Computational Methods for Investigating Dendritic Cell Biology

by

Ana Paula de Oliveira Sales

Computational Biology & Bioinformatics Program
Duke University

Date: ______________________

Approved:

__________________________
Thomas B. Kepler, Advisor

__________________________
Cliburn Chan

__________________________
Georgia D. Tomaras

__________________________
Mike West

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Computational Biology & Bioinformatics Program in the Graduate School of Duke University 2011
Abstract

(Computational Biology & Bioinformatics)

Computational Methods for Investigating Dendritic Cell Biology

by

Ana Paula de Oliveira Sales

Computational Biology & Bioinformatics Program
Duke University

Date: __________________________

Approved:

______________________________
Thomas B. Kepler, Advisor

______________________________
Cliburn Chan

______________________________
Georgia D. Tomaras

______________________________
Mike West

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Computational Biology & Bioinformatics Program in the Graduate School of Duke University 2011
Abstract

The immune system is constantly faced with the daunting task of protecting the host from a large number of ever-evolving pathogens. In vertebrates, the immune response results from the interplay of two cellular systems: the innate immunity and the adaptive immunity. In the past decades, dendritic cells have emerged as major players in the modulation of the immune response, being one of the primary links between these two branches of the immune system.

Dendritic cells are pathogen-sensing cells that alert the rest of the immune system of the presence of infection. The signals sent by dendritic cells result in the recruitment of the appropriate cell types and molecules required for effectively clearing the infection. A question of utmost importance in our understanding of the immune response and our ability to manipulate it in the development of vaccines and therapies is: “How do dendritic cells translate the various cues they perceive from the environment into different signals that specifically activate the appropriate parts of the immune system that result in an immune response streamlined to clear the given pathogen?”

Here we have developed computational and statistical methods aimed to address specific aspects of this question. In particular, understanding how dendritic cells ultimately modulate the immune response requires an understanding of the subtleties of their maturation process in response to different environmental signals. Hence, the first part of this dissertation focuses on elucidating the changes in the transcriptional
program of dendritic cells in response to the detection of two common pathogen-
associated molecules, LPS and CpG. We have developed a method based on Langevin
and Dirichlet processes to model and cluster gene expression temporal data, and
have used it to identify, on a large scale, genes that present unique and common
transcriptional behaviors in response to these two stimuli. Additionally, we have
also investigated a different, but related, aspect of dendritic cell modulation of the
adaptive immune response. In the second part of this dissertation, we present a
method to predict peptides that will bind to MHC molecules, a requirement for the
activation of pathogen-specific T cells. Together, these studies contribute to the
elucidation of important aspects of dendritic cell biology.
# Contents

Abstract iv  
List of Tables xi  
List of Figures xiii  
List of Abbreviations xvi  
Acknowledgements xviii  

1 Introduction 1  
1.1 Dissertation structure ........................................ 7  
1.2 Our contribution .............................................. 8  
1.2.1 Methodological context and accomplishments ............. 8  
1.2.2 Biological context and accomplishments .................... 12  

2 A primer on dendritic cell activation of the immune response 16  
2.1 The immune response ........................................... 16  
2.2 Dendritic cells bridge the innate and adaptive immune responses . . 19  
2.3 Costimulatory and polarizing signals ............................ 22  
2.3.1 Toll-like receptors .......................................... 24  
2.3.2 TLR signaling ................................................ 27  
2.3.3 Activation of DCs by LPS via TLR4 ......................... 30  
2.3.4 Activation of DCs by CpG via TLR9 ........................ 31  
2.4 Antigen-specific signal ......................................... 34
2.4.1 Antigen presentation by MHC class I ........................................... 35
2.4.2 Antigen presentation by MHC class II ........................................... 38

3 A primer on relevant statistical frameworks ....................................... 41
  3.1 Nonparametric Bayesian models ..................................................... 42
    3.1.1 Some examples of Bayesian nonparametric models ...................... 43
  3.2 Dirichlet process models ............................................................. 44
    3.2.1 The Dirichlet distribution and its aggregation property ............... 44
    3.2.2 The Dirichlet process as a generalization of the Dirichlet dis-
        tribution .............................................................................. 47
    3.2.3 Construction via stick-breaking ................................................. 48
    3.2.4 Modified Pólya-urn scheme ...................................................... 49
    3.2.5 The Chinese restaurant process ................................................. 50
    3.2.6 DP mixture models ................................................................. 51
  3.3 Gaussian processes ........................................................................ 53
    3.3.1 Gaussian processes applications and relation to other models ........ 56
    3.3.2 Covariance functions ............................................................... 57

4 Comparing individual gene trajectories .............................................. 60
  4.1 Introduction .................................................................................... 60
  4.2 The dendritic cell temporal gene expression dataset .......................... 63
  4.3 The Langevin process ..................................................................... 64
    4.3.1 Brownian motion and the Langevin equations ............................. 64
    4.3.2 Basic Langevin process ............................................................ 65
  4.4 Modeling temporal gene expression trajectories with the Langevin pro-
    cess ............................................................................................. 66
    4.4.1 Interpretation of gene expression through the Langevin process ..... 66

vii
4.4.2 Notation ................................................. 67
4.4.3 Statistical model ................................. 69
4.4.4 Parameters priors .................................. 70
4.4.5 Additive Langevin processes ................. 72
4.5 Test of flatness ..................................... 75
  4.5.1 Model selection .................................. 76
  4.5.2 Tests of the baseline effect .................... 77
  4.5.3 Tests of the treatment effect ................. 78
  4.5.4 Estimation of the treatment effect .......... 86
4.6 Test of flatness — results for the DC dataset .... 87
  4.6.1 Dynamics of gene expression .................. 88
  4.6.2 Cytokines ....................................... 91
  4.6.3 Transcription factors .......................... 94
4.7 Comparison of treatment effects across experiments .... 96
  4.7.1 Four model selection ............................ 98
4.8 Comparison of treatment effects across experiments — results for the DC dataset .... 101
  4.8.1 Cytokines ....................................... 106
  4.8.2 Transcription factors .......................... 107
4.9 Discussion .......................................... 109

5 Clustering gene trajectories .......................... 116
  5.1 Statistical Model .................................. 117
  5.2 Posterior simulation ............................... 122
  5.3 Post-processing of MCMC sampler output ....... 124
    5.3.1 Practical issues with the Gibbs sampler .... 126
    5.3.2 Post-clustering merge algorithm ............ 128
B.2 Cytokines ....................................................... 168

C Flatness tests additional results 169
   C.1 Treatment effect parameters ................................. 169
   C.2 Cytokines ..................................................... 170
   C.3 Transcription factors ........................................ 175

D Clustering additional results 180

E Prediction of peptide-MHCI binding additional results 182

Bibliography 186

Biography 198
List of Tables

2.1 Comparison of key aspects of the innate and adaptive immune responses.  19
4.1 Mapping of time index to hours. ........................................  64
4.2 Summary of tests for control effects. ...................................  78
4.3 Inferred hyperparameters for the MAP trajectories of IL33 under LPS stimulation. ........................................  80
4.4 Inferred hyperparameters for the MAP trajectories of IL15 under LPS stimulation. ........................................  81
4.5 Inferred hyperparameters for the MAP trajectories shown in Figure 4.5.  85
4.6 Summary of tests for treatment effects. ...............................  87
4.7 Summary of the number of differentially expressed cytokines under one, both, or neither of the two treatments. .............  91
4.8 Summary of the number of differentially expressed transcription factors under one, both, or neither of the two treatments. ..........  94
4.9 Summary of results of the four model selection on the DC dataset. . 102
4.10 Summary of results of the four model selection on the DC dataset — cytokines. ....................................................... 106
4.11 Cytokines with same treatment effect in response to LPS and CpG stimulation. ....................................................... 107
4.12 Cytokines with nonzero but different treatment effects in response to LPS and CpG stimulation. ................................... 108
4.13 Summary of results of the four model selection on the DC dataset — transcription factors. ........................................... 108
5.1 Summary of the nine-cluster configuration of the transcription factors differentially expressed in response to CpG.

6.1 Number of binders and non-binders in training sets.

6.2 Number of binders and non-binders in Peters et al. (2006) datasets for 5 alleles.

6.3 Comparison of classifiers performance as measured by AUC for 5 alleles.

C.1 Summary of results of treatment flatness tests for cytokines classified as not flat for both LPS and CpG treatments.

C.2 Summary of results of treatment flatness tests for cytokines classified as not flat for LPS treatment and as flat for CpG treatment.

C.3 Summary of results of treatment flatness tests for cytokines classified as not flat for CpG treatment and as flat for LPS treatment.

C.4 Summary of results of treatment flatness tests for transcription factors classified as not flat for both LPS and CpG treatments.

C.5 Summary of results of treatment flatness tests for transcription factors classified as not flat for LPS and as flat for CpG treatments.

C.6 Summary of results of treatment flatness tests for transcription factors classified as flat for LPS treatment and as not flat for CpG treatment. For brevity, here we only show 50 out of the 367 TFs not flat in CpG. These are the 50 TFs with highest values of $|DL_{treat}^L - DL_{treat}^C|$.

E.1 AUC values for different classifiers. The second and third columns correspond to the decision trees described in the present work. Note the improvement in the performance of the trees with the use of training costs. *Values extracted from table 2 of Peters et al. (2006).
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Correlation between time points induced by the Langevin process.</td>
<td>68</td>
</tr>
<tr>
<td>4.2</td>
<td>Langevin parameters prior distributions.</td>
<td>71</td>
</tr>
<tr>
<td>4.3</td>
<td>Example of null treatment effect: Il33 under LPS stimulation.</td>
<td>81</td>
</tr>
<tr>
<td>4.4</td>
<td>Example of non-zero treatment effect: IL15 under LPS stimulation.</td>
<td>82</td>
</tr>
<tr>
<td>4.5</td>
<td>Examples of the variety of expression patterns modeled by the Langevin</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>process.</td>
<td></td>
</tr>
<tr>
<td>4.6</td>
<td>Treatment effect corresponding to the trajectories shown in Figure 4.5.</td>
<td>84</td>
</tr>
<tr>
<td>4.7</td>
<td>Comparison of half-times of genes differentially expressed in response</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>to LPS and CpG stimulation.</td>
<td></td>
</tr>
<tr>
<td>4.8</td>
<td>Comparison of half-times of genes that are up- and down-regulated.</td>
<td>89</td>
</tr>
<tr>
<td>4.9</td>
<td>Histograms of number of inflection points for genes differentially</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>expressed under LPS and CpG stimulation.</td>
<td></td>
</tr>
<tr>
<td>4.10</td>
<td>Treatment effect of cytokines differentially expressed in response to both</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>CpG and LPS stimulation.</td>
<td></td>
</tr>
<tr>
<td>4.11</td>
<td>Comparison of the maximum change in treatment effect between transcription</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>factors not flat in both CpG and LPS.</td>
<td></td>
</tr>
<tr>
<td>4.12</td>
<td>Example of transcription factors determined to be differentially expressed</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>under both treatment by the flatness tests.</td>
<td></td>
</tr>
<tr>
<td>4.13</td>
<td>Transcription factor Tcf2 treatment effect in response to CpG stimulation.</td>
<td>97</td>
</tr>
<tr>
<td>4.14</td>
<td>Illustration of four model selection using Ccl22.</td>
<td>99</td>
</tr>
<tr>
<td>4.15</td>
<td>Example of a gene with the same nonzero treatment effect.</td>
<td>102</td>
</tr>
</tbody>
</table>
4.16 Example of a gene with the nonzero and different treatment effect under LPS and CpG stimulation. ........................................ 103
4.17 Box plot of treatment effect magnitude. ................................. 104
4.18 Treatment effect of genes classified to Model 3 and 4. ............... 105
4.19 Treatment effects of some of the cytokines that respond similarly to LPS and CpG. ........................................ 112
4.20 Treatment effects of some of the cytokines that respond differently to LPS and CpG. ........................................ 113
4.21 Magnitude of the treatment effects of transcription factors in Models 3 and 4. ........................................ 114
4.22 Treatment effect of transcription factors that present different responses to stimulation with LPS and CpG. .................. 114
4.23 Treatment effect of transcription factors that present the same response to stimulation with LPS and CpG. .................. 115

5.1 Artificial Langevin data. ........................................ 129
5.2 Langevin artificial data inferred optimal clusters. ..................... 130
5.3 Comparison of underlying and inferred Langevin data trajectories. . 130
5.4 Pairwise posterior probability matrix for Langevin data. ............ 132
5.5 Sizes of inferred subclusters that map to true Cluster 3. ............ 132
5.6 Clusters obtained for transcription factors stimulated with LPS in the dendritic cell dataset. ................................. 134
5.7 Clusters obtained for transcription factors stimulated with CpG in the dendritic cell dataset. ................................. 137
5.8 Half max of trajectories of clusters obtained by clustering transcription factors stimulated with CpG. ..................... 140

6.1 Theoretical relation between $\lambda_2$ and $E K(\theta)$ ................. 153
6.2 Classifier performance vs class distribution. ......................... 159
6.3 Balancing cost vs minimizing cost .................................. 161
6.4 Comparison of unit cost, balancing cost, under-sampling and oversampling ........................................ 162
6.5 Empirical relation between $\hat{\lambda}_2$ and $EK(\theta)$ ............................. 163

C.1 Histograms of parameters of the treatment effect model for LPS and CpG ............................................. 171

C.2 Histograms of error variance, $\Delta$ DL, and treatment effect magnitude for the treatment effect model for LPS and CpG ........................................................ 173

D.1 Pairwise posterior probability matrix for observations belonging to true Cluster 3 in the Langevin data .................... 180

D.2 Trace of number of clusters over the MCMC chain for clustering transcription factors under LPS stimulation ........................................... 181

D.3 Pairwise posterior probability matrix for transcription factors differentially expressed in response to LPS treatment ...................... 181

E.1 Classifier performance vs class distribution ...................... 184

E.2 Comparison of unit cost, balancing cost, undersampling and oversampling for alleles A0203, A3101 and B1501 ...................... 185
## List of Abbreviations

### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIC</td>
<td>Akaike information criterion.</td>
</tr>
<tr>
<td>ANOVA</td>
<td>Analysis of variance.</td>
</tr>
<tr>
<td>APC</td>
<td>Antigen-presenting cell.</td>
</tr>
<tr>
<td>BIC</td>
<td>Bayesian information criterion.</td>
</tr>
<tr>
<td>CTL</td>
<td>Cytotoxic T lymphocyte.</td>
</tr>
<tr>
<td>DD</td>
<td>Death domain.</td>
</tr>
<tr>
<td>DP</td>
<td>Dirichlet process.</td>
</tr>
<tr>
<td>DL</td>
<td>Description length.</td>
</tr>
<tr>
<td>ER</td>
<td>Endoplasmic reticulum.</td>
</tr>
<tr>
<td>GO</td>
<td>Gene ontology.</td>
</tr>
<tr>
<td>GP</td>
<td>Gaussian process.</td>
</tr>
<tr>
<td>IL</td>
<td>Interleukin.</td>
</tr>
<tr>
<td>LP</td>
<td>Langevin process.</td>
</tr>
<tr>
<td>LBP</td>
<td>Lypopolysaccharide binding protein.</td>
</tr>
<tr>
<td>LPS</td>
<td>Lypopolysaccharide.</td>
</tr>
<tr>
<td>LRR</td>
<td>Leucin-rich repeat.</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a posteriori.</td>
</tr>
<tr>
<td>MDL</td>
<td>Minimum description length.</td>
</tr>
<tr>
<td>MHC</td>
<td>Major histocompatibility complex.</td>
</tr>
<tr>
<td>Abbr.</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>MHCI</td>
<td>Major histocompatibility complex class I.</td>
</tr>
<tr>
<td>MHCII</td>
<td>Major histocompatibility complex class II.</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum likelihood estimator.</td>
</tr>
<tr>
<td>MDL</td>
<td>Minimum description length.</td>
</tr>
<tr>
<td>MyD88</td>
<td>Myeloid differentiation primary response gene 88.</td>
</tr>
<tr>
<td>NN</td>
<td>Neural network.</td>
</tr>
<tr>
<td>PAMP</td>
<td>Pathogen associated molecular pattern.</td>
</tr>
<tr>
<td>PRR</td>
<td>Pattern recognition receptor.</td>
</tr>
<tr>
<td>TIR</td>
<td>Toll-interleukin-1 receptor.</td>
</tr>
<tr>
<td>TLR</td>
<td>Toll like receptor.</td>
</tr>
<tr>
<td>TIRAP</td>
<td>TIR domain-containing adaptor protein.</td>
</tr>
<tr>
<td>TNF</td>
<td>Tumor necrosis factor.</td>
</tr>
<tr>
<td>TRAM</td>
<td>TRIF-related adaptor molecule.</td>
</tr>
<tr>
<td>TRIF</td>
<td>TIR domain-containing adaptor inducing IFN-β.</td>
</tr>
</tbody>
</table>
Acknowledgements

I would first and foremost like to thank my advisor Tom Kepler for his patience, guidance and encouragement. It has been a real honor to be your student and I am deeply grateful for it. I would like to express my gratitude to the members of my committee: Georgia Tomaras, Mike West and Cliburn Chan. Georgia was my mentor in one of my rotations here at Duke, and working with her has been an enriching experience and a true pleasure. The work presented in Chapter 6 is a joint project with her. I would also like to thank Mike for his invaluable statistical advice. My thanks to Cliburn, who has always had an open door when I needed help, and has provided numerous and helpful suggestions, comments and ideas over the years. I would also like to thank Lindsay Cowell, who served in my committee in my early years at Duke. Additionally, I would like to acknowledge my funding source: Multiscale Systems Immunology for Adjuvant Development, HHSN268200500019C, NIH. PI: Kepler.

Many thanks to all current and past members of the Kepler group and the Chan group for all their insightful questions and discussions. In particular, I would like to thank Feng Feng, who has been a close collaborator in the dendritic cell gene expression project. It has been a great pleasure to work with you, Feng! My thanks to Anna Maria, who has never failed to answer even my most obscure immunological questions! Thanks to Jeff Roach for enlightening discussions about my work and helpful suggestions, and for all the help with computer- and code-related issues.
Special thanks to Stanca Ciupe, who helped me many times with mathematical problems, and even more times when I needed my daily dose of afternoon sweets!

My experience here at Duke has been incredibly enriching and fulfilling in many different ways, and a large part of it has been due to the vibrant Duke community. Thus, I owe a huge debt of gratitude to all the Dukies that have contributed to make my experience here more enjoyable, productive and fulfilling. In particular, I would like to thank the CBB members, including fellow students, friends, faculty, and staff members who have contributed to my experience at Duke in one way or another.

I have been blessed with an amazing husband, incredible parents, and an awesome cheerleading team of many friends and family that have been truly supportive of me during this entire journey. Your encouragement and support have helped more than you can imagine. To all of you, my love and my deepest thanks!
Introduction

Most of the Earth’s biomass is comprised of microorganisms, many of which are prone to infecting higher organisms and possibly killing them. As such, the ability to detect and eliminate infection is crucial for survival of these organisms. In fact, virtually every eukaryotic organism has developed some sort of mechanism of defense against invading microbes. The task of defending against microbial infections, however, poses a challenging conundrum to the host. On one hand, the host must be able to identify an ever-increasing number of potentially infectious agents, which not only display an enormous amount of diversity, but also are constantly evolving. On the other hand, the host must be able to discriminate between ‘self’ and ‘non-self’. In other words, the host must be able to detect and fight all possible pathogens it might encounter, while not attacking itself. In the face of such a daunting task, metazoans have developed increasingly sophisticated immune systems to protect themselves against infection, maintaining host-microbe homeostasis, and ultimately guaranteeing the survival of the host.

In vertebrates, protection against infection is ensured by an immune system composed of two different cellular branches: the innate immunity and the adaptive (also
called acquired) immunity. These two systems have evolved distinct strategies for differentiating self from non-self, and in many ways, they are complementary to one another. The cellular pillars of the adaptive immunity, T and B lymphocytes, display receptors that recognize molecular patterns specific to the given invading pathogen. Each lymphocyte expresses only one type of receptor with a single and unique specificity. As a whole, the population of the adaptive immune system cells displays a repertoire of receptors with sufficient diversity to recognize a vast number of potential pathogens, or parts thereof, that the host might encounter. In contrast, each cell of the innate immunity displays a variety of receptors, each of which recognizes evolutionarily conserved molecular patterns common to the microbial world and lacking in the host.

Although the innate recognition of pathogens is substantially simpler than the mechanisms employed by the adaptive immunity, the innate immune response is remarkably effective in protecting the host from infection. This effectiveness, however, is hardly a surprise — innate immunity has been perfected and streamlined over the course of millions of years of evolution, dating back to prior to the divergence of vertebrates from invertebrates. In fact, aspects of the vertebrate innate immunity (specifically, the use of leucin-rich repeats in the recognition of pathogen-associated molecular patterns) have been suggested to having been utilized by the last common ancestor of plants and animals (Pancer and Cooper, 2006). A testament to the effectiveness of the innate immunity is the fact that most eukaryotic multicellular life forms rely solely on it for protection against infection, given that the adaptive branch of the immune response appears to be exclusive to vertebrates.

The innate immunity, however, was not given the attention it deserves until recently. For the best part of the little over a century since immunologists have demonstrated evidence of both the adaptive and innate immunities, the overwhelming majority of studies in the field had been focused on unraveling aspects of the
adaptive immune response. The main reason had been that, during this period of
time, it was understood that the innate immune response recognized pathogens in
a non-specific manner, and reacted to them in a similarly general fashion. As such,
innate immunity was primarily viewed as a rather oversimplistic and uninteresting
immune defense, especially in light of the sophistication of the adaptive immune re-
response. The intricacies and complexity of the various mechanisms utilized by the
lymphocytes of the acquired immunity were much more enticing, and hence, drew
much more scientific interest. Consequently, our understanding of the mechanisms of
operation of the innate immunity has lagged behind that of the adaptive immunity.

It was not until the mid to late 1990s that we gained a fuller appreciation for
the innate immune response, and became aware of it as an evolutionarily conserved,
ancient mechanism of defense. In 1996, Lemaitre et al. (1996) demonstrated that
loss-of-function mutations in a gene known for its involvement in the development of
fruit flies, the Toll receptor, led to high susceptibility to fungal infections. Shortly
thereafter, a human homolog of Toll was identified and implicated in the immune
response to LPS (Medzhitov et al., 1997). Since these seminal studies, the innate
immunity has gained renewed interest and significant strides have been made toward
elucidating what we now know to be an incredibly complex immune response. It
has become clear that the mechanisms involved in innate immunity are nowhere
close to the simplistic mechanisms they had once been assumed to be. Although the
pathogen-recognition receptors are germline encoded and do not undergo the kinds
of refinements observed in lymphocyte receptors, complexity in this system arises
from a number of other factors, such as the diversity of combination of receptors
utilized by the various innate immunity cells, as well as positive and negative feedback
mechanisms and cross-talk among these cells.

