolic flux balancing: Basic concepts, scientific and practical use. *Bio/technology*, 12.

Villaverde, A. F. and Banga, J. R. (2014). Reverse engineering and identification in systems biology: strategies, perspectives and challenges. *Journal of The Royal Society Interface*, 11(91):20130505.

Voit, E. O. (2013). Characterizability of metabolic pathway systems from time series data. *Mathematical Biosciences*, 5:1–11.

Voit, E. O. and Almeida, J. (2004). Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*, 20(11):1670–1681.

Žurauskienė, J., Kirk, P., Thorne, T., Pinney, J., and Stumpf, M. (2014a). Derivative processes for modelling metabolic fluxes. *Bioinformatics*, page btu069.

Žurauskienė, J., Kirk, P., Thorne, T., and Stumpf, M. (2014b). Bayesian non-parametric approaches to reconstructing oscillatory systems and the nyquist limit. *Physica A: Statistical Mechanics and its Applications*, 407:33–42.

Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., Haibe-Kains, B., and Goldenberg, A. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nature methods*, 11(3):333–337.

Wang, L., Liu, J., and Li, S. Z. (2000). Mrf parameter estimation by mcmc method. *Pattern recognition*, 33(11):1919–1925.

Weerasinghe, S. (2010). A missing values imputation method for time series data: an efficient method to investigate the health effects of sulphur dioxide levels. *Environmetrics*, 21(2):162–172.

Wilkinson, D. J. (2006). *Stochastic Modelling for Systems Biology*. Taylor and Francis, London.

Yau, C., Papaspoliopoulos, O., Roberts, G., and Holmes, C. (2011). Bayesian non-parametric hidden markov models with application to the analysis of copy-number-variation in mammalian genomes. *Stat Soc Series B Stat Methodol*, 73(1):37–57.

Zamboni, N. (2011). ^{13}c metabolic flux analysis in complex systems. *Current Opinion in Biotechnology*, 22:103–108.

Zhang, B. and Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1).

Zhang, S., Li, Q., Liu, J., and Zhou, X. (2011). A novel computational framework for simultaneous integration of multiple types of genomic data to identify microrna-gene regulatory modules. *Bioinformatics*, 27:i401–i409.