As we have have gained further insight into innate immunity, the central role
played by dendritic cells in the overall immune response has become more appar-
ent. Since they were first described by Steinman and Cohn (1973), their role as antigen-presenting cells has been actively investigated. However, as studies in innate immunity advanced and as we gained a better understanding of the variety of innate pathogen-sensing receptors and the complexities of their signaling pathways, our view of dendritic cells has shifted from being merely adjuvants of the adaptive immunity to being major controllers of it. We now know that these cells’ relevance is not limited to either branch of the immune system; rather, in addition to aiding in the first line of response provided by the innate immunity, dendritic cells play a central and crucial role in the development and modulation of the acquired immune response. In fact, their pervasive role in T cell biology starts as early as in the developmental stages of these lymphocytes, as dendritic cells help shape the repertoire of T cell clones in the thymus. Additionally, they affect the T cell response by capturing antigens and presenting them via MHC, and thus selectively activate T-cell clones. Finally, dendritic cells provide signals, via the secretion of cytokines and receptor interactions, that dictate the differentiation path taken by T cells, thereby modulating the cellular immune response.

Due to their central role in the immune response, much of the recent efforts in the development of therapies and vaccines have turned to dendritic cells. The elicitation of an effective immune response with long-lasting memory requires appropriate stimulation of dendritic cells in order for them to produce the necessary signals to correctly modulate the activation of the adaptive immunity. The importance of the co-administration of adjuvant molecules that stimulate dendritic cells along with the pathogen-specific antigen has been highlighted by the failure of peptide and DNA subunit vaccines, which have proved to be poorly immunogenic (Foged et al., 2002). Many studies have been, and continue to be, focused on exploiting the unique abilities of dendritic cells for providing co-stimulatory signals, as well as for antigen acquisition and presentation. So far, dendritic cells have been targeted in the rational
design of DNA-based vaccines (Collins and Cerundolo, 2004), vaccines and therapies against cancers (Nestle et al., 2005), optimization of vaccines against HIV (Ahlers and Belyakov, 2009) and other infectious diseases (Palucka et al., 2010; De Groot and Rappuoli, 2004), as well as in gene replacement therapy (Wu and Ertl, 2009) and tolerance to organ transplants (McCurry et al., 2006).

Our ability to manipulate dendritic cells in the development of these effective vaccines and therapies is directly related to our understanding of the interplay between dendritic cells and the multiple other cell types that together orchestrate the emergent adaptive immune response. This interplay involves complex interactions among dozens of cell types and thousands of different receptors and signaling molecules that operate both at the intracellular level, transducing signals that lead to phenotypic changes of the cells, as well as at the extracellular level, carrying information among cells. The complexity of this network is further increased by the addition of many positive and negative feedback regulatory mechanisms, and a substantial amount of redundancy, which is necessary to assure the robustness of the system. Hence, it has become increasingly clear that the conventional reductionist research method that focuses on single molecules and cells is not sufficient for a unified understanding of the large and complex network that comprises the immune response. The value of such scientific approach is undeniable, and much of our current knowledge of the immune response is due to studies conducted via this method. However, in order to be able to integrate this knowledge, we need to take a step back and look at entire ensembles of cells and molecules. Germain et al. (2011) provide an insightful analogy, comparing immune responses to symphonies of molecular and cellular interactions, where each of the players has its own role and all players together produce the composite behavior that leads to an effective host defense. They elaborate further, arguing that “just as listening separately to the notes played by the individual instruments fails to capture the ensemble effect achieved when an entire orchestra plays in unison, so
too are we limited in our understanding of how the immune system operates when we focus only on the properties or actions of one or a few unconnected components."

Technological advancements over the past decade or so have given us the ability to acquire data at an increased and unprecedented rate. A number of high-throughput platforms, such as microarrays and multiplexed flow cytometry, allow us to measure the state of entire ensembles of molecules and cells simultaneously. Therefore, these advancements have provided us the possibility to take a step back and approach aspects of the immune response from a broader, system-wise viewpoint. Using Germant's analogy, we can now listen to the entire orchestra instead of focusing on individual notes from a single instrument. The value of these large-scale data, however, is entirely dependent on our ability to extract meaningful information from them. This requires appropriate bioinformatics tools to organize and sort through such massive amounts of data, as well as computational, mathematical, and statistical models and methods to analyze them, identify trends, and extrapolate from them, making predictions and generating hypothesis. As such, these quantitative sciences have become integral to the advancement of modern immunology research.

The work described in this dissertation sits right in this intersection of large-scale immunological experimentation with computational and statistical modeling, which has become known as “computational immunology.” Here we present a series of studies that investigate two different aspects of dendritic cell biology relating to its ability to activate and modulate the adaptive immune response. We first investigate the changes in the dendritic cell transcriptional program over time in response to the recognition of pathogen-associated molecules. Then, we study dendritic cell presentation of peptides to T lymphocytes. In both studies, we have developed computational and statistical approaches designed to address specific biological questions and to take advantage of the properties and features of the data available.
1.1 Dissertation structure

The structure of the current dissertation is as follows. In Chapter 2, we describe relevant aspects of dendritic cell biology, providing the biological background that relates to the work presented throughout this dissertation. We focus specifically on dendritic cells’ unique ability to activate T cells by providing two types of signals: co-stimulatory and polarizing signals, delivered by an array of receptor interactions and other secreted molecules, and antigen specific signals, delivered by the presentation of peptides via MHC receptors. The former is related to the work presented in Chapters 4 and 5, whereas the latter is the focus of the work presented in Chapter 6.

In Chapter 3, we lay out the foundations upon which the methodological developments of Chapter 4 and 5 rest. We start by providing a brief introduction on the stochasticity of gene expression and the analysis of temporal gene expression data. We then move on to describing nonparametric Bayesian models, focusing on Gaussian processes and Dirichlet processes, which we use to model gene expression temporal profiles, and to cluster them, respectively.

In Chapter 4, we present and begin to analyze a temporal gene expression dataset produced in our laboratory by stimulating dendritic cells with either LPS (a bacterial wall component) or CpG (unmethylated DNA common to microbial pathogens). In order to analyze these data, we develop a method based on the Langevin process, a specific type of Gaussian process, to model the temporal mRNA profiles of genes. We use this model to analyze the entire dataset, being able to unravel interesting transcriptional patterns common and unique to either treatment. In Chapter 5, we continue to analyze the dendritic cell transcription data. Here, we present a clustering method that builds upon the model developed in Chapter 4, and use it to analyze the transcriptional profiles of groups of genes simultaneously.
Finally, in Chapter 6, we present our work that focused on the antigen-specific signal provided by dendritic cells in the activation of T cells. In particular, we present a method to perform predictions about the binding between peptides and various MHC alleles, which is relevant in the identification of antigens that can be potentially used in vaccines against specific pathogens.

1.2 Our contribution

1.2.1 Methodological context and accomplishments

Modeling gene expression temporal profiles

We have developed a novel methodology to analyze gene expression time series data. In our approach, the temporal expression profiles of genes are modeled with Langevin processes, which are Gaussian processes derived from the stochastic dynamical process represented by the Langevin equations. Specifically, we model the available temporal profiles of a given gene with additive Langevin processes, which allows us to compare temporal profiles directly. Although Gaussian processes have been extensively used in modeling time series data in a number of different fields, they have only been sparsely applied to the analysis of temporal gene expression data. Lawrence et al. (2006) inferred the concentration rates of transcription factor proteins using Gaussian processes. Yuan (2006) used Gaussian processes to compare the transcriptional response of genes under different conditions. Most recently, Kalaitzis and Lawrence (2011) applied Gaussian processes to rank differentially expressed genes. In all of these instances, the temporal gene expression data was modeled with Gaussian processes with the squared exponential covariance function, which is the standard covariance function used in regression. This covariance function is infinitely differentiable, which makes this model appropriate for modeling very smooth functions.

In addition to Gaussian processes, several other methods have been described to analyze this type of data. Yeung et al. (2001) analyzed cell cycle transcriptional
data using MCLUST (Banfield and Raftery, 1993), which models the data with the multivariate normal distribution. This model, however, was not tailored specifically toward time series data and does not take time dependencies into account, being invariant to permutations of time points. Hence, it is inappropriate for the analysis of temporal gene expression data. Medvedovic and Sivaganesan (2002) also uses multivariate normal distributions to model the gene expression data. Another approach for analyzing gene expression time series is spline regression (Hong and Li, 2004; Heard et al., 2006, 2005). While spline regression captures the temporal dependencies in the data, it has two main drawbacks: first, it imposes parametric form on the temporal profiles; second, the goodness-of-fit depends strongly on the number of basis functions utilized.

Gaussian processes represent a sensible alternative for time series analysis because, similarly to splines, they do account for time dependencies in the data. In contrast to splines, however, Gaussian processes do not impose any parametric form on the temporal expression profiles of genes, because they sample directly from the space of functions. Additionally, Gaussian processes allow for the modeling of non-stationary data, which is appropriate given our belief that gene expression is a non-stationary process. As such, Gaussian processes provide a flexible and appropriate approach to temporal gene expression modeling. In addition to inheriting all of these general advantages of Gaussian processes for modeling temporal gene expression, the Langevin process possesses other properties that make it particularly appealing for this task. The Langevin process models irreversible processes and its covariance function specifies that the correlation among time points increases over the course of the time course, which is exactly what we expect to happen in gene expression time series data. Finally, the Langevin process can be enlarged to include additional degrees of freedom.

Our key contributions to the modeling of time series gene expression data involves
the use of Langevin processes. As far as we are aware, this is the first application of a
Gaussian process derived from a dynamic stochastic process to temporal gene expres-
sion data. Additionally, this is the first work to model gene expression with additive
Gaussian processes, which represent a flexible way to model the data, account for
the control and treatment effects separately and perform direct comparisons among
transcriptional responses to different conditions.

Clustering multiple gene expression temporal profiles

We have developed a methodology to cluster gene expression expression patterns
that builds upon our Langevin process model. Cluster analysis aims to partition
heterogeneous data into smaller homogeneous subgroups, and it is an important step
towards gaining a greater understanding of large-scale gene expression data, such as
the dendritic cell dataset investigated here. It provides a global perspective of the
general types of transcriptional responses. Moreover, it aids in the identification of
groups of genes with similar temporal transcriptional responses to a given stimulus.
The rationale is that this similarity could be a result of genes being co-regulated,
sharing related functions, or participating in the same pathway.

Due to its importance, cluster analysis has been extensively explored in the con-
text of gene expression data analysis. Traditionally, heuristics algorithms such as k-
means, nearest neighbor and hierarchical agglomerative clustering, have been widely
used in analyzing gene expression data (Eisen et al., 1998; Tavazoie et al., 1999).
These methods continue to be used primarily by experimental biologists because
they are straightforward, easy to use and intuitively appealing. Much of the recent
developmental effort, however, has turned to model-based clustering methods, which
offer a principled alternative to ad-hoc procedures. Among the model-based cluster-
ing methods used in analysis of gene expression data, MCLUST (Fraley and Raftery,
1999, 2002) has been widely used. Notably, Yeung et al. (2001) used MCLUST, in
which clustering is done via a finite mixture of multivariate normal variables, to cluster yeast cell cycle gene expression data. As mentioned earlier, MCLUST was not tailored to cluster temporal data, and as such, does not take time dependencies into consideration. Ramoni et al. (2002) describes an autoregressive model that clusters stationary time series, which is probably inadequate for gene expression data since this type of data is unlikely to be stationary. Heard et al. (2006, 2005) proposed a Bayesian model-based agglomerative clustering scheme to cluster gene expression time series data, where the temporal profiles are modeled with splines.

The advantage of clustering with Dirichlet process mixture models (DPMM) over the methods described above is that DPMMs infer the number of clusters directly from the data, such that there is not requirement to pre-specify the number of clusters. Dirichlet processes are a popular approach in clustering applications, having been successfully use in many different domains, including in the clustering of gene expression data. For example, Ray and Mallick (2006) developed a Bayesian semiparametric wavelet model in which the parameters are modeled with the Dirichlet process. Medvedovic and Sivaganesan (2002) and Medvedovic et al. (2004) clustered gene expression temporal profiles with a Dirichlet process-induced mixture of multivariate normal distributions. Most recently, Dahl (2009) used DPMM for clustering gene expression univariate data.

Here we combine the Langevin process to model temporal gene expression profiles with the Dirichlet process to cluster these profiles. To the best of our knowledge, this is the first work to combine these two processes, using Dirichlet processes to cluster curves modeled with Gaussian processes. This approach combines the benefits of several of the methods described above, while avoiding their primary drawbacks. In terms of modeling the temporal profiles, our model accounts for the time dependency in the data and allows for unequal sampling intervals. Additionally, it is able to accommodate nonstationary and irreversible data, which are properties we expect in
gene expression temporal data. In terms of clustering, the use of DPMM allows us to bypass the need to pre-specify the number of clusters.

1.2.2 Biological context and accomplishments

We have presented a novel dataset containing dendritic cell temporal gene expression data. Murine dendritic cells were stimulated with one of two immune-stimulatory molecules, bacterial lypopolysacharide (LPS) or unmethylated DNA (CpG), and microarray mRNA measurements were obtained at eight different time points over the course of two days. As described above, we have developed methodology tailored specifically to analyze this dataset and compare the temporal transcriptional changes of dendritic cells in response to stimulation with LPS or CpG. Our goal has been to identify common and unique changes in the transcriptional response of dendritic cells in response to these two treatments.

Many other studies have investigated gene expression of various types of immune cells in both humans and mice (Abbas et al., 2005; Bles et al., 2007; Robbins et al., 2008; Messmer et al., 2003; Yang et al., 2011). In particular, several studies have compared the transcriptional response of macrophages stimulated with LPS and CpG. For example, Yang et al. (2011) contrasted the gene expression of murine macrophages in response to either six or 24 hours of stimulation with either LPS, CpG or poly I:C. They compared the responses at the two time points separately, and reported the number of transcripts differentially expressed in response to LPS to be five and three times greater than the number of differentially expressed transcripts in response to CpG at six and 24 hours, respectively. An earlier study (Gao et al., 2003) comparing gene expression of macrophages in response LPS and CpG stimulation also reported a larger number of genes differentially expressed in response to LPS than to CpG (measurements were taken at a single time point at six hours). Ramsey et al. (2008) compared the transcriptional response of murine macrophages
in response to CpG, LPS and four other TLR ligands over the course of 80 minutes, with measurements taken at 10 minutes intervals. They singled out TGIF1 as a main regulator of macrophage activation.

Fewer studies have specifically investigated the transcriptional changes of dendritic cells in response to these immune-stimulatory molecules. Klaschik et al. (2010) investigated the gene expression of murine splenic cells (which they reported to include B cells, CD4+ and CD8+ cells, monocytes, macrophages and plasmacytoid DCs) \textit{in vivo} over the course of 14 days, at time points zero, three, nine and 24 hours, three, five, seven, nine and 14 days. They reported the CpG transcriptional response to be bimodal, with one peak at three hours and a second peak at five days. Messmer et al. (2003) measured the gene expression of human dendritic cells stimulated with either LPS, CD40 ligand or CyC (a cocktail of inflammatory cytokines and prostaglandines) at two time points, 24 and 48 hours, and provided lists of genes classified as differentially expressed at these two time points.

In this context, our key contributions have been to provide the most extensive dendritic cell gene expression dataset comparing the temporal responses to LPS and CpG, as well as to provide the most thorough analysis of the temporal dynamics of these transcriptional responses. Most of the existing datasets comparing transcriptional response of immune cells to LPS and CpG were performed in macrophages and only include a single or just a few time points. Additionally, in the vast majority of these studies (with the notable exception of Ramsey et al., 2008) the methods employed to analyze the gene expression results were out-of-the-box \textit{ad-hoc} that may or may not be appropriate for the analysis of the given data. In other cases, such as in Yang et al. (2011), the time points were analyzed separately, such that the temporal dynamics of the response were not taken into account. Our analysis considers a time series including eight time points over the course of two days, carefully designed to capture most of the changes in the transcriptional response of DCs to
these two stimuli. For the rest of this section, we will describe the main findings of our analysis.

We have been able to characterize the overall transcriptional changes of dendritic cells in response to LPS and CpG stimulations. In addition to describing these global features of the two responses, we have also been able to identify commonalities and uniquenesses in the transcriptional dynamics of dendritic cells at the individual gene level. In general in this dataset, we have observed that the dendritic cell transcriptional response to CpG was stronger than that to LPS in two relevant aspects. The number of genes differentially expressed in response to CpG was more than the double of the number of genes differentially expressed in response to LPS. Furthermore, the magnitude of the change in the concentration of transcripts in dendritic cells stimulated with CpG was generally greater than in those stimulated with LPS. Regarding the timing of the changes in the transcriptional responses, we have identified both common and unique patterns. The transcriptional responses of dendritic cells to both LPS and CPG were bimodal. In both experiments, there was an early wave of transcription modulation that happened within two to three hours from the time of stimulation, and a later wave of transcription modulation that took place around 24 hours after the stimulation. These two waves of transcription were observed for both up- and down-regulated genes. The first wave of transcription in response to CpG, however, was generally faster than that in response to LPS, taking place about an hour earlier than the LPS response.

Regarding individual genes, we have identified both expected and unexpected differentially expressed genes in response to one or both treatments. Genes expected to be differentially expressed are the ones that have been previously reported in the literature to be activated in response to these treatments, such as cytokines, genes involved in the antigen-processing machinery and other genes that are involved in the maturation of dendritic cells. Among these, interferon beta presented a strong
upregulation of transcription with the same temporal profile in response to both stimulations. The alpha type I interferons, such as IFN-α2, IFN-α4, and IFN-α9, were both differentially expressed in response to both stimulations, however their temporal profiles were different for the two treatments. In general, the type I interferons (with the exception of IFN-β) were upregulated earlier in response to CpG than in response to LPS.

Among the genes that were differentially expressed in response to at least one of the two treatments, there were several genes that have not been reported in the literature to be involved in the dendritic cell response to neither LPS nor CpG, with many of them having no known function in any aspect of the immune response. Perhaps the most remarkable example is the transcription factor Tcf2, whose role in the immune response has never been described in the literature. This gene presented a fast and remarkably strong response to CpG stimulation, being the gene with the largest increase in mRNA level in the entire dataset.
A primer on dendritic cell activation of the immune response

2.1 The immune response

The detection and elimination of pathogenic microbial infections are essential for survival of the host. This can be a cumbersome task in light of the large diversity of microbes that can potentially infect the host and the fast rate with which some of them change. In vertebrates, this task is accomplished by a coordinated immune response that results from the interplay of the two branches of the immune system: the innate and the adaptive immune responses. The innate immunity provides an immediate and broad response against microbes, while the adaptive immune response is delayed and targets specific features of the invading pathogen. These two systems are complimentary to one another and together provide a very effective host defense.

The innate immune response is composed of cells and molecules that serve as a first line of defense against invading microbes. Macrophages, mast cells, neutrophils, natural killer cells and dendritic cells are all part of this branch of the immune system. These cells guard against invading pathogens, serving as sentinels of the host; as
such they typically are either circulating through the body or placed strategically at common points of entry of pathogens. Their primary functions are to detect the presence of invading pathogens, contain the spread of the infection and recruit specialized cells to the site of infection in order to completely obliterate the infection. Pathogens are identified by pattern recognition receptors (PRRs), which distinguishes self from non-self. Each of these PRR binds to molecules or parts thereof that are commonly found in microbes and invariant among entire classes of pathogens, but that are rare in vertebrates. Once the presence of a pathogen is detected, the innate immunity cells release anti-inflammatory and anti-viral molecules to contain the infection as well as signaling molecules to alert other cells about the imminent danger. They also release diffusible chemotactic factors to recruit other members of the immune system to the site of infection. In fact, the signals provided by the cells of the innate immunity, and particularly dendritic cells, are essential to determining the final outcome of the immune response. This is because depending on the cues they collect from the site of infection, dendritic cells provide different signals to the adaptive immune cells, essentially modulating their differentiation into effector cells. As such, the innate immune response not only provides a first line of defense, but also modulates the adaptive immune response.

While the innate immunity has been evolutionarily conserved and perfected over time in plants, invertebrates, and vertebrates, the adaptive immune system is relatively recent — it is exclusive to vertebrates (Hoffmann et al., 1999). It is composed primarily of cells known as T and B lymphocytes, and provides a highly specialized response that is tailored specifically toward the invading pathogen. In fact, the primary pathogen sensing mechanism utilized by the innate versus adaptive responses is one of the features that distinguish these two branches of the immune system. The receptors utilized by the innate cells are encoded in the germ line and recognize molecules or parts of molecules that have been evolutionarily conserved
among microbes, but that are, for the most part, lacking in vertebrates. As such, these receptors identify pathogens by distinguishing self from non-self. In contrast, individual receptors of the adaptive immunity do not make the self-non-self distinction; instead they simply identify specific antigens of the invading pathogen. This is achieved by clonal expansion of cells whose antigen-specific receptors have undergone an elaborate refinement process. This process is stochastic and entails genetic rearrangements (and in B cells, high rates of mutation) of the receptors in order to increase their specificity toward a particular antigen of the invading pathogen. Each lymphocyte displays receptors that will only bind to an extremely minute fraction of possible peptides, such that as a whole, the entire population of lymphocytes represents a repertoire of receptors with sufficient diversity to recognize any potential pathogen that the host might encounter. Such extravagance, of course, comes at a high cost to the host both in terms of resources and time. Regarding the former, because the receptor refinement process is random, inevitably receptors that recognize self antigens are produced. Thus, all cells presenting these self-recognizing receptors must be deleted in order to assure tolerance to self and avoid pathological immune responses. As for the latter, the receptor refinement process takes a substantial amount of time to complete, such that the acquired immune response takes days to reach its peak. Some of the advantages conferred by this branch of the immune system are that once these specialized cells have rearranged their receptors and differentiated into effector cells, they are incredibly effective in clearing infection and provide the host with “immune memory.” That is, once the infection has been cleared, most of the effector cells die, but some of them turn into “memory cells”, which remain in the body for long periods of time and can quickly respond to the same pathogen in subsequent infections.

Table 2.1 summarizes some of the key differences between the innate and adaptive immune responses. While the innate immunity provides a fast and broad response,
Table 2.1: Comparison of key aspects of the innate and adaptive immune responses. Table adapted from Janeway and Medzhitov (2002).

<table>
<thead>
<tr>
<th>Property</th>
<th>Innate immune system</th>
<th>Adaptive immune system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receptors</td>
<td>Fixed in genome</td>
<td>Encoded in gene segments</td>
</tr>
<tr>
<td></td>
<td>Rearrangement not necessary</td>
<td>Rearrangement necessary</td>
</tr>
<tr>
<td>Distribution</td>
<td>Non-clonal</td>
<td>Clonal</td>
</tr>
<tr>
<td></td>
<td>All cells of a class identical</td>
<td>All cells of a class distinct</td>
</tr>
<tr>
<td>Recognition</td>
<td>Conserved molecular patterns</td>
<td>Details of molecular structure</td>
</tr>
<tr>
<td>Action time</td>
<td>Immediate activation of effectors</td>
<td>Delayed activation of effectors</td>
</tr>
<tr>
<td>Self-nonself discrimination</td>
<td>Perfect: selected over evolutionary time</td>
<td>Imperfect: selected in individual somatic cells</td>
</tr>
</tbody>
</table>

the adaptive immunity provides specificity and memory. Although the innate response seems simpler than the adaptive, the fact that it is the only type of immunity in plants and invertebrates speaks to its effectiveness and evolutionary success. In vertebrates, the innate response not only activates and modulates the adaptive response, but also contains the infection during the period in which the adaptive immunity has not yet been fully activated. Janeway and Medzhitov (2002) refer to it as the delay tactics of the innate immunity. Both branches of the immune system contribute to the maintenance of the host-microbe homeostasis, and consequently, to the host survival. Together the innate and adaptive immune systems can, for the most part, effectively contain and eliminate infection.

2.2 Dendritic cells bridge the innate and adaptive immune responses

Between the innate and adaptive immune responses lie the dendritic cells, which play a critical role in bridging the two branches of the immune system. Dendritic cells (DCs) are produced in the bone marrow and migrate to various points of entry of pathogens in the host’s body. In fact, these cells are involved in the protection of virtually all tissues against infection (Janeway and Medzhitov, 2002). They remain
immature until they encounter an invading microbe. Immature DCs are primarily pathogen-sensing cells. They present strong endocytic capability and express a wide array of pattern recognition receptors (PRRs) in order to sense their environment and detect the presence of invading pathogens. Once they detect the presence of infection through these PRRs, they go through a developmental process called maturation. This entails a series of changes in their migration patterns as well as phenotypic characteristics. During maturation, DCs undergo a transient increase in their phagocytic capacity, followed by its downregulation. There is a concurrent upregulation of the expression of co-stimulatory molecules, cytokines, antigen-processing intracellular machinery, and antigen-presenting receptors, namely the major histocompatibility complex (MHC) molecules. There is also a change in the type of chemokine receptors expressed on their cell surface, which cause the DCs to migrate from the infection site to draining lymph nodes. As a result of the maturation process, DCs switch from being primarily pathogen-sensing cells with strong phagocytic capabilities to being very effective antigen-presenting cells.

Once the dendritic cells arrive at the draining lymph node, they have matured and are equipped to induce the activation of the adaptive immunity. In the lymph node, DCs interact with both CD4+ and CD8+ T cells, which require three signals to become activated (Kapsenberg, 2003; Arens and Schoenberger, 2010). Signal 1 is the antigen-specific signal. It is delivered by the interaction of the antigenic peptide bound to major histocompatibility complex (MHC) receptors on the surface of the DC with the cognate T cell receptor (TCR) expressed on the cell surface of the lymphocyte. This interaction will only occur at an appropriate level between DCs and T cells that have a TCR with the correct MHC-peptide specificity. Signal 2 is the costimulatory signal. This signal is decisive in the fate of T cells — Signal 1 in the absence of Signal 2 induces anergy or deletion of T cells, while Signal 1 in conjunction with the costimulatory signal leads to the activation of the T cell. Signal 2 is provided
by the costimulatory molecules that are upregulated on DCs during their maturation. This signal is mainly mediated by the triggering of CD28 on the T cell membrane by CD80 or CD86 on the surface of the dendritic cell. Although signals 1 and 2 have been the two known canonical signals in the activation of T cells, studies have shown that a third signal is also required (Arens and Schoenberger, 2010). Signal 3 is provided by the inflammatory cytokines interleukin-12 (IL-12) and the type I IFNs, and plays a role in effector T cell clonal expansion and function. A number of other cytokines also influence this process, such as IL-18 and IFN-γ. In addition to the contribution to T cell expansion, these cytokines and soluble factors released by the DC play an important role in determining the development of CD4+ T cells, biasing it toward a T\textsubscript{H}1 or a T\textsubscript{H}2 response. Because of this, Signal 3 is also referred to as the “polarizing signal” (Kalinski et al., 1999). For example, cytokines such as IL-12 and type I interferons (IFNs) are known to shift the adaptive response toward a T\textsubscript{H}1 response, whereas CCL10, IL-10 and TGF-β are T\textsubscript{H}2 polarizing factors.

The mechanisms through which dendritic cells parse the information collected from their environment and translate them into different patterns of receptor expression and secretion of cytokines and other signaling molecules in order to activate the appropriate branch of the immune response are rather complex and not entirely understood. In addition to the molecular nature of the ligand that binds to the pattern recognition receptor on the DC, inflammatory molecules released by other cells at the site of infection, the subtype of the DC, and the developmental state of the DC at the time of encounter with the pathogen are all factors known to influence the functional polarization of DCs during maturation (Kapsenberg, 2003). There is evidence that even the dose in which the antigen is presented to the dendritic cells changes their phenotype. For example, stimulation of DCs with low doses of LPS triggers a T\textsubscript{H}2 response, whereas stimulation of the same type of DC with high doses of LPS induce a T\textsubscript{H}1 response (Eisenbarth et al., 2002). The low dose of LPS would
be equivalent to that of inhaled antigen, and could be potentially associated with the mechanisms involved in the pathogenesis of asthma (Iwasaki and Medzhitov, 2004). In contrast, the large dose of LPS would be equivalent to that available during an infection with Gram-negative bacteria, in which case the T\textsubscript{H}1 would be the appropriate mechanism of defense. Finally, dendritic cells express a variety of combinations of PRRs, and it is unlikely that a given microbe will trigger only one of them. It is the interplay among the various triggered receptors and activated pathways that ultimately dictates the phenotype of the dendritic cell, and consequently the type of adaptive immune response that will be triggered.

In the rest of this chapter, we will describe in further detail the three signals provided by dendritic cells in order to activate T cells. Regarding the costimulatory and polarizing signals, we will give special emphasis to those induced by LPS and CpG, since they are the focus of a substantial part of the work presented in this dissertation.

2.3 Costimulatory and polarizing signals

The costimulatory and polarizing signals provided by dendritic cells dictate the fate of the T cell response. The costimulatory signal is provided by the interaction of surface receptors on the surface of DCs and T cells, and the polarizing signals is provided by cytokines released by DCs. The specific types and concentrations of receptors and cytokines, together with the number and subtypes of DCs recruited, determine the outcome of the T cell response, both in terms of the intensity and duration of the response, as well as in terms of the balance between the T\textsubscript{H}1 and the T\textsubscript{H}2 responses.

The appropriate balance between T\textsubscript{H}1 and T\textsubscript{H}2 responses is important for the well-functioning of the immune response. The T\textsubscript{H}1 response leads to cell-mediated immunity and the production of IgG. T\textsubscript{H}1 cells produce IFN-\textgamma and other proinflam-
matory cytokines, which lead to proinflammatory responses aimed at killing intracellular parasites. T\(_H\)2 cells produce primarily anti-inflammatory cytokines, such as IL-4 and IL-5, counteracting the proinflammatory effects of T\(_H\)1 in order to avoid uncontrolled inflammation which would lead to excessive tissue damage. The T\(_H\)2 response leads to production of IgE, which is often associated with allergic responses.

The hallmark cytokine required for the activation of T\(_H\)1 responses is IL-12. In fact, this cytokine has been described as a potent and obligatory inducer of differentiation of IFN-producing cells in vivo (De Becker et al., 1998). IL-12 seems to trigger the T\(_H\)1 response by activating the transcription factor STAT4 in CD4+ T cells. Type I interferons are also known to participate in the induction of T\(_H\)1 responses (Moser and Murphy, 2000). The contribution of dendritic cells to the development of the T\(_H\)2 response is less clear than that of the T\(_H\)1 response. One of the primary ways in which DCs trigger a T\(_H\)2 response seems to be via the release of IL-4, which activate CD4+ T cells via STAT6. IL-4-independent T\(_H\)2 activation mechanisms have been described, and involve the regulation of transcription factor GATA-3 (Moser and Murphy, 2000).

These costimulatory and polarizing signals are only provided by dendritic cells once they have matured. The phenotype of the activated dendritic cell, including the types of receptors expressed and cytokines produced, depends on its maturation process, which in turn depends on the environmental signals sensed by the immature DC that trigger its maturation process. DCs mature in response to “pathogen associated molecular patterns” (PAMP) recognized via “pathogen recognition receptors” (PRR), which are receptors genetically programmed to detect invariant features of the invading microbes. PRRs can be secreted, transmembrane or cytosolic. Examples of secreted PRRs are lectins, ficolins and pentraxins. Secreted PRRs bind to the pathogen surface and elicit complement pathways, leading to the opsonization of the pathogen and its phagocytosis by macrophages and neutrophils. Examples of cy-
tosolic PRRs are RLR (retinoic acid-inducible gene 1 (RIG-1)-like receptors), which detect viral pathogens. Two well-studied members of the RLR family of receptors are MDA4 and RIG-1. Finally, among the transmembrane PRRs are C-type lectins and Toll-like receptors (TLR). TLRs have been the most thoroughly investigated PRR and will be the focus of the rest of this section.

2.3.1 Toll-like receptors

The Toll-like family of receptors is the most thoroughly studied and best characterized class of pattern recognition receptors. They are named after the Toll receptor in *Drosophila*, whose role in immune defense was first described by Lemaitre et al. (1996). They demonstrated that a loss-of-function mutation in the Toll receptor of *Drosophila* resulted in high susceptibility to fungi infection and impaired host defense, normally characterized by the production of antifungal peptides. Shortly thereafter, a human homolog (now known to be TLR4) was identified, and demonstrated to be implicated in the mouse immune response to LPS (Medzhitov et al., 1997). These pioneering studies have revived interest in the innate immune system. Since then various other TLRs have been discovered in humans and mice, and much has been learned about their ligand specificities and signaling pathways.

TLRs are evolutionarily conserved, having been described in both invertebrates and vertebrates. In mammals, TLRs are strongly expressed in phagocytes, such as dendritic cells and macrophages. The quantitative and qualitative expression of the various TLRs in dendritic cells vary with the DC’s maturation stage and subtype. In humans, there is indication that most tissues express at least one type of TLR (Takeda et al., 2003). These receptors are transmembrane proteins with structurally conserved domains. They present a leucin rich repeat (LRR) domain in their extracellular region, which is commonly implicated in ligand binding and mediate the recognition of PAMPs. In the intracellular region, TLRs commonly present a Toll-
IL1 receptor (TIR) domain, which serves as a protein-protein interaction motif. The TIR domain of TLRs interacts with the TIR domains of various adaptor molecules present in the cytosol to trigger intracellular signaling pathways.

Since TLRs were first described a little over a decade ago, numerous members of this family have been identified in various species. The exact number varies among different species, and seems to range between 10 and 15 (Iwasaki and Medzhitov, 2004). So far, 10 members of the TLR family have been identified in humans, and 12 in mice. TLR1 through TLR9 are conserved between mice and humans, TLR10 is not functional in mice and TLR11 through TLR13 are not present in the human genome (Kawai and Akira, 2010). Each of these TLRs recognize one or more types of PAMPs. Broadly speaking, the TLRs can be divided into two rough groups based on their ligand and cellular localization.

The first group is composed of TLRs 1, 2, 4, 5, 6, and 11. These TLRs are expressed on the cell surface, and recognize primarily components of the membranes of microbes, such as bacteria and fungi. For example, TLR4 recognizes LPS from Gram-negative bacteria such as *Escherichia coli* and *Salmonella* spp. TLR2 recognizes several atypical types of LPS from other Gram-negative bacteria, like *Leptospira interrogans* and *Prophyromonas gingivalis* (Takeda et al., 2003). In fact, TLR2 appears to recognize the largest number of ligands. It forms heterodimers with both TLR1 and TLR6 to discriminate among various microbial components, such as lipoproteins and peptidoglycan from bacteria, and zymosan from fungi. TLR5 binds to flagellin, a conserved component of highly complex structure called flagellar present in the outer membrane of Gram-negative bacteria.

The second group of TLRs is composed of TLRs 3, 7, 8 and 9, which are typically located in intracellular compartments, such as early and late endosomes, and recognize primarily nucleic acids. TLR3 binds to double-stranded RNA (dsRNA), which is produced by many viruses as part of their replication process. TLR7 rec-
ognizes single-stranded RNA (ssRNA) from viruses, such as influenza A virus and human immunodeficiency virus. TLR8 is phylogenetically very similar to TLR7, and also recognizes ssRNA. However, TLR8 seems to play a less prominent role in the recognition of ssRNA as evidenced by studies showing that mice lacking TLR7, but not TLR8, respond normally to stimulation with ssRNA (Kawai and Akira, 2010).

Lastly, TLR9 detects unmethylated CpG DNA motifs, which are rare in mammals, whose DNA is mostly methylated, but common in bacteria because they lack the CpG methylation enzyme.

In addition to the molecular nature of the ligands and cellular localization, the various TLRs also differ from one another in several other ways. For example, the nucleic acid recognizing TLRs have the additional difficulty that their ligands are not necessarily exclusive to microbes. For those TLRs, accessibility to the ligand plays a big role in self versus non-self discrimination. As such, these receptors typically act within endosomal compartments, from which self DNA is typically excluded under normal functioning of the cells. TLRs also differ in their expression patterns across different cells. As mentioned briefly above, different dendritic cell subtypes express different combinations of TLRs. In humans, for instance, myeloid DCs express TLR1, 2, 4, 5 and 8, whereas plasmacytoid DCs (pDCs) express exclusively TLR7 and 9 (Takeda et al., 2003). There is also evidence that some cells that are not part of the innate immunity express TLR. Examples are B cells and some epithelial cells that line parts of the intestines. Finally, although most TLRs recruit many of the same downstream signaling molecules and largely share the same signaling pathways, there seems to be evidence that each TLR also use some unique signaling molecules, activating an overlapping yet distinct set of transcription factors. These elements unique to each TLR would presumably account for the differences in cytokines and receptors expression patterns induced by the different TLRs. Thus, although there is a lot of overlap and redundancy among the downstream events triggered by the
different TLRs, there are also uniquenesses which eventually lead to differential mat-
uration of DCs and modulation of the adaptive immune response. Understanding
these commonalities and uniquenesses is essential to our understanding of the im-
mune response in general, and to our ability to manipulate it with vaccines and
therapeutics.

2.3.2 TLR signaling

All TLRs trigger conserved signaling pathways that lead to the activation of nuclear
factor (NF)-κB and activating protein-1 (AP-1). These stimulate the production
of a variety of cytokines and other co-stimulatory molecules that can facilitate T
and B cell activation. However, each TLR elicits specific biological responses. At
least part of the differences in the physiological response triggered by the different
TLRs can be traced to what adaptor molecules they recruit. Upon recognition
of its cognate PAMP, the TLR triggers conserved inflammatory pathways by first
recruiting adaptor molecules. These adaptors initiate signaling cascades within the
cell, which commonly culminate with the translocation of transcription factors, such
as NK-κB, to the nucleus and the initiation of transcription of a number of genes. To
date, two sets of adaptor molecules used by the different TLRs have been identified.
The first consists of myeloid differentiation factor 88 (MyD88) and TIRAP, which
trigger pathways that result in the induction of inflammatory cytokines. The other
set of adaptors consists of TRIF and TRAM, which induce not only inflammatory
cytokines, but also type 1 interferons (IFNs).

The first adaptor molecule discovered was MyD88, which has a TIR domain that
interacts with the TIR domain in the intracellular tail of TLRs. It turns out that the
MyD88-dependent pathway is utilized by all known TLRs, except for TLR3 (Blasius
and Beutler, 2010). This conserved pathway results in the activation of NF-κB and
MAPK, both of which induce the transcription of inflammatory cytokines. Once trig-
gered by its cognate PAMP, the TLR recruits MyD88, whose binding is facilitated by another TIR containing adaptor molecules, TIRAP. MyD88 also contains a death domain (DD) which serves to recruit the IRAK proteins. These form a complex with TRAF-6, which adds a polyubiquitin tail to itself as well as to IRAK-1. These polyubiquitin tails serve as docks for TAK-1, TAB2, and TAB3, which activate an IKK complex that phosphorilates the NF-κB inhibitor, IκB. Upon phosphorilation, IκB is tagged for degradation by the proteosome, releasing NF-κB, which then translocates to the nucleus, where it initiates the transcription of its target genes. TAK-1 also phosphorilates MEK1, which activates a MAPK pathway that eventually leads to the activation of members of the AP-1 family of transcription factors, such as JNK and p38. This MyD88-dependent pathway plays a critical role in TLR-induced DC maturation and subsequent induction of adaptive response. It is essential for signaling by many TLRs, with one remarkable exception being TLR4, which uses an alternative signaling pathway in addition to the MyD88-dependent pathway. Interestingly, lack of MyD88 seems to result in a selective defect in the mounting of the Th1 but not Th2 response. This is consistent with the fact that TLRs are essential for the recognition of pathogens, such as prokaryotic, fungal, viral, or protozoan pathogens, whose PAMPs are known to trigger a Th1 response (Janeway and Medzhitov, 2002).

In addition to the classic MyD88-dependent signaling pathway, it has been shown that MyD88 operates through a different pathway in response to the triggering of TLR7 and TLR9, which are expressed exclusively in plasmacytoid dendritic cells. The MyD88-dependent pathway elicited by these two TLRs trigger the induction of type I interferons in addition to the pro-inflammatory cytokines. Honda et al. (2005) showed that the induction of interferons in this pathway is mediated primarily by IRF7, which is constitutively expressed in plasmacytoid DCs. In this context, MyD88 recruits IRF7 and a number of intracellular adaptors, such as IRAK1, IRAK4, TRAF6 and TRAF3, forming a multiprotein complex. This complex brings
IRF7 in close proximity to IRAK1, so that IRAK1 can phosphorilate IRF7. This causes IRF7 to dissociate from the complex and translocate to the nucleus, where it triggers the transcription of type I interferons. Although IRF7 is the master regulator of type I IFN in these cells, other IRFs play secondary roles in the production of IFN. Honda et al. (2005) demonstrated that while the induction of IFN-α is entirely dependent on IRF7, transcription of IFN-β is triggered by IRF3 in addition to IRF7.

The second set of adaptor molecules that mediate TLR signaling consists of TRAM and TRIF. These are recruited by TLR4 and TLR3. In fact, the TRIF-dependent pathway is the sole signaling mechanism utilized by TLR3. These adaptor molecules trigger pathways that induce the production of both inflammatory cytokines and type I IFNs. The signaling cascade that leads to the production of inflammatory cytokines share many common elements with the MyD88-dependent pathway: it operates via the recruitment of TRAF6, which induces the NF-κB and MAPK pathways by activating TAK-1. The type I IFN-inducing pathway triggered by TRIF involves the recruitment of a non-canonical IKKs, TBK1 and IKKe. These IKKs form a complex that catalyze the phosphorilation of interferon inducible factor 3 (IRF3), allowing it to translocate to the nucleus, where it initiates the transcription of type 1 IFN genes.

For the most part, the location of the TLR dictates the signaling cascade used, and consequently the set of genes that are induced. For instance, type I interferons are produced solely by the recognition of ligands present in the cytosol or withing the lumen of intracellular compartments (Blasius and Beutler, 2010). As explained in detail in Section 2.3.3, TLR4 induction of IFN depends on the transport of PAMP-bound TLR4 to endosomal compartments, where the MyD88-dependent pathway is inhibited and the TRIF-dependent pathway is activated, leading to the production of IFNs.
2.3.3 Activation of DCs by LPS via TLR4

Lipopolysaccharide (LPS) is the main component of the outer membrane of Gram-negative bacteria. Because it is crucial for the structural integrity of these bacteria, it has been conserved in these organisms. LPS is an endotoxin that elicits strong inflammatory responses in mammals, and in the case of excessive signaling, it can induce systemic inflammation and sepsis (Beutler and Rietschel, 2003). The LPS-sensing machinery is rather complex, and it began to be elucidated by two seminal works showing that LPS binds lipopolysaccharide binding protein (LBP) (Tobias et al., 1986), and then membrane-bound receptor, CD14 (Wright et al., 1990). This complex, however, lacks a cytoplasmatic domain as well as the ability to traverse the cell membrane; as such, it was apparent that at least another molecule with cytoplasmatic domain responsible for the signaling transduction was part of the complex. It was not until a decade later that this molecule was shown to be TLR4 (Medzhitov et al., 1997; Poltorak et al., 1998; Qureshi et al., 1999), which was in fact the first Toll-like receptor identified in mammalian species (Medzhitov et al., 1997).

TLR4 is a transmembrane complex presenting leucin-rich repeats (LRR) on its extracellular domain, and both a TIR (Toll-interleukin-1 receptor) domain and a death domain on its intracellular tail. Upon recognizing the CD14-LBP-LPS complex, TLR4 undergoes oligomerization, which allows for the recruitment of its downstream adaptor molecules through protein-protein interactions via the TIR domains. This, in turn, initiates the signaling cascade that culminates with the translocation of transcription factors, such as NF-κB and interferon regulatory factors, to the nucleus and the transcription of proinflammatory cytokines, interferon genes, and interferon-inducible genes.

TLR4 signals via both the MyD88 and the TRIF pathways (described in Section 2.3.2), with the former pathway being responsible for the production of proin-
flammatory cytokines, and the latter pathway triggering the induction of type I interferons and interferon-inducible genes (Lu et al., 2008). In fact, TLR4 is the only TLR known to utilize all four adaptor molecules, MyD88, TIRAP, TRIF and TRAM. TLR4 first signals through MyD88 while it is still on the cell surface, then it is sequestered into endosomal compartments. At this point, the MyD88 pathways is inhibited and the TRIF pathway is activated. TRIF plays a key role in the activation of transcription factor IRF-3, and late phase activation of NF-κB and MAPK (Lu et al., 2008).

2.3.4 Activation of DCs by CpG via TLR9

Bacterial DNA is known for being one of the major immunostimulatory types of pathogen associated molecular pattern. They are characterized by unmethylated CpG motifs, which contrasts with the highly methylated mammalian DNA. In 2000, TLR9 was identified as the long sought receptor for unmethylated CpG DNA (Hemmi et al., 2000). Since then, TLR9 has been implicated in the immune response against a number of bacteria, such as *Mycobacterium tuberculosis*, *Streptococcus pneumoniae*, and *Helicobacter pylori*, and viruses that also have unmethylated nucleic acid, such as mouse cytomegalovirus and herpes simplex virus Kumagai et al. (2008).

TLR9 is highly expressed in plasmacytoid dendritic cells, but has also been shown to be present in other types of cells, such as conventional dendritic cells (cDCs), B cells and macrophages (Barber, 2011). It has a strong ability to induce Th1 response, and plays a pivotal role in the activation of dendritic cells in response to viral and bacterial DNA. It does so both directly and indirectly, by facilitating the cross-talk between plasmacytoid and conventional DCs. TLR9 stimulation leads cDCs to produce IL-15, which induces the expression of CD40L on pDCs. The interaction of pDCs with cDCs via CD40-CD40L binding activates the production of IL-12 by cDCs. TLR9-activated DCs also participate in the activation of NK cells. Finally,
TLR9 stimulation directly activates B cells, inducing its proliferation and suppressing class switch from IgM and IgG2a to IgG1 and IgE (Pasare and Medzhitov, 2005). Kumagai et al. (2008) notes that this switch is also suppressed by T\(_H\)1 response, suggesting TLR9’s role in the activation of B cells and the suppression of allergic T\(_H\)2 response.

Differently from most TLRs, which are present on the cell membrane, TLR9 is sequestered into the endoplasmic reticulum in unstimulated cells. Upon ligand stimulation, it is quickly translocated to endolysosomes. This trafficking of TLR9 has been shown to be assisted by a transmembrane protein called UNC93B (Kumagai et al., 2008). In fact, mutations in this protein not only obliterate trafficking of TLR9 but also signaling (Kawai and Akira, 2010). Once in the endolysosome, TLR9 is proteolytically cleaved by proteases; this leads to the functional form of this receptor, which can then initiate its signal transduction cascade. TLR9 depends on MyD88 to trigger its intracellular signaling pathway. As described in Section 2.3.2, MyD88 forms a complex with IRAK1 and IRAK4, among other molecules, which triggers a signaling cascade that results in the activation of IRF5, NF-κB and AP-1 transcription factors (Kawai and Akira, 2006). Together, these transcription factors upregulate the expression of a number of inflammatory cytokines, such as TNF-α, IL-6, IL-1\(_β\) and IL-12, as well as MHC class II and costimulatory molecules.

In plasmacytoid dendritic cells, TLR9 signaling also leads to the production of large amounts of type I interferons. It turns out that these cells, also known as “professional IFN-producing cells”, constitutively express IRF7 in an inactive form, unlike most other subtypes of DCs. Upon formation of the MyD88 and IRAKs multiprotein complex, IRAK1 is brought into close proximity with IRF7, phosphorylating it. IRF7 then detaches from this complex and translocates to the nucleus, triggering the transcription of type I IFNs as well as of interferon-inducible genes (Kumagai et al., 2008). Type I interferon production can also be induced in cDCs stimulated
with CpG. In these cells, MyD88-activated IRF1, and not IRF7, is required for the expression of IFN-β (Negishi et al., 2006).

TLR9 has been long known to be activated by unmethylated CpG. The rationale for why mammalian DNA does not elicit a TLR9 response had been based on primarily three things: 1) CpG motifs are relatively rare in mammals in comparison to microbial DNA; 2) mammalian DNA is methylated; and 3) there could be some inhibitory motifs in mammalian DNA that prevented it from triggering TLR9 (Yasuda et al., 2005). Questions, however, have been raised as to whether the sequence of DNA itself (i.e., the CpG motifs) is the trigger of TLR9, suggesting instead that features of the backbone of the DNA are the actual triggers of TLR9 (Blasius and Beutler, 2010). Yet another study demonstrated that vertebrate DNA does indeed induce TLR9-dependent activation of dendritic cells when transfected directly into endosomal compartments (Yasuda et al., 2005). This and other studies have demonstrated that rather than specific characteristics of the DNA, ultimately the localization to endosomal compartments dictates TLR9 activation. Uptake of microbial DNA into endolysosomes could happen in two main ways. First, viruses can be delivered to these compartments through receptor-mediated phagocytosis or endocytosis, or non-specific fluid phase endocytosis. Second, viruses that have already infected the cell and are present in the cytosol could be delivered to endolysosomes via autophagy of the cell. In contrast, under normal functioning of cells, self DNA does not have access to endosomal compartments.

Thus, TLR9 recognizes nucleic acid released into endosomes after the virus or bacteria have been internalized and lysed. Once viruses enter the cytosol and initiate replication, however, TLR9 no longer has access to its DNA. Studies have shows that INF-β production in response to a number of viruses remains unaffected in TLR9-deficient mice (Kawai and Akira, 2010). Indeed, TLRs are not the only DNA sensors in dendritic cells. Other molecules and pathways believed to be involved in
the recognition of microbial DNA have been described. One alternative pathways utilizes RNA helicases present in the cytosol, RIG-1 and MDA, to detect viruses within the cell. This pathway leads to the activation of IRF3 and NF-κB (Kawai and Akira, 2006). Yet another pathway using unidentified DNA sensors triggers expression of type I IFNs (Kawai and Akira, 2010). Very recently, a new molecule has been identified, called stimulator of interferons (STING), as being an integral part of this pathway (Barber, 2011), and surely more aspects of this pathway will continue to be revealed as research in this topic remains active.

2.4 Antigen-specific signal

The antigen-specific signal is delivered by the interaction of peptide-MHC complexes present on the surface of the dendritic cell with their cognate TCRs on the surface of the T cell. The TCR binds to the peptide-MHC complex with high specificity, such that it can only appropriately bind to one such combination of peptide and MHC receptor. In addition to the peptide-MHC recognition molecules, there are two other complexes that associate with TCR and participate in the transduction of the antigen-specific signal: 1) a co-receptor, either CD8 or CD4, which aids in strengthening the bond between the TCR-peptide-MHC complex; and 2) a signaling complex, called CD3, which triggers the intracellular signaling cascade when the TCR binds to its cognate peptide-MHC complex. This signaling cascade, together with those initiated by costimulatory and polarizing signals, leads to the differentiation and clonal expansion of the T cells. This results in a population of effector T cells with the same peptide-MHC specificity.

As described earlier, upon detection of infection via their pattern recognition receptors, dendritic cells transiently increase their phagocytic capacity before shutting it down. Additionally, DCs upregulate the expression of MHC molecules as well as the expression of molecules that participate in the antigen processing machinery.
These features not only increase the concentration of peptide-MHC complexes on the surface of the DCs, but also makes this population of peptide-MHC complexes enriched for those presenting antigens captured at the time of sensing the infection signal (Masson et al., 2008). The high concentration of peptide-MHC complex, along with the expression of costimulatory molecules, make DCs incredibly proficient antigen-presenting cells (APC) with an exquisite capacity to activate both CD8+ and CD4+ T cells.

There are two classes of MHC molecules involved in the presentation of antigens to T cells: MHC class I (MHCI) and MHC class II (MHCII). The molecules belonging to these distinct MHC classes share many structural and functional features, but they differ in subtle ways that make them most fit for the different roles they play in the activation of the adaptive immune system. They differ in three most important ways: the nature of the antigens presented, the type of cells in which they are expressed, and the type of T cell with which they interact. In the following sections, we will describe some of the most relevant of these differences and similarities from the perspective of a dendritic cell.

2.4.1 Antigen presentation by MHC class I

The major histocompatibility complex (MHC) class I is a family of protein complexes that are expressed in almost every type of cells in humans, as well as in most vertebrates (Kindt et al., 2006). They participate in the activation of the cytolytic branch of the adaptive immune response, which is responsible for destroying cells that are malfunctioning, such as cells that have been infected by intracellular pathogens or that have suffered deleterious mutations. MHC class I is a transmembrane complex that binds to a peptide and presents it at the cell surface to CD8+ T lymphocytes, also called cytotoxic T lymphocytes (T_c cells or CTLs) for their ability to induce apoptotic death of their target cells. CTLs with receptors that recognize self peptide-
MHCI complexes are deleted or permanently inactivated during development, such that theoretically in a healthy individual all circulating T cells will only recognize MHCI receptors bound to nonself peptides. When an activated CD8+ T cell encounters a cell expressing its cognate peptide-MHCI complex, it secretes factors that kill the cell presenting the foreign peptide. In this sense, MHCI serves as a window through which the immune system inspects the interior of virtually all cells in the body, checking whether they are normal and destroying the ones that are anomalous, thereby ensuring that the body as whole is in a healthy state.

Although most cells express MHCI, only professional APCs are capable of activating CD8+ T cells. This is due to the fact that these cells lack the costimulatory molecules expressed by professional APCs (recall that, in addition to the antigen-specific signal provided by the binding of TCR to peptide-MHCI complexes, T cells also require the costimulatory and polarizing signals). In fact, studies have suggested that dendritic cells might be the exclusive APCs priming naive antigen-specific CD8+ T cells, suggesting that other APCs such as macrophages and B cells only play limited roles in vivo (Masson et al., 2008).

MHCI typically binds to peptides that are produced within the cell cytoplasm by the ongoing process of protein turnover, which is part of the normal functioning of cells. Intracellular proteins are degraded into short peptides by a proteolytic system present in all cells. The major member of this system is the proteosome, a multisubunit protein complex, which is aided by several other proteases and peptidases. In the presence of type I interferons, some of the subunits of the proteosome are replaced with others, which catalize cleavage of peptides such that the peptides produced by it have increased MHCI affinity. This proteosome is commonly referred to as the immunoproteosome. Peptide fragments are then translocated to the lumen of the endoplasmic reticulum (ER), where they bind to MHCI molecules. Finally, the MHCI complex is transported to the cell membrane via the Golgi apparatus,
where it display the peptide for inspection by CD8+ T cells.

For a long time it was believed that MHCI was only capable of presenting peptides derived exclusively from proteins synthesized within the cytosol via the pathway described above (Kindt et al., 2006). It turns out, however, that this is not always the case. Professional antigen presenting cells, and especially dendritic cells, have the capacity to present exogenous peptides via MHCI (Rock and Shen, 2005). This process is termed cross-presentation (of antigens) or cross-priming of (CD8+ T cells), and can happen via two distinct pathways. In both cases the protein is acquired by the APC via phagocytosis or endocytosis. Then, in one of the pathways, the antigen is transferred into the cytosol and processed via the classic MHCI antigen processing pathway just described. In the second pathway, the protein is degraded and bound to MHCI within the endocytic compartments themselves, never accessing the cytosol (Rock and Shen, 2005). We now know that cross-priming plays a crucial role in immune surveillance.

**MHC class I structure**

MHCI molecules are heterodimers, composed of a transmembrane heavy chain (the α-chain) and a water-soluble globular chain (the β2-microglobulin) (Bjorkman et al., 1987). The membrane-distal end of the complex comprises a groove in which the peptide binds to MHCI and is presented to T cells. An eight-stranded anti-parallel β-sheet forms the base of the groove, and two parallel α-helices make up its two side walls, encountering each other at both ends of the groove and completely blocking them (Madden, 1995). The MHC locus has a remarkably high level of polymorphism (Klein et al., 1993) and the most variable residues are located in the peptide-binding groove and on the top region of the α-helices, where the MHC molecule interacts with the TCR (Madden, 1995; Bjorkman et al., 1987). This different level of polymorphism at the various functional parts of the MHC structure allows each molecule to preserve
the overall MHC shape and simultaneously have unique peptide- and TCR-binding properties (Madden, 1995; Klein et al., 1993), such that each MHC molecule binds a discrete, but large and diverse, set of peptides.

The closed structure of the groove imposes strict constraints on the length of peptides that can bind to it, which typically range between 8 and 10 amino acids, although peptides as long as 14 residues have been reported to bind to class I molecules (Burrows et al., 2006). Two clusters of residues, one at each end of the binding cleft, are conserved among virtually all class I molecules (Madden, 1995; Zhang et al., 1998) and serve to anchor the ends of the peptides, such that short peptides bind in a fairly straight conformation and longer peptides bulge in the middle of the groove. The side chains of the residues that make up the β-sheet and the sides of the α-helices form dips and ridges in the groove, imposing additional stereo-chemical constraints to peptides, which are important in determining what peptides are able to bind to the MHC groove (Madden, 1995; Zhang et al., 1998).

2.4.2 Antigen presentation by MHC class II

The major histocompatibility complex class II (MHCII) is a family of protein complexes that are expressed only in specialized professional antigen-presenting cells (APC). In particular, they are expressed in dendritic cells, and activated B cells and macrophages. MHC class II is a transmembrane heterodimer, composed of two homologous peptides, an α and a β chain. It binds primarily to exogenous peptides, which are peptides that are not produced within the cytosol of the cell. These peptides originate from either the extracellular environment or from parasites, such as the protozoan Leishmania and bacteria Mycobacterium tuberculosis, that reside inside intracellular vesicles in cells like macrophages. Once bound to a peptide, MHC class II is transported to the extracellular membrane of the APC, where it presents the peptide to CD4+ T lymphocytes, also called helper T cells, which are essential
in the activation of CTLs and of B cells, promoting B cells antibody class switch.

MHC class II molecules are assembled in the endoplasmic reticulum (ER), and translocated to the endocytic pathway in an inactive form, in which the complex is associated with a protein called MHC class II-associated invariant chain (Ii), which prevents binding of the complex to peptides present in the ER. In the absence of the Ii or a bound peptide, the MHCII is unstable and aggregates. Additionally, the Ii is involved in directing the MHC class II complex out of the ER and into a low-pH endosomal compartment. Once in the endocytic pathway, the Ii is cleaved, and peptide loading onto the MHCII can occur. MHCII molecules that do not bind peptide after dissociation from the invariant chain are unstable; as such, they eventually aggregate and are degraded. Another important molecule that participates in the exogenous antigen processing and presenting pathway is HLA-DM. This molecule catalyzes the release of unstably bound peptides from MHCII heterodimers. This process, also referred to as peptide editing, ensures that peptide-bound MHCII complexes are stable enough in order to last the necessary time for proper stimulation of CD4+ T cells.

MHC class II structure

Similarly to MHC class I, MHC class II is a transmembrane dimeric complex in which the membrane-distal part comprises a highly polymorphic peptide-binding groove whose base is a β-sheet and sides are α-helices. One of the α-helices is shorter and closer to the base of the groove than the corresponding one in MHCII. In addition, the side chains present at the ends of the groove are smaller than those in MHCII or are reoriented away from the middle of the groove (Madden, 1995). As a consequence, the MHCII groove is open at both ends, allowing binding of peptides of various lengths, such that a core of 9 amino acids (called the register) bind to the groove and the rest of the residues extend out of it from both ends (called the peptide-flanking
residues, PFRs). Peptides as short as 10 and as long as 30 amino acids have been observed to bind to MHC class II (Sercarz and Maverakis, 2003). In fact, the affinity of peptides to MHCII has been shown to increase with the peptide length (Chang et al., 2006; Sercarz and Maverakis, 2003) and two mechanisms that explain this advantage have been proposed. First, the peptide’s residues that extend past the groove not only shield the residues interacting with the groove from being charged or degraded (Max et al., 1993), but they also interact with MHCII residues outside of the groove (Max et al., 1993; Chang et al., 2006; Sercarz and Maverakis, 2003). The second explanation is that longer peptides shift within the MHC class II binding-groove, changing the register that sits on the groove, such that longer peptides have more registers and thus higher chances of having a high affinity register.
A substantial part of this dissertation focuses on the analysis of the gene expression temporal data we have produced in our laboratory by stimulating dendritic cells with different adjuvant molecules. In Chapter 4, we present the model we have developed to analyze gene expression time series. Our model is based on a Gaussian process whose covariance function is generated from a stochastic dynamic system, the Langevin equations. In Chapter 5, we build upon this model, coupling our Gaussian process with a Dirichlet process to cluster the gene expression temporal profiles in order to identify common transcriptional profiles of genes in response to the same treatment. In both cases, the methods and models presented rely on the well-established Bayesian nonparametric framework.

The goal of the current chapter is to lay the statistical foundation upon which we will build in the next two chapters. Here, we will provide the relevant background necessary for an understanding of the methodology developed in following chapters. We start by describing the general Bayesian nonparametric framework in Section 3.1, providing a brief overview of these methods and of the role they play in model selection. Then, in Section 3.2, we move on to Dirichlet processes, exploring their
various representations, and their contribution to the field of mixture models. Finally, in Section 3.3, we explore Gaussian processes, providing some emphasis to their covariance functions, which play a crucial role in Gaussian process models.

3.1 Nonparametric Bayesian models

Nonparametric Bayesian models are models in which the distribution describing the data cannot be indexed by finitely many parameters. Because of this, Müller and Quintana (2004) suggested that a more descriptive term for nonparametric models would be “massively parametric” models. Although apparently contradictory, the term nonparametric is fitting in the sense that these models do not impose parametric assumptions on distributions. By extending the parameter space into infinitely many dimensions, these models are able to avoid the restrictiveness of parametric models, thereby encompassing all the possible solutions (under the given modeling assumptions) to the question at hand. In the case of clustering, this infinite-dimensional parameter space is the set of all partitions of the data; that is, all possible arrangements of the observed data into any number of clusters.

As clearly described in Orbanz and Teh (2010), despite spanning an infinite-dimensional parameter space, nonparametric Bayesian models are tractable even for large datasets because they evoke only a finite subset of the parameters when evaluated on a finite data set. In this way, the complexity of the models, as measured by the number of parameters, is adaptable and changes with the sample size. In practical terms, given a finite sample, although the parameter space is unbounded, the number of parameters in a nonparametric Bayesian model turns out to be finite. This is in direct contrast with many parametric models, where the number of parameters may be fixed \textit{a priori} and independent of the observed data.

Nonparametric Bayesian models represent an alternative to the problem of model selection that arises in parametric modeling. As mentioned above, in parametric
modeling the number of parameters is fixed, determined \textit{a priori} independently from the data, which often results in model mis-specification. Selecting the “best” model for a dataset given a number of parametric models is not a straightforward task. In fact, although many approaches to model selection exists, such as approaches based on MAP estimates, maximum likelihood and Bayesian information content, model selection continues to be an open problem with no optimal solution. Nonparametric Bayesian models avoid the difficulties present in model selection, automatically balancing goodness-of-fit with model complexity by using priors with very large support (which allow the parameters to vary greatly) and posterior distributions over the parameters (which, given that all else is equal, naturally favor simpler models over more complex models).

### 3.1.1 Some examples of Bayesian nonparametric models

There exists a vast literature on nonparametric Bayesian models related to model selection in areas ranging from clustering and density estimation to regression and latent feature models. In clustering problems, Dirichlet process (DP) priors represent useful nonparametric alternatives to existing parametric approaches; some examples are Escobar and West (1995), Teh (2010), MacEachern and Müller (1998). As will be discussed in Section 3.2, a convenient consequence of using DP prior on the cluster parameters is that the number of clusters is inferred directly from the data; as such the problem of model selection is avoided altogether. In density estimation problems, one Bayesian nonparametric approach involves using Pólya trees (Lavine, 1992) as priors on probability distributions. Pólya trees are defined by an infinite recursive binary partition of the sample space, where each sub-space defined by the partition is associated with a beta random variable that determines the relative amount of mass put on each side of the binary partition. Incidentally, DPs are special cases of Pólya trees. Much like the Dirichlet processes, Pólya trees are distributions over function
spaces, but their advantage over DPs as priors in density estimation problems is that they can generate both discrete and continuous distributions.

In nonlinear regression problems, one wishes to infer a continuous function given a training set consisting of input-output pairs. In parametric models, a function is parametrized and the observed data is used to infer the parameters of this function. The Bayesian nonparametric approach to nonlinear regression consists of placing a prior over continuous functions instead of giving an explicit parametric form for the function. A common choice of priors for nonparametric nonlinear regression are Gaussian processes (Williams, 1999; MacKay, 1998), which are generalizations of Gaussian distributions over infinitely many parameters, such that the joint distribution of each finite subset is described by a multivariate Gaussian distribution.

The above are just a few examples of Bayesian nonparametric models applied to different types of statistical problems. There are many other examples, such as beta processes and the Indian buffet process in applications with latent features (Fox et al., 2010; Thibaux and Jordan, 2007), and infinite HMMs to model temporal data (Fox et al., 2008).

The field of nonparametric Bayesian models continues to be energetically developed, with the Dirichlet process being by far the most popular Bayesian nonparametric prior used in the literature. For reviews of nonparametric Bayesian models see Walker et al. (1999), Orbanz and Teh (2010) and Dunson (2010).

3.2 Dirichlet process models

3.2.1 The Dirichlet distribution and its aggregation property

The Dirichlet distribution is the multivariate generalization of the beta distribution and, as such, it is commonly used to describe the distribution of unknown probabilities. Just as the beta distribution is the natural conjugate prior for the binomial distribution, the Dirichlet distribution is the conjugate prior for the multinomial dis-
distribution. In fact, its conjugacy with the multinomial distribution has placed the Dirichlet distribution as an important player in Bayesian statistics, being frequently used in applications involving order statistics and, in general, categorical data analysis as well as in nonparametric and mixture modeling.

An intuitive aspect to the Dirichlet distribution is the Pólya urn scheme. Consider an urn with \( n \) balls of \( k \) different colors, such that there are \( n_j \) ball of color \( a_j \), with \( j = 1, \ldots, k \). The Pólya urn scheme proceeds as follows:

1. Draw a ball, \( x_i \), from the urn, where the balls are drawn with equal probability;
2. Look at its color, \( a(x_i) \), and return it to the urn along with another ball of the same color.

Perform this task \( N \) times. Then, the probability that the \((n+1)^{th}\) ball will be of color \( a_j \) will be

\[
p(a(x_{n+1}) = a_j | a(x_1), \ldots, a(x_n)) = \frac{n_j + \sum_{i=1}^{N} \delta_{a_j}(a(x_i))}{n + N},
\]

where \( \delta_x(\cdot) \) is a point mass distribution at \( x \). As \( N \) approaches infinity, the proportions of colors in the urn will follow a Dirichlet distribution, \( \text{Dir}(a_1, \ldots, a_k) \).

Formally, the random \( k \)-dimensional vector \( \theta = (\theta_1, \ldots, \theta_k) \) follows a Dirichlet distribution, denoted as \( \theta \sim \text{Dir}(\mathbf{a}) \), if \( \theta_i > 0 \) for \( i = 1, \ldots, k \) and \( \sum_{i=1}^{k} \theta_i = 1 \), and its probability density function is

\[
p(\theta_1, \ldots, \theta_k | a_1, \ldots, a_k) = \frac{1}{B(\mathbf{a})} \prod_{i=1}^{K} \theta_i^{a_i - 1},
\]

where the \( k \) elements of the parameter vector \( \mathbf{a} = (a_1, \ldots, a_k) \) are constrained to be positive, \( a_i > 0 \) for \( i = 1, \ldots, k \). The normalizing constant of the Dirichlet distribution is the multinomial beta function, \( B(\mathbf{a}) = \frac{\prod_{i=1}^{k} \Gamma(a_i)}{\Gamma(a)} \), with concentration
parameter $\alpha = \sum_{i=1}^{k} a_i$. Although $\theta$ has $k$ elements, this distribution is in practicality a distribution on $k-1$ dimensions, since the $k^{th}$ item in $\theta$ is implied by the rest of $\theta$, $\theta_k = 1 - \sum_{i=1}^{k-1} \theta_i$.

The intuition behind the parameter vector of the Dirichlet distribution, $\mathbf{a}$, is best described by its sum, $\alpha$, which is referred to as the concentration parameter or the precision parameter. As $\alpha$ decreases, the variance of $\theta_i$ increases and $\theta_i$ becomes more diffused. As $\alpha$ increases, $\theta_i$ concentrates around its mean, $a_i/\alpha$. When $\alpha = 1$ and $a_i = \alpha/k$ for $i = 1, \ldots, k$, the Dirichlet distribution simplifies to a uniform distribution, which can be used as a non-informative prior to reflect the lack of knowledge about which components should be favored over others.

Another interpretation of the concentration parameter, $\alpha$, is that it represents the strength of the prior belief in a Bayesian analysis. Consider the Dirichlet distribution in the context of the multinomial sampling distribution. Here $\theta = (\theta_1, \ldots, \theta_k)$ is the vector of probabilities on the $k$ categories of a multinomial distribution $X \sim Mn(\theta)$, with a Dirichlet prior distribution, $\theta \sim Dir(\mathbf{a})$. Given observations $x_1, \ldots, x_n$, where $n_j$ is the number of observations belonging to category $j$, the posterior pdf of $\theta$ is updated to

$$p(\theta) \propto \prod_{i=1}^{k} \theta_i^{a_i+n_i-1},$$

which means $\theta | x_1, \ldots, x_n \sim Dir(a_1+n_1, \ldots, a_k+n_k)$. This provides another intuition for the concentration parameter of the Dirichlet distribution, $\alpha$, as the prior sample size in the case of the multinomial conjugate analysis and, in general, it suggests that $\alpha$ represents the strength of the prior belief.

An important property of the Dirichlet distribution is that it aggregates; that is, if $\theta = (\theta_1, \ldots, \theta_k) \sim Dir(\mathbf{a})$, then the $(k-1)$-dimensional vector $\theta' = (\theta_1, \ldots, \theta_i + \theta_j, \ldots, \theta_k) \sim Dir(\mathbf{a}')$, where $\mathbf{a}' = (a_1, \ldots, a_i + a_j, \ldots, a_k)$. The aggregation prop-
The aggregation property can be easily used to show that the univariate marginals of the Dirichlet distribution are beta distributions. Take again \( \theta = (\theta_1, \ldots, \theta_k) \sim \text{Dir}(a_1, \ldots, a_k) \), and collapse the \( k \)-dimensional \( \theta \) vector into a 2-dimensional vector by combining the probabilities of components 1 through \( i \) together, and combining the probabilities of components \( i + 1 \) through \( k \) together, resulting in the vector \( \theta' = (\theta_1 + \ldots + \theta_i, \theta_{i+1} + \ldots + \theta_k) \). Then,

\[
\theta' \sim \text{Dir}(a'), \quad \text{with pdf} \quad p(\theta') = \frac{\Gamma(a'_1 + a'_2)}{\Gamma(a'_1)\Gamma(a'_2)} \theta'_1^{(a'_1-1)} \theta'_2^{(a'_2-1)}
\]

where \( a' = (a_1 + \ldots + a_i, a_{i+1} + \ldots + a_k) \). Because of the restriction that \( \sum_{i=1}^k \theta_i = 1 \), \( \theta'_2 = 1 - \theta'_1 \), and thus the univariate marginals of the Dirichlet distribution follow a beta distribution:

\[
\theta'_1 \sim \text{Be}(a'_1, a'_2), \quad \text{with pdf} \quad p(\theta'_1) = \frac{\Gamma(a'_1 + a'_2)}{\Gamma(a'_1)\Gamma(a'_2)} \theta'_1^{(a'_1-1)} (1 - \theta'_1)^{(a'_2-1)}.
\]

### 3.2.2 The Dirichlet process as a generalization of the Dirichlet distribution

As explained in previous sections, the Dirichlet process (DP) is a generalization of the Dirichlet distribution into infinite dimensions. It was first introduced by Ferguson (1973), and it is defined as follows: for a positive real number \( \alpha \) and a random distribution \( G_0 \) over \( \Theta \), a random distribution \( G \) follows a Dirichlet process, \( G \sim \text{DP}(\alpha, G_0) \), if for every finite partition \( A_1, \ldots, A_k \) of \( \Theta \), the vector \( (G(A_1), \ldots, G(A_k)) \) follows a Dirichlet distribution; that is

\[
(G(A_1), \ldots, G(A_k)) \sim \text{Dir}(\alpha G_0(A_1), \ldots, \alpha G_0(A_k)).
\]
Because of the aggregation property of Dirichlet distributions, the implied marginals of a Dirichlet process are Dirichlet distributions, allowing us to carry out computations and make inference on any chosen partition of \( \Theta \) using simple Dirichlet distributions, despite the complexity of the infinitely-dimensional Dirichlet process.

Much like with the Dirichlet distribution, the hyperparameters of the Dirichlet process carry helpful intuition. The base distribution \( G_0 \) is the mean of the DP, that is, \( \mathbb{E}[G(A_i)] = G_0(A_i) \). As in the Dirichlet distribution, in the DP the concentration parameter \( \alpha \) indicates how much the DP concentrates its mass around its mean \( G_0 \). The concentration parameter is inversely proportional to the variance of the DP, so the larger \( \alpha \) is, the closer to \( G_0 \) the draws from \( G \) will be. Because of this, the concentration parameter is also referred to as the precision parameter of the DP. In terms of posterior inference on \( G \), \( \alpha \) indicates the strength of the DP prior, such that when \( \alpha \to 0 \) the prior becomes non-informative and the resulting posterior becomes purely empirical.

There are several very useful representations of the Dirichlet process that have been successfully used to explain the intuition behind DPs, as well as to establish its existence and to inspire computational approaches to simulate DPs. In the following subsections, I will briefly describe the most relevant analogies, namely: the construction of DPs via stick-breaking analogy, the modified Pólya urn scheme, and the Chinese restaurant process.

### 3.2.3 Construction via stick-breaking

Sethuraman (1994) provided a constructive definition of the Dirichlet process, in which its discreteness property described in Ferguson (1973) becomes explicit. This construction is referred to as the “stick-breaking” definition of DPs due to a simple and effective analogy, beginning with an imagined stick of length 1. Draw a sample \( \pi_k^i \) from a beta distribution, which by definition will be a value between 0 and 1,
exclusive. Now break the stick at the fraction \( \pi_k \), and keep the remainder of the stick, which will represent \( (1 - \pi_k) \) of the length of the stick. Repeat this process an infinite number of times, each time starting with the remainder of the stick that was just broken. The result is an infinite number of stick fragments, where only a finite number of them will have relevant length.

The stick-breaking construction shows that a random measure distributed according to a Dirichlet process with concentration \( \alpha_0 \) and base distribution \( G_0 \), \( G \sim DP(\alpha_0, G_0) \), can be obtained via an infinite mixture of mass points drawn from \( G_0 \), where the weights of the mixture come from a beta distribution. Formally, Sethuraman (1994) showed that for the infinite sequences of random variables

\[
(\pi_k)_{k=1}^{\infty}, \quad \text{where} \quad \pi_k \sim Be(1, \alpha_0), \text{and} \\
(\phi_k)_{k=1}^{\infty}, \quad \text{where} \quad \phi_k \sim G_0,
\]

the random measure \( G \sim DP(\alpha_0, G_0) \) can be obtained by

\[
G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}, \quad \text{where} \quad \pi_k = \pi_1 \prod_{l=1}^{k-1} (1 - \pi_l).
\]

Note that \( \delta_{\phi_k} \) is an atom at \( \phi_k \) and that \( \sum_{k=1}^{\infty} \pi_k = 1 \).

Thus, \( G \) consists of an infinite mixture of atoms drawn from its base distribution \( G_0 \), where the weights for each atom are represented by the length of the stick fragments. In other words, draws from \( G \) take values \( \phi_k \) with probability \( \pi_k \).

### 3.2.4 Modified Pólya-urn scheme

Blackwell and MacQueen (1973) provided further intuition into Dirichlet processes by showing its connections with Pólya sequences. Let \( \theta_1, \theta_2, \ldots \) be a sequence of iid random variables drawn from \( G \)

\[
\theta_i \sim G, \\
G \sim DP(\alpha_0, G_0).
\]
The modified Pólya urn scheme represents the sequence \( \theta_1, \theta_2, \ldots \) of draws from the Dirichlet process. In the traditional Pólya urn scheme representing the draws from a Dirichlet distribution, described in Section 3.2.1, each time a ball is drawn from the urn, it is placed back into the urn along with a new ball of the same color. The modified Pólya urn scheme differs from the traditional one by one extra step. It proceeds as follows:

1. Draw a ball from the urn, where the balls are drawn with equal probability.

2. If the ball is black, draw a new color from \( G_0 \), paint a new ball with the new color, and place it into the urn. The proportion of black balls in the urn is \( \alpha_0 \).

3. Otherwise, place the ball back into the urn, along with a new ball of the same color.

Blackwell and MacQueen (1973) showed that the conditional distribution of a draw from this sequence has the following form:

\[
\theta_i | \theta_1, \ldots, \theta_{i-1} \sim \frac{\alpha}{i + \alpha} G_0 + \frac{1}{i + \alpha} \sum_{j=1}^{i-1} \delta_{\theta_j}(\theta_i),
\]  

(3.2)

This form of the conditional distribution illuminates several properties of Dirichlet processes: 1) It shows that draws from DP are discrete, and as such there is a positive probability that two or more values will coincide; 2) This, in turn, shows the clustering property of DPs; and 3) It displays the self-reinforcing property of DP, in which each time a particular value is observed, it is more likely to be observed again. This property is also referred to as the “rich gets richer” property of DPs.

3.2.5 The Chinese restaurant process

Aldous et al. (1985) described yet another metaphor for the DP, the Chinese restaurant process (CRP), which describes a distribution on exchangeable random partitions. Similarly to the Pólya urn scheme, the CRP refers to draws, \( \theta_i \), from a DP,
\( G \sim DP(\alpha_0, G_0) \). The metaphor goes as follows: consider a Chinese restaurant with an infinite number of round tables, each of which can seat an infinite number of customers. The first customer arrives and sits at a table. Each of the following customers that arrive sits at an occupied table with probability proportional to the number of customers already sited at that table, and at an unoccupied table with probability proportional to \( \alpha_0 \).

### 3.2.6 DP mixture models

The most common use of Dirichlet processes in the literature is as prior probabilities in infinite mixture models. The set of observed data \( \{y_1, \ldots, y_n\} \) is modeled as exchangeable observations from an infinite mixture

\[
y_i \sim G, \quad \text{and} \quad G \sim DP(\alpha, G_0),
\]

where \( G \) follows a Dirichlet process with concentration \( \alpha \) and base distribution \( G_0 \).

Being a draw from a Dirichlet process, \( G \) is guaranteed to be discrete, which can be an unrealistic constraint for some applications. One way to relax this restriction, allowing the distribution to be smooth, is to introduce an additional convolution, which leads to estimates equivalent to traditional kernel density (Müller and Quintana, 2004):

\[
y_i \sim F,
\]

\[
F(y) = \int f(y|\theta)dG(\theta), \quad (3.3)
\]

\[
G \sim DP(\alpha, G_0).
\]

Here, the data \( y \) comes from a mixture of distributions \( F \), mixing with respect to the parameters \( \theta \) of the kernel \( f \). In many clustering application, Gaussian kernels are a popular choice of \( f \).
The introduction of a latent variable $\theta$ breaks the mixture, simplifying the model and subsequent computations,

$$y_i|\theta_i \sim f(\theta_i),$$

$$\theta_i|G \sim G, \quad (3.4)$$

$$G \sim DP(\alpha, G_0).$$

Once considered a problem for representing an overly-strict constraint on distributions, the discreteness of $G$ in the context of the introduction of the latent variable $\theta$ is now a feature of the model that allows for practical implementation of MCMC. Because $G$ is discrete, values of $\theta_i$ will coincide with positive probability, such that each unique value of $\theta$ will correspond to one component in the mixture. As a result, $y_i$’s with the same value of $\theta_i$ will belong to the same component. This becomes obvious when $G$ is integrated out in the conditional prior distribution of the mixture component parameters $\theta$ (given in Blackwell and MacQueen, 1973) as shown in Equation (3.2), and restated here for reference:

$$\theta_i|\theta_1, \ldots, \theta_{i-1} \sim \frac{\alpha}{i-1+\alpha} G_0 + \frac{1}{i-1+\alpha} \sum_{j=1}^{i-1} \delta_{\theta_j}(\theta_i), \quad (3.5)$$

where $\delta_\theta(\cdot)$ is the Dirac delta distribution with mass point concentrated at $\theta$. Because the $\theta_i$ are exchangeable, Equation (3.5) can be generalized into

$$\theta_i|\theta^{(i)} \sim \frac{\alpha}{n-1+\alpha} G_0 + \frac{1}{i-1+\alpha} \sum_{j=1:j\neq i}^{n} \delta_{\theta_j}(\theta_i), \quad (3.6)$$

where $\theta^{(i)} = \{\theta_j, j \neq i\}$. As described above, the values of $\theta$ will coincide, such that for a dataset of size $n$, there will be only $k$ unique values of $\theta$, denoted as $\theta^*$, such that $k \leq n$. Hence, the conditional prior distribution of $\theta$ can be further simplified
Similarly, the predictive distribution for $\theta_{n+1}$ of a new observation $y_{n+1}$ will be

$$
\theta_{n+1}\mid \theta_1, \ldots, \theta_n \sim \frac{\alpha}{n + \alpha} G_0 + \frac{1}{n + \alpha} \sum_{i=1}^{n} \delta_{\theta_i}(\theta_{n+1})
$$

(3.8)

In practice, in most applications $k \ll n$. In fact, Antoniak (1974) shows that $k$ grows nearly linearly with $\alpha$ and logarithmically with $n$. This makes MCMC computations more manageable, despite the fact that this is an infinite mixture model. See Neal (2000) and MacEachern and Müller (1998) for surveys of MCMC inference procedures for DP Mixture Models.

One quantity of interest that arises from the MCMC simulations is the number of clusters in the mixture. In the process of sampling the values of $\theta$, the model automatically generates the number of components in the mixture. As described in Escobar and West (1995), a histogram approximation of the $k$ can then be used to address questions regarding the number of clusters and uncertainties about it, serving as an empirical posterior distribution of $k$. Additionally, histograms on the number of modes can also be obtained by evaluating the values of cluster parameters, $\theta$, over a fine grid and then using a mode-search algorithm to count the number of modes in each posterior sample obtained via the MCMC.

### 3.3 Gaussian processes

A Gaussian process (GP) is an extension of the multivariate Normal distribution into an infinite dimensional space (MacKay, 1998). Formally, a collection of random variables follows a Gaussian process if any finite subset of it have a joint Gaussian
distribution. This is also referred to as the marginalization property of Gaussian processes (Rasmussen, 2004). Based on this definition, any multivariate distribution is an example of a simple Gaussian process; but these GPs are not particularly interesting. We are interested in Gaussian processes as smoothers, particularly to perform inference and prediction on the time course data of gene expression, as will be discussed further in Chapter 4.

Much like a normal distribution is completely specified by its mean vector and a covariance matrix, a Gaussian process is completely specified by its mean and covariance functions. It follows that the GP defined a distribution over functions; as such, a sample from a GP is in itself a function. For example, consider the problem of modeling time series data from gene expression. In this case, we are given a set of \( N \) gene expression measurements, \( y = \{y_n\}_{n=1}^N \), taken at time points, \( t = \{t_n\}_{n=1}^N \), and our task is to learn the function, \( f(t) \), underlying these data. Thus, we say that \( f \) follows a Gaussian process,

\[
    f(t) \sim \mathcal{GP}(\mu(t), C(t_p, t_q)),
\]

with mean and covariance functions,

\[
    \mu(t_p) = \mathbb{E}[f(t_p)], \quad (3.10)
\]
\[
    C(t_p, t_q) = \mathbb{E}[f(t_p) - \mu(t_p)][f(t_q) - \mu(t_q)]. \quad (3.11)
\]

Often the mean function is taken to be the zero function, in which case the covariance function reduces to

\[
    C(t_p, t_q) = \mathbb{E}[f(t_p)f(t_q)]. \quad (3.12)
\]

The function \( f : \mathbb{R} \to \mathbb{R} \), provides a map from the space of time to that of gene expression measurements. Because gene expression data is very noisy, we model the measurements \( y \) as noisy observations of the function \( f \),

\[
    y = f(t) + \varepsilon, \quad (3.13)
\]

54
where $\varepsilon = \{\varepsilon_n\}_{n=1}^N$ are independent and identically distributed Gaussian noise, $\varepsilon_p \sim \mathcal{N}(0, \sigma^2)$.

Because of the marginalization property of Gaussian processes, any inference or prediction one wishes to make based on this model can be done in a very straightforward manner using well-established multivariate normal theory. For example, consider the task of predicting the expression level $y_{n+1}$ at a new time point $t_{n+1}$. For simplicity, take $\mu(\cdot) = 0$. The predictive distribution of $y_{n+1}$ is simply the conditional distribution of $y_{n+1}$ given $y, t$ and $t_{n+1}$, which follows a Gaussian distribution,

$$
(y_{n+1}|y, t, t_{n+1}) \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)
$$

(3.14)

with mean and covariance functions

$$
\hat{\mu} = K(y_{n+1}, y)[K(y, y) + \sigma^2 I]^{-1}y
$$

(3.15)

$$
\hat{\sigma}^2 = K(y_{n+1}, y_{n+1}) - K(y_{n+1}, y)[K(y, y) + \sigma^2 I]^{-1}K(y, y_{n+1})
$$

(3.16)

where $K(t, t)$ is the $N$-by-$N$ matrix obtained from evaluating the covariance function $C(\cdot, \cdot)$ at all pairwise combinations of $(t_p, t_q) \in t$. Similarly $K(t_{n+1}, t)$ is a 1-by-$N$ matrix, i.e., a row-vector, obtained from evaluating the covariance function $C(\cdot, \cdot)$ at $(t_{n+1}, t_p)$ for all $t_p \in t$.

Sampling from a Gaussian process is similarly straightforward, also relying on the marginalization property of GPs. First, create a vector, $t^* = \{t_n\}_{n=1}^{N^*}$, of length $N^*$. In order for the sample from the GP to look smooth, choose $T^*$ so that the distance between two consecutive values of $t^*_p$ is small. Then, compute the $N^*$-by-$N^*$ covariance matrix, $K^*$, by evaluating the covariance function $C(\cdot, \cdot)$ at all pairs $(t^*_p, t^*_q) \in t^*$. Finally, generate a sample vector from the normal distribution with the zero mean vector and covariance matrix $K^*$ and then plot the sampled vector against the vector $t^*$. 

55
3.3.1 Gaussian processes applications and relation to other models

Although we are primarily interested in Gaussian processes applications in regression, depending on the nature of the target variables, GPs can also be used for classification problems. When the target variable is a real number, GPs are used for regression or interpolation. For example, Brahim-Belhouari and Bermak (2004) uses GPs in time series forecasting. When the targets are categorical, GPs are used for classification. For example, Chu et al. (2005) uses Gaussian processes for predicting ordinal variables in order to identify in prostate cancer microarray gene expression data. Some well-known and widely used Gaussian processes are the Wiener and the Langevin processes, often employed to represent diffusion of Brownian particles; Kalman filters, widely used to model speech forms; and the method of kringing used in geostatistics, which is a Gaussian process regression model.

There are several methods commonly used in the machine learning community that are related to Gaussian processes. Neal (1996) brought Gaussian processes to the attention of the neural network (NN) community when he showed that the properties of a neural network with one hidden layer converge to those of a Gaussian process as the number of hidden neurons tends to infinity. Rasmussen (1996) performed extensive comparisons between Gaussian processes and neural networks, and have reported better performances of Gaussian processes over the MAP Bayesian NN for a wide range of problems. MacKay (1998) demonstrated that splines priors, used in spline smoothing, are in fact Gaussian processes. Additionally, he explains how simple parametric models, such as standard models with fixed basis functions and Gaussian distribution on the unknown parameters, are simple examples of Gaussian processes, where the predictive distribution follows a multivariate normal with mean depending linearly on the values of the targets. Rasmussen (2004) compared Gaussian processes and support vector machines (SVM) in both regression and clas-
sification domains. He showed that there is a close relationship between the MAP solution of the Gaussian process and the SVM solution, but highlighted that the fact that GPs produce outputs with clear probabilistic interpretation makes Gaussian processes more attractive.

### 3.3.2 Covariance functions

The choice of covariance function is a crucial design decision in Gaussian process models. The covariance function defines the space of functions that is supported by the Gaussian process, effectively determining what types of functions can be generated by the GP. They establish the relationship among the inputs of the GP, and thus, reflect our assumptions about the functions we are interested in. Additionally, the continuity of the functions generated by the Gaussian process is determined by the continuity properties of its covariance function: if the GP covariance function is a continuous function of its arguments, then the sample obtained from this GP are also continuous.

Gaussian process covariance functions must produce non-negative definite covariance matrices for any set of inputs \( \{t_n\}_{n=1}^N \) in its support. Any function that meets this constraint can be used as a covariance function of GPs. In the remainder of this section, we will briefly present some general classes of Gaussian process covariance functions, provide examples of commonly used functions, and describe some strategies for generating new functions.

**Stationary covariance functions**

Stationary covariance functions depend on the inputs only through their difference, that is, they are functions of \( \Delta t = t_p - t_q \). Hence, stationary covariance functions are said to be translation invariant. Stationary covariance functions are often described in terms of their Fourier transform, which is also known as the power spectrum of
the GP (MacKay, 1998)

Stationary covariance functions that only depend on the magnitude of the distance between the input values, $|t_p - t_q|$, are called homogeneous or isotropic covariance functions (Seeger, 2004). One example of such function is squared exponential function,

$$C(t_p, t_q) = \exp\left(-\frac{(t_p - t_q)^2}{2l^2}\right),$$

(3.17)

where $l$ is the length scale variable. Large values of $l$ lead to the the GP outputs, $y$, essentially being constant functions of the inputs $t$. This covariance function is infinitely differentiable, leading to very smooth sampled functions (Rasmussen, 2004). The squared exponential function has been widely used in GP applications.

Periodic functions are yet another class of commonly used GP covariance functions, being an appealing choice for modeling seasonal data. MacKay (1998) provides the following example,

$$C(t, t') = \exp\left[-\frac{1}{2} \sum_i \sin\left(\frac{\pi}{\lambda_i} (t'_i - t_i)\right)^2\right],$$

(3.18)

where the inputs $t$ and $t'$ are vectors, and $\lambda_i$ are known periods.

Other types of stationary covariance functions are the Matern class of covariance functions, the rational quadratic covariance function and polynomial covariance functions. (Rasmussen, 2004) provides a number of other examples of covariance functions.

Non-stationary covariance functions

In contrast to stationary covariance functions, non-stationary covariance functions depend on the actual values of the inputs $t_p$ and $t_q$. The simplest type of such covariance function results from linear regression $y(t) = \sum_i w_i t_i + c$, where $w_i$ and
c follow normal distributions with zero mean and standard deviations $\sigma_w$ and $\sigma_c$ (MacKay, 1998). The resulting covariance function takes the following form

$$ C(t, t') = \sum_i \sigma_w^2 i t_i + \sigma_c^2. \quad (3.19) $$

**Constructing new covariance functions**

There are a number of ways to obtain new GP covariance functions based on existing covariance functions. Below we describe three of the most straightforward ways to combine existing covariance functions into new ones:

1. **Addition.** If $C_1(t, t')$ and $C_2(t, t')$ are covariance functions, then $D(t, t') = C_1(t, t') + C_2(t, t')$ is also a covariance function.

2. **Product.** If $C_1(t, t')$ and $C_2(t, t')$ are covariance functions, then $D(t, t') = C_1(t, t') \cdot C_2(t, t')$ is also a covariance function. This is applicable in cases in which the functions $C_1$ and $C_2$ are over the same or different spaces.

3. **Vertical scaling.** If $C_1(t, t')$ is a covariance function and $a(t)$ is some deterministic function, the $D = a(t)C_1(t, t')a(t')$ is also a covariance function.

Other ways to generate covariance functions include convolutions and O’Hagan’s model, both of which are reviewed in Rasmussen (2004) and citeMackay1998. In Section 4.3.2, we describe the Langevin process, in which we use a stochastic dynamical system to generate the covariance function.
Comparing individual gene trajectories

4.1 Introduction

Dendritic cells play a central role in the modulation of the immune response. They constantly survey their environment, and once they detect an infection, they go through a maturation process and release signals to activate the appropriate branch of the immune response, which will target specifically the invading pathogen. One of the primary ways in which DCs detect pathogens is via activation through different TLRs. Although there is much similarity and overlap among the major TLR signaling pathways, each TLR activation ultimately leads to mature DCs with slightly different receptor expression and cytokine releasing patterns, tilting the immune response in different directions. Therefore, each TLR must also activate unique elements in their signaling pathways and/or activate unique sets of transcription factors and downstream genes. Indeed, there are a number of such examples that have been unraveled. For example, we know that TLR3 is the only TLR to not use the MyD88-dependent pathway, discussed in Chapter 2, while we know that among the currently known TLRs, TLR4 is the only TLR that uses all four adaptor molecules: MyD88,
TIRAP, TRIF and TRAM.

Ultimately the difference in phenotype of DCs activated via the different TLRs results from a combination of factors, such as the subtype of DC, the signaling molecules available in DC intracellular compartments and what transcription factors are activated. We believe that some, if not most, of the phenotypic differences in DCs stimulated via different TLRs must be due to differences in gene expression activation patterns induced by these TLRs. Over the past decade we have learned a great deal about the main players in driving each one of these responses; however, there is even more that we do not know about. Unraveling what genes present similar or unique transcription expression pattern under different stimulation will be essential to our understanding of how dendritic cells modulate the immune response.

The Kepler laboratory has produced a unique, large, and high-quality microarray gene expression dataset time series by stimulating murine dendritic cells with either LPS (a major TLR4 ligand) or CpG (a major TLR9 ligand). The time series consists of 8 time points over the course of 46 hours from the time of stimulation. This dataset provides us a unique opportunity to investigate the commonalities and uniquenesses of the transcription expression patterns activated by either TLR. In the current and next chapters, we explore these data. The goal of the analysis presented in the current chapter is two-fold. First, we want to analyze each experiment, LPS and CpG, separately and identify which genes change their temporal transcriptional program in response to the stimulus. Second, we want to analyze the two data groups simultaneously and be able to determine, for each gene, whether it is differentially expressed in response to one (and if so, which one), both, or neither of the two stimuli. We emphasize that we are primarily interested in comparing the treatment effects between samples, rather than the expression levels per se.

Here we have developed methodology to perform this type of analysis using a special family of Gaussian processes, called Langevin processes (LP). As described
in Chapter 3, Gaussian processes provide a flexible and powerful approach to modeling time series, allowing for the comparison of whole trajectories without regard to specific parametric forms for the functional relationship between time and gene expression.

A key step in Gaussian process modeling is choosing a covariance matrix that reflects prior knowledge and assumptions about the data structure. For our temporal gene expression data, there are three major aspects of the data that are relevant to the choice of covariance function. First, gene expression data measured by microarrays are typically noisy. Second, in a time series experiment, we expect the correlation between consecutive time points to decrease over time. Third, we expect the temporal gene expression data to be non-stationary.

The remainder of this chapter is organized as follows. First, in Section 4.2, we describe the dendritic cell gene expression dataset that we have produced in our lab and that will be the focus of this and the next chapter. In Section 4.3, we describe the basic Langevin process and provide further insight into its covariance function. Then, in Section 4.4, we describe the Langevin process in the context of gene expression time series. We first present a model to analyze a single time series, developing it further to be able to compare two time series (control and treatment) and finally to compare all data available for a given gene simultaneously. In Section 4.5, we demonstrate the utility of modeling gene expression with Langevin processes by describing flatness tests, and then applying them to the dendritic cell dataset. In Section 4.7 we present the results of comparing all data available for a given gene by applying this approach to the dendritic cell dataset. We end this chapter with a brief discussion in Section 4.9 about the relevance of the methodology developed here and the results obtained using this model.
4.2 The dendritic cell temporal gene expression dataset

The dendritic cell temporal gene expression dataset consists of two separate experiments, each with two complete time series of microarray mRNA measurements from murine splenic dendritic cells. Each time series is composed of measurements taken at seven or eight time points over the course of 46 hours. In the first experiment, one of the time series consists of measurements obtained from unstimulated dendritic cells (control time series), and the other time series consists of measurements obtained from dendritic cells stimulated with LPS. Similarly, in the second experiment, there is a control time series and another time series obtained from dendritic cells stimulated with CpG. All four time series were obtained in duplicates. In both experiments, mRNA was collected at the following time points (in hours post stimulation): 0.5, 1.5, 3.25, 6.5, 12.5, 24, and 45.5. In the control time series, additional measurements were collected at zero hours.

The rationale used in the design of these time series was the following. Given a fixed time interval, $\Delta t$, the amount of information contained in $\Delta t$ early in the time series is greater than the information contained in the same $\Delta t$ at a later time in this series. Specifically, we assume that the information decreases geometrically as time progresses. Thus, in order to obtain time intervals that represent approximately the same amount of information, we have designed our time series with the geometrically increasing time intervals described above.

Because we consider this time transformation to lead to time intervals that contain similar amounts of information, here we use the time indexes instead of the time in hours both as input to the Langevin process as well as for plotting purposes. Thus, for the rest of the thesis, unless we explicitly refer to the time in hours, we will refer to time indexes as time. For clarity, Table 4.1 shows the mapping of time index to time in hours that reflects our time transformation. Further information about
the data collection process and the purity of dendritic cells obtained is described in Appendix A.

<table>
<thead>
<tr>
<th>Time index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hour</td>
<td>0</td>
<td>0.5</td>
<td>1.5</td>
<td>3.25</td>
<td>6.5</td>
<td>12.5</td>
<td>24</td>
<td>45.5</td>
</tr>
</tbody>
</table>

4.3 The Langevin process

4.3.1 Brownian motion and the Langevin equations

Brownian motion describes the movement of small particles immersed in fluid. It is named after the botanist Robert Brown, who observed and described the movement of pollen grains, and later that of dust particles, suspended in water under a microscope. The fundamental equation describing a Brownian motion is the Langevin equation. Einstein described the position $x(t)$ and velocity $v(t)$ of a particle with mass $m$ as a function of time,

$$\frac{dx(t)}{dt} = v(t), \quad (4.1)$$
$$\frac{dv(t)}{dt} = -\frac{\phi}{m} v + \frac{1}{m} \xi(t), \quad (4.2)$$

where $\phi = 6\pi \eta a$ is Stoke’s coefficient of friction, with $a$ being the radius of the particle and $\eta$ being the viscosity of the fluid in which the particle is moving. $\xi(t)$ is a stochastic process that represents the effect of background noise due to the fluid interactions with the particle. $\xi$ has zero mean and is memoryless, i.e., given the current state, future events are independent of past history. In fact, $\xi$ is a Wiener process, $\xi dt = dW(t)$, which is a type of Gaussian process. In its construction, the Langevin equation describes the movement of a particle in fluid as a balance between
two type of forces: random forces and friction. It is driven by random forces (via the Wiener process) and dampened by friction forces.

The theory of Brownian motion and its description by the Langevin equations have found broad application in areas as diverse as biochemistry, market analysis, and robotics, having been extended to situations in which the floating object is not necessarily a real particle. In the next section, we describe our own application of it in the analysis of gene expression data. For overviews of the theory and its impact in science, see Lemons (1997); Hanggi and Marchesoni (2005).

4.3.2 Basic Langevin process

The Langevin process arises from the Langevin equations and is an example of a Gaussian process; this can be seen by the fact that the Wiener process driving the Langevin process is in itself Gaussian. We are interested in a more general view of the Langevin process that is not restricted to particle motion. In this case, parameters such as particle mass and fluid viscosity may not have a direct interpretation. Hence, here we show the Langevin equations in a reparametrized form to reflect this general view,

\[
\begin{align*}
\frac{dx}{dt} &= v dt, \\
\frac{dv}{dt} &= -\gamma v dt + \zeta \sqrt{\gamma} dW,
\end{align*}
\]

where \( x \) is the state variable of interest, and \( dW \) is a differential Wiener process. \( \gamma \) and \( \zeta \) are constant parameters and have useful interpretations. \( \gamma \) determines how fast the correlation among time points drops to zero, providing an indication of how volatile the trajectories sampled from the LP are. The larger \( \gamma \), the more sharply curved the trajectory. In other words, as \( \gamma \to \infty \), the values of the curve become independent. \( \zeta^2 \) is the mean square slope of the trajectory defined by this process; as such, \( \zeta \) serves as a proxy for one’s subjective judgment of how much \( x \) varies.
over time. For large $\zeta$ and small $\gamma$, the trajectory may be a nearly straight non-flat line segment. For large $\gamma$ and small $\zeta$, the trajectory may have low-amplitude high-frequency fluctuations about its steady state flat value.

The mean function of the Langevin process is

$$\mu(t) = \mathbb{E}[x(t)] = \mathbb{E}[x(0)] + \mathbb{E}[v(0)] \frac{1}{\gamma} (1 - e^{-\gamma t}), \quad (4.5)$$

where $\mathbb{E}[-]$ denotes expectation over the Wiener process. Here, we take the expected value of the initial velocity to be zero, such that

$$\mu(t) = \mathbb{E}[x(0)]. \quad (4.6)$$

The covariance function of the Langevin process is

$$C(t, t + \tau) = \mathbb{E}[(x(t) - \mu(t))(x(t + \tau) - \mu(t + \tau))]$$

$$= \frac{\zeta^2}{\gamma^2} \left\{ \gamma t - (1 - e^{-\gamma t}) [1 + \frac{1}{2} (1 - e^{-\gamma t}) e^{-\gamma \tau}] \right\}. \quad (4.7)$$

Note that if $\zeta = 0$, then $x = x(0)$, i.e., there is no force driving the dislocation of $x$; as such, $x$ remains constant.

4.4 Modeling temporal gene expression trajectories with the Langevin process

4.4.1 Interpretation of gene expression through the Langevin process

The level of mRNA of a given gene in a cell is a function of several cellular and molecular processes, such as transcription and mRNA degradation. All of these processes are stochastic phenomena; consequently, gene expression is also a stochastic phenomenon (Kepler and Elston, 2001; Kaern et al., 2005; Lipniacki et al., 2006). Hence, it is natural to model gene expression with stochastic processes. The Langevin process is particularly well-suited for this problem because, in addition to providing
an appealing analogy to the process of gene expression, the implicit assumptions that this model makes about the data correspond to our prior knowledge and assumptions about the gene expression process and the microarray-generated data.

The terms in the Langevin equation, shown in Equation (4.3) and Equation (4.4), can be interpreted to correspond to microarray data in the following manner: \( x \) represents the level of mRNA measured in a cell via microarray experiments. The changes in \( x \) over time are driven by a Wiener process, \( dW \). Finally, the factors responsible for decreasing the levels of mRNA in the cell, such as mRNA degradation, are encompassed in \(-\gamma vdt\).

As stated earlier, we know \( a \ priori \) that the gene expression process is a non-stationary one. Likewise, the Langevin process is a non-stationary process, i.e., \( C(t_p, t_{p+\tau}) \neq C(t_q, t_{q+\tau}) \) for \( p \neq q \). The structure and properties of covariance matrices produced by the covariance function of the LP are also in agreement with our prior knowledge about the data. For instance, we mentioned earlier that we expect the correlation among expression levels increases over time. As shown in Figure 4.1, this is in strong agreement with the covariance function of the Langevin process \(- \rho(t_p, t_{p+\tau}) > \rho(t_q, t_{q+\tau}) \) for \( p > q \), where \( \rho(t, t') \) is the correlation between time points \( t \) and \( t' \), \( \rho(t, t') = \frac{C(t,t')}{\sqrt{C(t,t)}\sqrt{C(t',t')}} \). Thus, the Langevin process is a natural and appealing choice for modeling microarray gene expression data.

4.4.2 Notation

Our dataset consists of two experiments, each of which comprises two time series: a control time series and a treatment time series. The controls in both experiments are effectively biological replicates, i.e., they are separate biological samples that have undergone the same experimental process. The treatments are stimulation with LPS and CpG. For each time point in each time series of each experiment, we have two replicates. Hence, each mRNA measurement is identified by five indices, \( y_{ijktr} \). The
Figure 4.1: Langevin process correlations, \( \rho \), between time points.

The first index, \( i \), denotes the probe set ID, where \( i \in \{1, \ldots, N\} \), and \( N = 45101 \) is the total number of probe sets in the Affymetrix microarray chip. The second index, \( j \), identifies the experiment, where \( j \in \{0, 1\} \), with \( j = 0 \) denoting the experiment for the LPS treatment, and \( j = 1 \) the experiment for the CpG treatment. The third index, \( k \), distinguishes the control time series (\( k = 0 \)) from the treatment time series (\( k = 1 \)). The fourth index, \( t \), denotes time index, with \( t \in \{0, 1, \ldots, 7\} \). Finally, the fifth index, \( r \), denotes replicate, with \( r \in \{0, 1\} \).

For brevity, when no confusion occurs we suppress some of the indices of \( y \) to represent collections of measurements. For example, \( y_i \equiv y_{i...} \) denotes the vector of all observations for the four time series available for a given probe set ID, \( y_i = \{y_{ijktr} : j = 0, 1; k = 0, 1; t = 0, 1, \ldots, 7; r = 0, 1\} \). Similarly, \( y_{ij} \equiv y_{ij...} \) denotes the
vector containing the control and treatment time series from a single experiment,
\( y_{ij} = \{ y_{ijktr} : k = 0, 1; t = 0, 1, \ldots, 7; r = 0, 1 \} \), and \( y_{ijk} \equiv y_{ijk} \) identifies a single
time series for a given probe set ID, \( y_{ijk} = \{ y_{ijktr} : t = 0, 1, \ldots, 7; r = 0, 1 \} \).

4.4.3 Statistical model

We first consider the temporal expression of an individual gene in response to one
experimental stimulation. For simplicity, since in this case we are only concerned
with a single time series of a single gene, we omit all indices, and let \( y(t) \) be the
mRNA measurement at time \( t \) and \( y \) be the vector of all measurements \( y(t) \). Then,
the \( y(t) \) is assumed to be a noisy observation from an underlying trajectory \( x \) at time
\( t \),

\[
y(t) = x(t) + \varepsilon_t
\]

where \( \varepsilon_t \) are independent, identically distributed Gaussian measurement errors, \( \varepsilon_t \sim \mathcal{N}(0, \sigma^2) \). The underlying trajectory \( x(t) \) follows a Gaussian process, \( x \sim \mathcal{GP}(\mu, C) \),
with mean, \( \mu(t) \), and covariance, \( C(t, t + \tau) \), functions as defined in Equation (4.5)
and Equation (4.7), respectively. Recall that, by definition, any finite set of random
variables from a Gaussian process will necessarily have a joint Gaussian distribution.
In practice, this means that by integrating over all trajectories (given the parameters
\( \gamma, \zeta, \) and \( \sigma \), and the initial conditions \( x_0 \) and \( v_0 \)), the resulting distribution of \( y \) will
be a linear combination of normal distributions, which is in itself a multivariate
Gaussian distribution with mean vector, \( \mu \), whose components are

\[
\mu_p \equiv \mathbb{E}[y_{tp}] = \mu(t_p)
\]

and covariance matrix, \( \Sigma \), with components,

\[
\Sigma_{pq} \equiv C(t_p, t_q) + \sigma^2 \delta_{pq}
\]
where $\delta_i$ is Kronecker’s delta,

$$
\delta_{pq} = \begin{cases} 
1 & \text{if } p = q, \\
0 & \text{if } p \neq q. 
\end{cases}
$$

Note that although the covariance function, $C(t_p, t_q)$, of the Langevin process produces singular matrices, the covariance matrix shown in Equation (4.10) conforms to the constraint that the covariance function of a GP must be one that produces a non-negative definite matrix for any set of points $\{t_n\}_{n=1}^N$ (as explained in Section 3.3.2).

In order to assess the goodness-of-fit resulting from the Langevin process, we examine the residuals. We compute the residuals using the trajectory with the maximum posterior probability given the data, the parameters, and the initial conditions,

$$
\mathcal{T}(t) = \mu(t) + c(t)\Sigma^{-1}(y - \mu),
$$

where $c(t)$ is a row vector obtained by evaluating $C(t, t_i)$ for all sampling times $t_i$.

### 4.4.4 Parameters priors

**Prior on $\sigma^2$**

$\sigma^2$ represents the experimental error variance; as such, it must be non-negative. In fact, we expect that $\sigma^2 > 0$ for most of the data. In addition, as $\sigma^2 \to 0$, $\Sigma$ (whose individual components are shown in Equation (4.10)) becomes singular. For these two reasons, $\sigma^2$ must be bound away from zero. Thus, we define the following prior distribution for $\sigma^2$,

$$
\sigma^2 \sim Ga(2, 1),
$$

whose density function is shown in Figure 4.2A.

**Prior on $\zeta$**

In choosing a prior distribution for $\zeta$, we take three facts into consideration:
1. As described earlier, $\zeta^2$ is the mean squared slope of functions sampled from the LP; as such, $\zeta$ can only take non-negative values.

2. For any given stimulus, we expect that only a subset of the genes will present any changes in their transcriptional response, with most genes being unresponsive to the stimulus. This translates into most genes having $\zeta$ close to zero, and a small subset of them have $\zeta > 0$.

3. We know empirically that the log-transformed mRNA measurements range between zero and 15, approximately. Because the time intervals considered here are all of one unit, we expect the slope of curves to vary between zero and 15, approximately.

Thus, we have chosen to use a gamma prior on $\zeta$

$$\zeta \sim \mathcal{Ga}(1, 10), \tag{4.13}$$

which is equivalent to the exponential distribution with parameter 0.1. This prior, shown in Figure 4.2B, suits well the requirements for a prior distribution for $\zeta$, allowing for large values, while favoring smaller values of $\zeta$.

**Figure 4.2**: Prior probability density functions for the Langevin process hyperparameters: (A) $\sigma^2 \sim \mathcal{Ga}(2, 1)$, (B) $\zeta \sim \mathcal{Ga}(1, 10)$, and (C) $\gamma \sim \mathcal{Ga}(1, 1)$.
Prior on $\gamma$

The primary \textit{a priori} restriction on $\gamma$ is that it cannot be allowed to take very large values. As mentioned in Section 4.3.2, as gamma approaches infinity, the time points become independent. In this case time is considered discrete, and thus is a different model than the continuous time model we are considering here. We restrict the support of gamma by using the prior distribution $\gamma \sim Ga(1, 1)$.

4.4.5 Additive Langevin processes

Having established the model for a single time course, we can now take a step further and build upon this. The time trajectories of a given gene under any of the four time courses are related, and as such should be modeled together. We use an additive model, where the factors contributing to the observed trajectory over time are treated as separate processes. For example, consider the expression of a given gene under stimulation with LPS. If this gene does not respond to the stimulation, then its trajectory should be the same as that for the control, differing from it only by the Gaussian noise. If the gene responds to the stimulation, then its trajectory will be the result of the underlying trajectory (that of the control) plus a treatment effect. Additionally, the information from any one of the four time series is certainly correlated with any of the other time series. Thus, it makes sense to model all four time series simultaneously. Without loss of generality, here we use the average of the replicates, such that $y_{ijkt} = \frac{1}{2}(y_{ijkt0} + y_{ijkt1})$. The vector $y_i$ containing the four time series for gene $i$ is now written,

$$y_i = \{ y_{i000}, \ldots, y_{i00T_{00}}, y_{i011}, \ldots, y_{i01T_{01}}, y_{i100}, \ldots, y_{i10T_{10}}, y_{i111}, \ldots, y_{i11T_{11}} \},$$

(4.14)

where $T_{jk}$ is the last time point measured for experiment $j$ and time series $k$. Note that at time point zero, the stimulus, LPS or CpG, has not been added to the cell.
culture yet. For this reason, for both control time series the first measurement is taken at time point 0, while for both treatment time series the first measurement is taken at time point 1. Then, the observed mRNA measurement of a gene \( i \) is

\[
y_{ijkt} = \xi(t) + \Delta \xi_j(t) + \delta_{k1}(\tau(t) + \Delta \tau_j(t)) + \varepsilon_{ijkt},
\]

(4.15)

where \( \xi, \Delta \xi_j, \tau_j \) and \( \Delta \tau_j \) are independent Langevin processes.\(^1\) \( \zeta + \Delta \xi_j \) correspond to the baseline trajectory of experiment \( j \), whereas \( \tau + \Delta \tau_j \) correspond to the treatment effect in experiment \( j \). As before \( \varepsilon_{ijkt} \sim \mathcal{N}(0, \sigma^2) \) are independent and identically distributed Gaussian noise, and \( \delta_{.} \) is Kronecker’s delta.

The model obeys the following constraints,

\[
\Delta \xi \equiv \Delta \xi_0 = -\Delta \xi_1, \quad (4.16)
\]

\[
\Delta \tau \equiv \Delta \tau_0 = -\Delta \tau_1. \quad (4.17)
\]

The Langevin processes \( \xi, \Delta \xi_j, \tau_j \) and \( \Delta \tau_j \) all have zero mean function. Take the covariance function of a Langevin process \( g(t) \) to be

\[
C_g(t, t') \equiv \mathbb{E}[g(t)g(t')].
\]

Then by applying the constraints above, we obtain the four covariance functions,

\[
C_{\xi}(t, t') \equiv \mathbb{E}[\xi(t)\xi(t')], \quad (4.18)
\]

\[
C_{\Delta \xi}(t, t') \equiv \mathbb{E}[\Delta \xi_j(t)\Delta \xi_j(t')], \quad (4.19)
\]

\[
C_{\tau}(t, t') \equiv \mathbb{E}[\tau_j(t)\tau_j(t')], \quad (4.20)
\]

\[
C_{\Delta \tau}(t, t') \equiv \mathbb{E}[\Delta \tau_j(t)\Delta \tau_j(t')]. \quad (4.21)
\]

As per Equation (4.15), the vector \( y_i \) follows a linear combination of four Gaussian processes, and some Gaussian noise. Consequently, \( y_i \) is a multivariate normal

\(^1\) We take the liberty to use time and time index exchangeably because all of our times are in fact integer values.
random variable, with mean zero, and covariance matrix \( \Sigma \) defined by the following covariance function,

\[
K(y_{ijkt}, y_{ij'k't'}) = C(y_{ijkt}, y_{ij'k't'}) + \delta_{tt'} \sigma^2,
\]

(4.22)

where,

\[
C(y_{ijkt}, y_{ij'k't'}) = C_\xi(t, t') + (-1)^{j-j'} C_\Delta_\xi(t, t') + \delta_{k1} \delta_{k'1} C_\tau(t, t') + \delta_{tt'} \delta_{k1} \delta_{k'1} C_\Delta_\tau(t, t').
\]

(4.23)

Notice that, because the four GPs are independent, all cross-terms in the covariance function vanish. Let \( \Sigma_g \) be the covariance matrix obtained by evaluating the covariance function \( C_g(t, t') \) at all pertinent \( (t, t') \) pairs. Then, the covariance matrix of \( y_i \) is

\[
\Sigma + \sigma^2 I,
\]

(4.24)

where \( \Sigma \) can be written in terms of blocks of smaller covariance matrices,

\[
\Sigma = \begin{pmatrix}
\Sigma_\xi + \Sigma_\Delta_\xi & \Sigma_\xi - \Sigma_\Delta_\xi & \Sigma_\xi + \Sigma_\Delta_\xi & \Sigma_\xi - \Sigma_\Delta_\xi \\
\Sigma_\xi - \Sigma_\Delta_\xi & \Sigma_\xi + \Sigma_\Delta_\xi & \Sigma_\xi - \Sigma_\Delta_\xi & \Sigma_\xi + \Sigma_\Delta_\xi \\
\Sigma_\xi + \Sigma_\Delta_\xi & \Sigma_\xi - \Sigma_\Delta_\xi & \Sigma_\xi + \Sigma_\Delta_\xi + \Sigma_\tau + \Sigma_\Delta_\tau & \Sigma_\xi - \Sigma_\Delta_\xi - \Sigma_\tau + \Sigma_\Delta_\tau \\
\Sigma_\xi - \Sigma_\Delta_\xi & \Sigma_\xi + \Sigma_\Delta_\xi & \Sigma_\xi - \Sigma_\Delta_\xi + \Sigma_\tau - \Sigma_\Delta_\tau & \Sigma_\xi + \Sigma_\Delta_\xi + \Sigma_\tau + \Sigma_\Delta_\tau
\end{pmatrix}
\]

(4.25)

Similarly to the single time series model described above, here we use the maximum a posteriori trajectory,

\[
\mathcal{T}(t) = \hat{x}_0 + c(t)(\Sigma + \sigma^2 I)^{-1}(y_i - \hat{x}_0),
\]

(4.26)

where \( c(t) \) is a row vector obtained by evaluating \( C(t, t_i) \), shown in Equation (4.23), for all sampling times \( t_i \). \( \hat{x}_0 \) is the maximum likelihood estimator (MLE) for the initial conditions of both experiments,

\[
\hat{x}_0 = \hat{x}_{00} l_0 + \hat{x}_{01} l_1,
\]

(4.27)
where, \( l_0 = \{1, 0\} T_{00}, \{0, 1\} T_{01}, \{0, 0\} T_{00}, \{0, 1\} T_{10} \) and \( l_1 = \{0, 0\} T_{00}, \{1, 0\} T_{01}, \{0, 1\} T_{10}, \{1, 1\} T_{11} \) are vectors of zeros and ones that correspond to the items in \( y_i \), shown in Equation (4.14). \( \hat{x}_{0j} \) is the MLE of the initial condition for experiment \( j \), obtained by solving the system of linear equations

\[
1_0^T \Sigma^{-1} 1_0 \hat{x}_{00} + 1_1^T \Sigma^{-1} 1_0 \hat{x}_{01} = y_i^T \Sigma^{-1} 1_0 \tag{4.28}
\]

\[
1_0^T \Sigma^{-1} 1_0 \hat{x}_{00} + 1_1^T \Sigma^{-1} 1_1 \hat{x}_{01} = y_i^T \Sigma^{-1} 1_1 \tag{4.29}
\]

Note that this implies that the initial conditions for both the control and the treatment time series in a given experiment are the same. The reason is that at time point zero both control and treatment groups have received the same treatment, and are expected to be the same except for the background noise.

### 4.5 Test of flatness

One of the basic questions we typically want to address in the analysis of the transcriptional response to changes in environmental conditions is whether the abundance of a particular transcript changes over time subsequent to the perturbation. The observed temporal mRNA expression pattern of genes in both control and treatment experiments are influenced by several stimuli in the experimental procedure, both intentional and unintentional. First and most obvious, changes in gene expression are caused by the activation of transcription factors directly by the signaling cascade resulting from the treatment stimulus. That, however, is not the only stimulus influencing the changes in gene expression. The steps of the experimental procedure, such as plating of cells, interaction of cells with the medium, and handling of cells to extract the mRNA are all stimuli to which the cells react by regulating the expression of genes. Because of this, the time course of the control experiment is not always as flat as it would ideally be, and the observed trajectory of the treatment group is not purely a consequence of the treatment effect; instead it is the result of a complex
interaction of all of these stimuli.

The additive Langevin process model described in Equation (4.15) accounts for this by separating a given temporal trajectory into two components, treatment \((\tau + \Delta \tau_j)\) and baseline \((\zeta + \Delta \zeta_j)\), and determining whether the contribution of one or both of these components are null or not. In this context, asking whether or not a gene’s trajectory is flat corresponds to testing whether or not a given component of the model is null. Using this model, we are able to test and identify the contributions from the baseline and the treatment effects to the observed temporal expression pattern of the gene. Here we first describe our model selection procedure, and subsequently demonstrate some of the tests about the inferred trajectory of a gene that can be done with the Langevin process model.

4.5.1 Model selection

We use the minimum description length (MDL) (Rissanen, 1978) approach to model selection. The MDL principle is based on the argument that any regularity in the data can be used to compress it, and equates learning with finding regularity in the data. From a statistical viewpoint, this corresponds to finding one density within a family of densities that best describes the data.

The description length (DL) for a given model, \(M\), has the form

\[
DL_M = - \log p(y|\hat{\theta}) + \frac{k}{2} \log n, \tag{4.30}
\]

where \(k\) is the number of parameters in the model, and \(n\) is the total number of data points. The term proportional to \(k\) is referred to as the complexity term. \(\hat{\theta}\) is the set of parameters that minimize the penalized likelihood, \(p(\cdot|\cdot)\), which is proportional to the posterior density of the data,

\[
- \log p(y|\hat{\theta}) = - \log \mathcal{L}(y; \hat{\theta}) - \log p(\hat{\theta}), \tag{4.31}
\]

76
where \( \log \mathcal{L}(y; \hat{\theta}) \) is the log likelihood of the data, and \( p(\hat{\theta}) \) is the prior of the parameters \( \hat{\theta} \). In our model, \( \theta = (\gamma, \zeta, \sigma^2) \).

Model selection via the MDL principle entails finding \( \theta \) within a parametric distribution family that leads to the smallest value of DL. This approach automatically protects against overfitting, by balancing goodness-of-fit and model complexity (Grünwald, 2005). Furthermore, it has connections to both frequentist (via the maximum likelihood principle) and Bayesian approaches to statistics (Hansen and Yu, 2001). In fact, Equation (4.30) is the negative of the Bayesian information criterion (BIC) utilized in Bayesian model comparison. Hence, under regularity conditions, and taking into consideration that the MDL described in Equation (4.30) is obtained via a Taylor series expansion that ignores \( O(1) \)-terms, the MDL and BIC are effectively equivalent model selection criteria.

### 4.5.2 Tests of the baseline effect

The baseline effect accounts for the variations in gene expression in the control group resulting from the cells’ transcriptional response to stimulation due to parts of the experimental process itself, such as the handling of cells in different stages of the experimental procedures and cell interactions with different media. The Langevin process allows us to test whether or not the baseline effect is null. To do so, we consider the time series of a single control group alone, \( y_{ij0} \). Because we are considering a single gene, experimentand only the control time series, the only relevant index here is the time index \( t \). Thus, for convenience, here we take \( y = \{y_t : t = 1, \ldots, T\} \), and compare the DL of the posterior mean trajectory under the null model and the full model.

The mean posterior trajectory under the baseline effect model is

\[
\mathcal{T}_{+}^{\text{baseline}}(t) = \hat{x}_0 + c_\xi(t)(\Sigma_\xi + \sigma^2 I)^{-1}(y - \hat{x}_0 1),
\]

(4.32)
where $\Sigma_\xi$ has parameters $\gamma_\xi$ and $\zeta_\xi$, and $c_\xi(t)$ is a row vector obtained by evaluating $C_\xi(t, t')$ for $t' \in (0, 1, \cdots, T)$.

The mean posterior trajectory under the null baseline effect model is

$$\mathcal{T}_\text{baseline}(t) = \hat{x}_0,$$

which is the same as Equation (4.32) when $\zeta_\xi = 0$. Thus, the test of whether the baseline effect is null is done by observation of the parameter $\zeta_\xi$ in Equation (4.32). If the model with $\zeta_\xi = 0$ leads to the minimum description length between the two models, then we conclude that the baseline effect is null, that is, the control temporal trajectory is estimated to be flat. Otherwise, we conclude that there is in fact a baseline effect. Table 4.2 summarizes the results from the flatness tests applied to the dendritic cell gene expression data.

Table 4.2: Summary of tests for control effects: Total number of probes determined to have a non-flat or flat control trajectory in the LPS and the CpG experiments.

<table>
<thead>
<tr>
<th></th>
<th>Flat</th>
<th>Not Flat</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPS</td>
<td>41478</td>
<td>3623</td>
</tr>
<tr>
<td>CpG</td>
<td>39987</td>
<td>5114</td>
</tr>
</tbody>
</table>

4.5.3 Tests of the treatment effect

A much more interesting test than that of the baseline effect is the test of treatment effect, which accounts for changes in gene expression due to the treatment stimulus itself. This test addresses the question of whether or not the treatment time series differs from the corresponding control time series. We now consider the vector $y_{ij}$ containing the control and treatment time series corresponding to gene $i$ and experiment $j$. Similar to the baseline effect test, we compare the DL of the posterior mean trajectory under the null treatment effect model, $\mathcal{T}_-^{\text{treat}}$, and the treatment effect model, $\mathcal{T}_+^{\text{treat}}$. Since we are considering a single experiment and a single gene, we can
for convenience drop the two indices \((ij)\). Thus the model for \(y_{kt}\) in Equation (4.15) reduces to

\[
y_{kt} = \xi(t) + \delta_{k1}\tau(t) + \varepsilon_{kt}.
\]  

(4.34)

Hence, \(y_k\) follows a Gaussian process with covariance function

\[
K_{\text{treat}}^{+}(y_{kt}, y_{k' t'}) = C_{\text{treat}}^{+}(y_{kt}, y_{k' t'}) + \delta_{t't}\sigma^2,
\]  

(4.35)

where

\[
C_{\text{treat}}^{+}(y_{kt}, y_{k' t'}) = C_{\xi}(t, t') + \delta_{k1}\delta_{k'1}C_{\tau}(t, t').
\]  

(4.36)

Similar to Equation (4.25), the covariance matrix of \(y_k\) under this model is

\[
\Sigma_{\text{treat}}^{+} + \sigma^2 I,
\]  

(4.37)

where \(\Sigma_{\text{treat}}^{+}\) can be written in terms of blocks of smaller covariance matrices,

\[
\Sigma_{\text{treat}}^{+} = \begin{pmatrix}
\Sigma_{\xi} & \Sigma_{\xi} \\
\Sigma_{\xi} & \Sigma_{\xi} + \Sigma_{\tau}
\end{pmatrix}.
\]  

(4.38)

The test of whether the treatment effect is null is done by observation of the parameter \(\zeta_{\tau}\). When \(\zeta_{\tau} \neq 0\), there is a treatment effect, such that the mean posterior trajectory is

\[
\mathcal{F}_{\text{treat}}^{+}(t) = \hat{x}_0 + c_{\text{treat}}^{+}(t)(\Sigma_{\text{treat}}^{+} + \sigma^2 I)^{-1}(y - \hat{x}_01),
\]  

(4.39)

where \(c_{\text{treat}}^{+}(t)\) is a row vector obtained by evaluating \(C_{\text{treat}}^{+}(t, t')\) for \(t' \in (0, 1, \cdots, T_{j0})\).

When \(\zeta_{\tau} = 0\), the treatment effect is null and the mean posterior trajectory is equivalent to that of Equation (4.33),

\[
\mathcal{F}_{\text{treat}}^{-}(t) = \hat{x}_0 + c_{\xi}(t)(\Sigma_{\xi} + \sigma^2 I)^{-1}(y - \hat{x}_01),
\]  

(4.40)
Thus, if \( \zeta = 0 \) leads to the minimum description length among the two models, then we conclude that the treatment effect is null, which implies that the control and treatment trajectories are equal except for the Gaussian noise. An example of this scenario is the expression of the gene encoding IL33 given LPS stimulation (shown in Figure 4.3). In this case \( \text{DL}_{\text{null}}^{\text{treat}} > \text{DL}_{\text{null}}^{\text{treat}} \); consequently the treatment effect is concluded to be null. Figure 4.3A shows the MAP trajectories for both the control and the treatment (LPS) time series under the null treatment model, and Figure 4.3B shows the MAP trajectories under the treatment model. Notice that the treatment trajectory under the null model is smoother than that under the treatment effect model. This results from the null model treating the additional variance observed in this time series as noise, whereas in the treatment effect model the additional variance is treated mainly as signal. This becomes apparent by contrasting the parameters of the two models (see Table 4.3): under the null model, \( \hat{\sigma}^2 \) is larger than under the effect model, and \( \hat{\gamma} \) and \( \hat{\zeta} \) are smaller.

Table 4.3: Inferred hyperparameters of the MAP trajectories of IL33 under the null and treatment effect models, shown in Figure 4.3.

<table>
<thead>
<tr>
<th></th>
<th>( \hat{\sigma}^2 )</th>
<th>( \hat{\zeta} )</th>
<th>( \hat{\gamma} )</th>
<th>( \zeta )</th>
<th>( \gamma )</th>
<th>DL</th>
</tr>
</thead>
<tbody>
<tr>
<td>null treatment effect</td>
<td>1.14</td>
<td>0.93</td>
<td>0.72</td>
<td>NA</td>
<td>NA</td>
<td>20.04</td>
</tr>
<tr>
<td>treatment effect</td>
<td>0.32</td>
<td>0.42</td>
<td>8.31</td>
<td>1.40</td>
<td>0.90</td>
<td>20.88</td>
</tr>
</tbody>
</table>

If the model in which \( \zeta \neq 0 \) leads to the minimum description length, we conclude that the treatment effect is not null, that is, the treatment stimulus leads to changes in the transcriptional expression of the gene. The expression of the gene encoding Il15 given LPS stimulation (shown in Figure 4.4) is an example. In this case the increase in goodness-of-fit is sufficient to justify a more complex model, with \( \text{DL}_{\text{null}}^{\text{treat}} > \text{DL}_{\text{null}}^{\text{treat}} \). Hence, IL15 is concluded to be differentially expressed in response to LPS. Figure 4.4A shows the MAP trajectories for both the control and the treat-
Figure 4.3: Example of null treatment effect. Maximum \textit{a posteriori} trajectories of the gene encoding IL33 for both the control and the treatment (LPS) time series under the null treatment effect model (A) corresponding to Equation (4.40) and under the not null treatment effect model (B) described in Equation (4.39). In this case, the treatment effect model has larger description length than the null treatment effect model; as such IL33 is concluded to not be differentially expressed in response to LPS.

Table 4.4: Inferred hyperparameters of the MAP trajectories of IL15 under the null and treatment effect models, shown in Figure 4.3.

<table>
<thead>
<tr>
<th></th>
<th>$\hat{\sigma}^2$</th>
<th>$\hat{\zeta}_x$</th>
<th>$\hat{\gamma}_x$</th>
<th>$\hat{\zeta}_r$</th>
<th>$\hat{\gamma}_r$</th>
<th>DL</th>
</tr>
</thead>
<tbody>
<tr>
<td>null treatment effect</td>
<td>0.85</td>
<td>0.95</td>
<td>0.97</td>
<td>NA</td>
<td>NA</td>
<td>17.67</td>
</tr>
<tr>
<td>treatment effect</td>
<td>0.20</td>
<td>1.07</td>
<td>8.31</td>
<td>0.92</td>
<td>0.80</td>
<td>16.37</td>
</tr>
</tbody>
</table>
Figure 4.4: Maximum a posteriori trajectories of the gene encoding IL15 for both the control and the treatment (LPS) time series under the null treatment effect model (A) corresponding to Equation (4.40) and under the treatment effect model (B) described in Equation (4.39). In this case, the treatment effect model has smaller description length than the null treatment effect model; as such IL15 is concluded to be differentially expressed in response to LPS.
**Figure 4.5**: Inferred trajectories of six genes under the treatment effect model: (A) Igf1, insulin-like growth hormone; (B) Pcolce2, procollagen C-endopeptidase enhancer 2; (C) Rab3il1, RAB3A interacting protein (rabin3)-like; (D) Ifi44, interferon induced protein 44; (E) Ifit1, Interferon-induced protein with tetratricopeptide repeats 1; and (F) Ifnb1, interferon beta 1.
Figure 4.6: Treatment effect corresponding to the trajectories shown in Figure 4.5.
Table 4.5: Inferred hyperparameters for the MAP trajectories shown in Figure 4.5.

<table>
<thead>
<tr>
<th>Gene</th>
<th>$\hat{\sigma}^2$</th>
<th>$\hat{\zeta}_\xi$</th>
<th>$\hat{\gamma}_\xi$</th>
<th>$\hat{\zeta}_\tau$</th>
<th>$\hat{\gamma}_\xi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Igf1</td>
<td>0.40</td>
<td>1.17</td>
<td>1.00</td>
<td>1.55</td>
<td>0.91</td>
</tr>
<tr>
<td>Pcolce2</td>
<td>1.03</td>
<td>0.62</td>
<td>0.95</td>
<td>3.25</td>
<td>1.33</td>
</tr>
<tr>
<td>Rab3il1</td>
<td>0.43</td>
<td>0.70</td>
<td>1.04</td>
<td>1.31</td>
<td>0.55</td>
</tr>
<tr>
<td>Ifi44</td>
<td>0.39</td>
<td>1.16</td>
<td>1.07</td>
<td>1.623</td>
<td>0.87</td>
</tr>
<tr>
<td>Ifit1</td>
<td>0.04</td>
<td>1.96</td>
<td>1.97</td>
<td>1.79</td>
<td>0.97</td>
</tr>
<tr>
<td>Ifnb1</td>
<td>0.36</td>
<td>0.20</td>
<td>0.93</td>
<td>3.33</td>
<td>1.27</td>
</tr>
</tbody>
</table>

We have argued that one of the advantages of using the LP to model temporal gene expression data is its flexibility and ability to model smooth functions of vastly different shapes, which is possible due to the fact that Gaussian processes do not impose any parametric form on their sampling space. We highlight this aspect of the LP in Figure 4.5, which shows the inferred trajectories of genes that are classified as differentially expressed by this model, i.e., genes for which $DL_{treat}^+ < DL_{treat}^-$. These are some examples of the wide variety of transcription expression patterns (in both the control and treatment times series) that we observe in our data. In Section 4.3.2, we provided some intuition into how the $\zeta$ and $\gamma$ hyperparameters affect the covariance matrix and the functions sampled from the LP. This influence of the parameters on the trajectories can be seen by comparing the inferred trajectories in Figure 4.5 and the values of the corresponding parameters, shown in Table 4.5. Contrast, for instance, the control and treatment trajectories of AFB1, shown Figure 4.5F as black and red curves, respectively. On one hand, the control trajectory is mostly flat, having small values of $\hat{\zeta}_\xi$ and $\hat{\gamma}_\xi$. On the other hand, the treatment trajectory is a lot more volatile, with a sharp increase of more than five fold within just a couple of time points; as such, it has larger values of $\hat{\zeta}_\xi$ and $\hat{\gamma}_\xi$ than the control trajectory.
4.5.4 Estimation of the treatment effect

As we described earlier, the temporal expression trajectory of a gene is a combination of its response to the treatment itself as well as to unintentional stimuli; as such, the inferred trajectory of the treatment time series, shown as red curves in Figure 4.5, reflects this compounded effect. It is useful, however, to separate the contributions of these two types of stimuli and to inspect the effect due solely to the intentional stimulus. This is very straightforward with the additive Langevin process. Thus, in addition to detecting whether the treatment effect is null or not, the LP allows us to also quantify the treatment effect, \( \mathcal{T}_{\Delta \text{treat}}(t) \) for any time point, \( t \). This is accomplished by the simple calculation below,

\[
\mathcal{T}_{\Delta \text{treat}}(t) = \mathcal{T}_{+ \text{treat}}(t) - \mathcal{T}_{- \text{treat}}.
\]  

(4.41)

This is especially useful for situations in which the control trajectory is not flat, which happen fairly frequently as described in Section 4.5.2. Figure 4.6 shows the treatment effect corresponding to the trajectories shown in Figure 4.5. Notice that, for a gene with nearly flat control trajectory, the treatment effect and the treatment trajectory are very similar to each other; examples of this scenario are the genes encoding Pcolce2, Rab3il and Ifnb1, shown respectively in panels B,C, and F of Figure 4.5 and Figure 4.6. Genes with non-flat control trajectories, such as Igf1, Ifi44 and Ifit1 (panels A,D and E, respectively), have clear differences between the treatment effect and the treatment trajectories. Notice, for instance, gene Ifi44. While its treatment trajectory indicates that there is a decrease in its mRNA concentration after time point 4, the treatment effect shows that the response to the treatment flattens out at that time point. Thus, it is useful to consider both the treatment trajectory and the treatment effect when studying the expression patterns of genes.
4.6 Test of flatness — results for the DC dataset

We have performed the treatment flatness test for all probe sets in the dendritic cell gene expression dataset. A summary of the results is shown in Table 4.6. The four-fold increase in the number of genes differentially expressed under CpG stimulation compared to LPS treatment is striking. We have compared the distributions of the parameters of the models, the difference in description length between the null and treatment effect models and the range of the treatment effect (see Appendix C for details), and have found no indication that this increase in the number of non-flat genes in the CpG experiment is artifactual. It is important to keep in mind, however, that classification is an inherently lossy procedure. It reduces our entire knowledge about a given system to a set of labels; as a result, part of the information available is discarded. In the flatness test, classification is decided based on which model, treatment or null treatment effect models, has the smallest DL. For some of the genes $\Delta DL = |DL_{treat} - DL_{null}|$ is large, in which cases the class assignment (flat or not flat) is obvious. For other genes, however, $\Delta DL$ is small, indicating that the probabilities of the treatment effect being null or not are similar to each other and that the class assignment is less clear. In order to provide a more complete picture of our results, we list in Appendix C the DL under the treatment effect and under the null effect models for some of the relevant genes described here.

Table 4.6: Summary of tests for treatment effects: Total number of probes determined to have a treatment effect (i.e., not flat trajectory) or a null treatment effect (i.e., flat trajectory) when stimulated with either LPS or CpG.

<table>
<thead>
<tr>
<th></th>
<th>Flat</th>
<th>Not Flat</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPS</td>
<td>43588</td>
<td>1513</td>
</tr>
<tr>
<td>CpG</td>
<td>38801</td>
<td>6300</td>
</tr>
</tbody>
</table>
4.6.1 Dynamics of gene expression

One of the advantages of modeling gene expression with a continuous time model such as the Langevin process is that it provides insight into the dynamics of gene expression.

One informative aspect of the time trajectory of gene expression is its half-time, $t_{1/2}$, which is defined as the time in which the treatment effect reaches half of its range, where the range is the maximum value of the treatment effect curve minus the minimal value of the same curve. For both CpG and LPS experiments, we have computed the half-time of each gene for which the treatment effect is not null. We also noted the direction of the trajectory at half-time.

![Histograms of half-time indices for (A) up- and (B) down-regulated genes under LPS and CpG stimulation. The lines are the estimated densities of the histograms, obtained by fitting a mixture of two normal distributions using MCLUST (Fraley and Raftery, 2002, 2006). For upregulated genes, the density of half-times for LPS was \[0.87\mathcal{N}(2.51, .54) + 0.13\mathcal{N}(6.05, .16)\] and for CpG was \[0.84\mathcal{N}(2.06, .94) + 0.16\mathcal{N}(5.86, .28)\], shown as red and green lines, respectively, in panel A. For downregulated genes, the density of half-time indices for LPS was \[0.94\mathcal{N}(3.15, 1.34) + 0.06\mathcal{N}(6.6, 0.2)\] and for CpG was \[0.95\mathcal{N}(2.55, 1.26) + 0.05\mathcal{N}(5.99, 0.21)\], shown as red and green lines, respectively, in panel B.]

Figure 4.7: Histograms of half-time indices for (A) up- and (B) down-regulated genes under LPS and CpG stimulation. The lines are the estimated densities of the histograms, obtained by fitting a mixture of two normal distributions using MCLUST (Fraley and Raftery, 2002, 2006). For upregulated genes, the density of half-times for LPS was \[0.87\mathcal{N}(2.51, .54) + 0.13\mathcal{N}(6.05, .16)\] and for CpG was \[0.84\mathcal{N}(2.06, .94) + 0.16\mathcal{N}(5.86, .28)\], shown as red and green lines, respectively, in panel A. For downregulated genes, the density of half-time indices for LPS was \[0.94\mathcal{N}(3.15, 1.34) + 0.06\mathcal{N}(6.6, 0.2)\] and for CpG was \[0.95\mathcal{N}(2.55, 1.26) + 0.05\mathcal{N}(5.99, 0.21)\], shown as red and green lines, respectively, in panel B.
Figure 4.7 shows a comparison of the half-times for genes differentially expressed under LPS as well as under CpG stimulation. A number of interesting observations can be made from this comparison. In general, the response time of genes that are differentially expressed under CpG stimulation is faster than that for genes under LPS stimulation. The gene expression in response to the stimuli happens in two clearly separated waves of transcription. The first burst of transcription peaks at around 1.47 hours (time index = 2.06) and 2 hours (time index = 2.5) for treatment with CpG and LPS, respectively. The second wave of transcription happens at around 20.5 hours (time index=5.86) and 23.4 hours (time index = 6.05) for CpG and LPS, respectively. Note in Figure 4.7A that, in addition to being delayed, the first peak in transcription of genes upregulated in response to LPS has a much smaller variance than that of CpG. Finally, note that the first burst of transcription is much stronger than the second one, representing between 85-95% of the transcription activity.

![Genes differentially expressed in LPS](image)

![Genes differentially expressed in CpG](image)

**Figure 4.8**: Histograms of half-time indices for (A) up- and (B) down-regulated genes under LPS and CpG stimulation.

Out of the 1513 genes that are differentially expressed under LPS stimulation,
438 genes are upregulated and 1075 genes are downregulated. For CpG stimulation, 2285 genes are upregulated and 4015 are downregulated. Hence, a slightly higher proportion of genes are upregulated in CpG (36%) compared to LPS (29%). Figure 4.8 presents a different view of the same data in Figure 4.7 — it shows a direct comparison of genes up- and down-regulated under a particular stimulation. As noted before, the distribution of half-times is bimodal, representing two waves of transcription. Note that, under LPS treatment, the distributions of up- and down-regulated genes are quite different from one another. In particular, the variance of the first peak in transcription for upregulated genes is significantly smaller than that of downregulated genes.

Another aspect of the gene expression temporal dynamics worth examining is its number of inflection points. An inflection point is a point in which the curve changes sign. Thus, the number of inflection points in the gene expression trajectory informs us of the number of times that particular gene changes from being up to down-regulated, and vice-versa. A curve with zero inflection points is monotonically increasing or decreasing. Thus, genes whose trajectories have no inflection points are up- or down-regulated through the entirety of the time course. In contrast, genes whose trajectories have one inflection point are transiently up- or down-regulated. Figure 4.9 provides a summary of the distribution of the number of inflection points in genes up- or down-regulated under LPS or CpG stimulation. For the CpG experiment, most genes have either zero or one inflection points, and the distribution of this number is relatively similar between down- and up-regulated genes. For the LPS experiment, there is a remarkable different in the distribution of the number of inflection points between up- and down-regulated genes. The majority of up-regulated genes are only transiently so, while the majority of down-regulated genes don’t change direction.
Figure 4.9: Histograms of number of inflection points for genes differentially expressed under (A) LPS and (B) CpG stimulation.

4.6.2 Cytokines

A total of 279 probe sets in the Affymetrix gene chip map to genes that encode cytokines (details about how this list was created are described in Appendix B). Table 4.7 shows the number of cytokines that were differentially expressed under one, both, or neither of the two treatments. Lists of the three groups of differentially expressed cytokine probe sets is shown in Appendix C.2, along with additional information about the description length of each model and the magnitude of the treatment effect.

Table 4.7: Summary of the number of differentially expressed cytokines under one, both, or neither of the two treatments.

<table>
<thead>
<tr>
<th></th>
<th>LPS and CpG</th>
<th>LPS only</th>
<th>CpG only</th>
<th>Neither</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of</td>
<td>12</td>
<td>6</td>
<td>56</td>
<td>205</td>
</tr>
<tr>
<td>cytokines</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The 12 probesets (which map to 10 cytokines) with non-flat treatment effect
Figure 4.10: Treatment effect of cytokines differentially expressed in response to both CpG and LPS stimulation.
in both experiments are shown in Figure 4.10. Their expression patterns are remarkably similar under both treatments. This does not come as a surprise since TLR4 and TLR9 share the conserved MyD88 pathway for the activation of NF-κB and subsequent induction of pro-inflammatory cytokines. The only cytokine with a markedly different expression pattern between the two treatments is IFN-α4, which raises much faster in response to CpG stimulation than to LPS stimulation. One possible explanation for this difference could be the fact that TLR4 and TLR9 utilize different pathways for the induction of type I IFNs (Kawai and Akira, 2006). On one hand, TLR9 activates type I IFN via a MyD88-dependent pathway, which activates both IRF7 and IRF3 (although IRF3 plays a less important role in the activation of transcription of IFN-I than IRF7) (Honda et al., 2005). On the other hand, TLR4 activates type I IFNs via adaptor molecules TRIF and TRAM, activating transcription factor IRF3. Additionally, LPS-bound TLR4 is only able to activate the TRIF pathway once it has been transported into endolysosomal compartment; as such, the activation of IRF3 is relatively delayed. Thus, these facts together would explain both the bi-modality in the treatment effect of IFN-α4 under CpG stimulation and its delayed response under LPS stimulation.

A search for these cytokines in the interferome database (Samarajiwa et al., 2009) reveals that seven out of the 10 cytokines differentially expressed under both treatments are known to be regulated by type I interferons. Namely these are: Tnfsf10, IL15, Cxcl11, Cxcl10, Ifnb1, Ifna4 and Csf2. Out of these, IL15, Cxcl11 and Ifna4 have also been reported to be regulated by type II interferons. A similar analysis for the 56 genes differentially expressed only under CpG stimulation reveals 30 genes regulated by interferons, eight of which are regulated by only type I interferon, two only by type II interferons (IL6 and IL11) and 10 by both types.
4.6.3 Transcription factors

A total of 2885 probe sets in the Affymetrix gene chip map to genes suggested to regulate transcription. Many of these are known transcription factors, and here we loosely refer to all entities in this list as transcription factors. Details about how this list was created and which genes it encompasses are described in Appendix B. Table 4.8 shows the number of these transcription factors that were differentially expressed under one, both, or neither of the two treatments.

Table 4.8: Summary of the number of differentially expressed genes potentially encoding transcription factors under one, both, or neither of the two treatments.

<table>
<thead>
<tr>
<th></th>
<th>LPS and CpG</th>
<th>LPS only</th>
<th>CpG only</th>
<th>Neither</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31</td>
<td>50</td>
<td>367</td>
<td>2437</td>
</tr>
</tbody>
</table>

Figure 4.11: Scatterplot of the maximum range in treatment effect between transcription factors independently determined to be differentially expressed in both CpG and LPS via the flatness test.
Among the 50 transcription factors differentially expressed in both groups, there are a few known to be activated by TLRs, such as Stat1, Stat2, Zfp287 and Fos. The latter is known to be activated in response to MyD88 signaling and form a complex with AP-1, in order to activate transcription of pro-inflammatory cytokines. We have compared the treatment effect of these genes; in general, the magnitude of their responses are comparable between the two treatments (see Figure 4.11). Notably, IRF7 is activated by both ligands and presents a strong reaction to both. We have noted in Section 4.6.2 that IRF7 plays an important role in the first wave of signaling by TLR9 and not TLR4, but it is interesting to observe that both pathways induce a substantial transcriptional response of this gene (see Figure 4.12A). Trim30 (tripartite motif-containing 30) also presents a strong response to both treatments (see Figure 4.12B). Recently, it has been shown that this transcription factor inhibits NF-κB activation, and thus, negatively regulates TLR signaling, promoting the degradation of TAB2 and TAB3 (Shi et al., 2008). Another noteworthy transcription factor is Pou3f1 (POU domain, class 3, transcription factor 1). Although it is differentially expressed in both groups, its response to CpG appears to be an order of magnitude stronger than to LPS, as shown in Figure 4.12C. Furthermore, this transcription factor has been extensively investigated in the literature for its role in the nervous system, and particularly Schwan cell, development. However, to the best of our knowledge, this transcription factor has not yet been implicated in the CpG-triggered pathways, nor in the immune response in general for that matter.

The complete list of transcription factors classified as differentially expressed in either LPS or CpG, or both, by the flatness test is provided in Appendix C.3, along with the description length for both flat and non-flat models. We would like to bring attention to one transcription factor that is classified as flat under LPS stimulation, but as not flat under CpG stimulation: Tcf2. This transcription factor presents the strongest response of all 1885 TFs we have tested. Over the time course, Tcf2
log₂ mRNA concentration increases by over 8-fold in response to CpG, as shown in Figure 4.13. Remarkably, there is no record in the literature about Tcf2 participating in innate immunity pathways.

4.7 Comparison of treatment effects across experiments

In Section 4.5, we addressed the questions of whether or not the control time series of a given experiment is flat and of whether or not the treatment time series of a given experiment differs from the control time series. The answer to the former tells us whether or not the gene is differentially expressed in response to the stimulation given in that experiment. Here, instead of considering the experiments separately, we are interested in making inferences about the relationships among the time series of a given gene under the different experiments. For each gene in the microarray chip, we have a total of four time series: two control time series, one resulting from the stimulation with LPS and another resulting from the stimulation with CpG. The additive nature of the Langevin process described in Equation (4.15) and restated here for reference,

\[ y_{ijkt} = \xi(t) + \Delta \xi_j(t) + \delta_{k1}(\tau(t) + \Delta \tau_j(t)) + \varepsilon_{ijkt}, \]
Figure 4.13: Transcription factor Tcf2 treatment effect in response to CpG stimulation.

allows us to make comparisons between any set of these four trajectories for a given gene. For example, we can ask questions such as whether or not the two control time series are the same, and whether the two controls are the equal but the treatment effects for each experiment are different. These comparisons, much like in Section 4.5, are done by computing and contrasting the description lengths of models with different sets of terms in Equation (4.15) set to zero. For example, to test whether or not the two control time series are equal, we would compute two models,

\[ \mathcal{M}_1 : y_{i00} = y_{i01} = \xi, \]

and

\[ \mathcal{M}_2 : \begin{cases} y_{i00} = \xi + \Delta \xi \\ y_{i01} = \xi - \Delta \xi \end{cases}, \]
and select the one with the smallest description length. This is effectively a test of whether $\Delta \xi = 0$ ($M_1$) or whether $\Delta \xi \neq 0$ ($M_2$). As hinted at earlier, we are not limited to testing just one term at a time; much to the contrary, we can test any number of terms at once.

### 4.7.1 Four model selection

In comparing a given gene’s response to two different stimulations, there are two most important questions that we are interested in addressing. First and foremost, we would like to know if the treatment effects under both experiments are the same or not. Second, we would like to quantify the common treatment effect, in the case in which the responses of the gene to both stimulations are the same; in the other case in which the responses are different, we would like to quantify the difference in treatment effects. Although less important, it is also interesting to make the same type of comparisons regarding the control effect. Hence, we compare four different nested models,

$$
\begin{align*}
M_1 : & \quad \xi \neq 0, \quad \Delta \xi = \tau = \Delta \tau = 0 \\
M_2 : & \quad \xi \neq 0, \quad \Delta \xi \neq 0, \quad \tau = \Delta \tau = 0 \\
M_3 : & \quad \xi \neq 0, \quad \Delta \xi \neq 0, \quad \tau \neq 0, \quad \Delta \tau = 0 \\
M_4 : & \quad \xi \neq 0, \quad \Delta \xi \neq 0, \quad \tau \neq 0, \quad \Delta \tau \neq 0.
\end{align*}
$$

Model 1, $M_1$, says that there are no differences among any of the four trajectories. Model 2, $M_2$, states that there are differences among the control trajectories, but both treatment effects are null. In model 3, $M_3$, the controls are allowed to differ from one another, but both experiments have the same treatment effect. Finally, in model 4, $M_4$, all four trajectories are different. Again, to determine which model best represents the data for a given gene, we fit the four models to the gene data, optimizing the parameters of the models, and then compare the four description
lengths, selecting the model with smallest DL.

Figure 4.14: Illustration of the four model selection using the gene encoding cytokine Ccl22. In this case, the best model is Model 1. DL_{M_1} = 10.17, DL_{M_2} = 14.58, DL_{M_3} = 12.46, and DL_{M_4} = 16.88.

This is illustrated in Figure 4.14, which shows the inferred trajectories under each one of the models above for the gene encoding cytokine Ccl22. The DLs of Models 1 through 4 are 10.17, 14.58, 12.46, and 16.88, respectively. Hence, the
model that best describes the behavior of the Ccl22 gene is Model 1, indicating that all four trajectories are identical, given the initial conditions and background noise. In this case, the improvements in data fit provided by the more complex models are surpassed by the increase in model complexity and consequent encoding cost.

The instances in which either Model 3 or Model 4 have the best DL are the most interesting cases, worth being further investigated. When Model 3 is the best model for a given gene, the treatment effects of the two experiments are nonzero, but are the same. These are the genes that present the same response to both LPS and CpG. In these cases, $\Delta \tau = 0$ and our main interest is in learning about $\tau$, the treatment effect. Take $\mathcal{T}_{jk}^{M_i}$ to be the mean posterior trajectory of time series $jk$ under model $M_i$. As described in Section 4.4.2, $jk$ takes values 00, 01, 10 and 11 for Control 0, Control 1, LPS and CpG, respectively. Then, the common treatment effect under Model 3 is

$$\mathcal{T}_{M_3}^{\tau} = \frac{1}{2} \left[ (\mathcal{T}_{M_3}^{10} - \mathcal{T}_{M_3}^{00}) + (\mathcal{T}_{M_3}^{11} - \mathcal{T}_{M_3}^{01}) \right]. \quad (4.43)$$

IFN-β is an example of a gene with the same treatment effect under both LPS and CpG stimulation. Figure 4.15A shows the mean posterior trajectories for the four time series for IFN-β under Model 3, and Figure 4.15B shows the actual treatment effect common to both groups. Note that, while we were able to say in Section 4.6 that IFN-β was differentially expressed under both treatments, and although the separate treatment estimates for both groups were very similar to each other (as shown in Figure 4.10), we were not able to make comparisons between the two groups. By analyzing all the data for this gene together, making these types of comparisons is very straightforward.

When Model 4 is the best model for a given gene, the two treatment effects are nonzero and different from one another. These are the genes that respond differently to LPS and CpG. In these cases, our main interests are to determine the two treat-
ment effects, as well as to quantify their difference. The difference in treatment effect is simply the mean posterior trajectory inferred by the Langevin process represented by $\Delta \tau$,

$$\mathcal{T}_{\mathcal{M}_4}^{\Delta \tau} = \frac{1}{2} \left[ (\mathcal{T}_{\mathcal{M}_4}^{10} - \mathcal{T}_{\mathcal{M}_4}^{00}) - (\mathcal{T}_{\mathcal{M}_4}^{11} - \mathcal{T}_{\mathcal{M}_4}^{01}) \right].$$  \hfill (4.44)

The treatment effect of LPS and CpG are respectively,

$$\mathcal{T}_{\mathcal{M}_4}^{\text{treatLPS}} = \mathcal{T}_{\mathcal{M}_4}^{10} - \mathcal{T}_{\mathcal{M}_4}^{00},$$ \hfill (4.45)

and

$$\mathcal{T}_{\mathcal{M}_4}^{\text{treatCpG}} = \mathcal{T}_{\mathcal{M}_4}^{11} - \mathcal{T}_{\mathcal{M}_4}^{01}. \hfill (4.46)$$

Figure 4.16 shows an example of a gene, IFN-\(\alpha\)2, for which the best model is Model 4. Panels A-D show the inferred trajectories of the four time series under this model, the difference in treatment between the two experiments, and the estimated treatment effects for LPS and CpG, respectively.

4.8 Comparison of treatment effects across experiments — results for the DC dataset

We have compared the response of all probe sets in the dendritic cell gene expression dataset to LPS and CPG by performing the four model selection on each of them. Table 4.9 presents a summary of these results. As expected and consistent with the flatness tests results, the vast majority of probesets had Model 1 as the best model, meaning that these genes do not respond to the treatments with LPS or CpG. In total, there were close to 3000 probe sets that responded to either LPS or CpG, or both. Interestingly, the number of genes with the same treatment effect for both experiments and those with different treatment effects were very close.

We have computed the $t_{1/2}$ (time in which the treatment effect reaches half of its magnitude) for the genes with best models being $\mathcal{M}_3$ and $\mathcal{M}_4$. Similarly to
Figure 4.15: Example of a gene, Ifnb1, for which $\mathcal{M}_3$ is the best model given the data. (A) Mean posterior trajectory of the four time series under $\mathcal{M}_3$, and (B) estimated treatment effect under $\mathcal{M}_3$.

Table 4.9: Summary of the number of probesets in the DC dataset for which each of the four models is best, i.e., has the smallest DL.

<table>
<thead>
<tr>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>41617</td>
<td>589</td>
<td>1490</td>
<td>1405</td>
</tr>
</tbody>
</table>

Section 4.6.1, we observe that the transcription of genes in response to both LPS and CpG proceed in two waves, an earlier one peaking at around 3 hours, and a later one after 24 hours. Regarding the intensity of the responses, we have computed the range of the treatment effect for these genes. Figure 4.17 shows the results. In general, the magnitude of the treatment effect for genes that respond similarly or differently to the two stimulations are very similar. There are a few genes that present particularly strong responses in $\mathcal{M}_3$ and to CpG in $\mathcal{M}_4$, and only five genes with log treatment effects larger than six: Sct in $\mathcal{M}_3$, and IFN-α2, Tcf2, D830007B15Rik and Mfsd7c (previously provisorily named BC011209) in $\mathcal{M}_4$. These are noted in Figure 4.17, and their treatment effects are shown in Figure 4.18 (except for IFN-α2,
Figure 4.16: Example of a gene, IFN-α2, for which $\mathcal{M}_4$ is the best model given the data. (A) Mean posterior trajectory of the four time series under $\mathcal{M}_4$, (B) estimated difference in treatment effect under $\mathcal{M}_4$, and treatment effect for LPS (C) and CpG (D) stimulation.
whose trajectories and treatment effects are presented in Figure 4.16). As explained in Section 4.6.2, type I interferons, such as IFN-α2, are transcriptionally activated in TLR4-stimulated cells only by IRF3, whereas in TLR9-stimulated cells it is activated by both IRF7 and IRF3. We believe that this could be the reason for the strong difference in expression of this gene between the two groups. Tcf2 is the gene with the strongest response to CpG, measured by magnitude of the treatment effect. It is a transcription factor that, to the best of our knowledge, has never been implicated in DC response to any type of PAMP. Interestingly, Tcf2 is associated with the gene ontology term for regulation of Wnt receptor signaling pathway (GO:0030111), where Wnt is a family of proteins with cytokine function, and one of its members, Wnt1 (shown in Figure 4.20), also has a very strong response to CpG. Regarding the other two probes, D830007B15Rik and Mfsd7c (BC011209), very little is known about them.

**Figure 4.17:** Box plot of the maximum range in the common treatment effect for genes for which $\mathcal{M}_3$ is the best model, and the CpG and LPS effects for genes for which $\mathcal{M}_4$ is the best model.
Figure 4.18: Treatment effect of genes with particularly strong common responses to both CpG and LPS (Model 3), or with very strong responses to CpG and not LPS (Model 4).
Only a small subset of the 279 probe sets mapping to cytokines were classified as differentially expressed: 14 were determined to have the same treatment effect for the two groups, and 24 were determined to have different treatment effects for the two groups. Table 4.10 summarizes these results. Among the cytokines with a common treatment effect for the two groups, IFN-β presents the strongest response. Other cytokines with common treatment effect are IL4, IL6, Cxcl11 and IFN-γ. The latter is commonly known for being produced by T cells, but studies have shown that DCs are indeed capable of producing them also (Moretto et al., 2007). Table 4.11 list all 14 cytokines, and Figure 4.19 shows the treatment effect of some of these cytokines with the strongest responses. Note that some cytokines appear twice in Table 4.11. That is because two probe sets in the Affymetrix chip mapping to the cytokine were classified as being differentially expressed.

Table 4.10: Number of probe sets that map to genes encoding cytokines classified to each of the four models

<table>
<thead>
<tr>
<th></th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>235</td>
<td>3</td>
<td>24</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 4.12 shows the 14 cytokines with nonzero and different treatment effects in response to LPS and CpG. Many of these cytokines are hallmark cytokines of T\textsubscript{H}1 cytokines, such as the three type I interferons (IFN-α-2, IFN-α-4 and IFN-α-9) and Lta. Figure 4.20 contrasts the treatment effects of some of these cytokines to LPS and CpG. It is interesting to note that in all but one case, the response to CpG is stronger than that to LPS. Note also that, in agreement with our previous observations, the responses of IFN-α 4 and 9 to CpG are not only of higher magnitude, but also happen earlier than the corresponding responses to LPS.
Table 4.11: Cytokines with same treatment effect in response to LPS and CpG stimulation.

<table>
<thead>
<tr>
<th>Gene name</th>
<th>DL_{M_1}</th>
<th>DL_{M_2}</th>
<th>DL_{M_3}</th>
<th>DL_{M_4}</th>
<th>ΔT^*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ifnb1</td>
<td>50.70</td>
<td>55.11</td>
<td>38.57</td>
<td>42.97</td>
<td>5.92</td>
</tr>
<tr>
<td>Cdl70</td>
<td>40.47</td>
<td>44.88</td>
<td>40.31</td>
<td>43.26</td>
<td>5.79</td>
</tr>
<tr>
<td>Cxcl11</td>
<td>46.56</td>
<td>50.95</td>
<td>32.99</td>
<td>37.39</td>
<td>5.78</td>
</tr>
<tr>
<td>Iil2b</td>
<td>38.88</td>
<td>43.29</td>
<td>38.14</td>
<td>41.04</td>
<td>4.51</td>
</tr>
<tr>
<td>Cxcl11</td>
<td>37.62</td>
<td>42.02</td>
<td>29.65</td>
<td>34.06</td>
<td>4.45</td>
</tr>
<tr>
<td>Ifng</td>
<td>42.11</td>
<td>46.52</td>
<td>32.37</td>
<td>35.37</td>
<td>4.12</td>
</tr>
<tr>
<td>Iil5</td>
<td>25.28</td>
<td>29.68</td>
<td>8.37</td>
<td>12.77</td>
<td>4.12</td>
</tr>
<tr>
<td>Iil6</td>
<td>39.54</td>
<td>43.96</td>
<td>36.72</td>
<td>39.42</td>
<td>3.84</td>
</tr>
<tr>
<td>Iil4</td>
<td>31.35</td>
<td>32.18</td>
<td>29.72</td>
<td>34.23</td>
<td>3.20</td>
</tr>
<tr>
<td>Tnfsf10</td>
<td>27.58</td>
<td>31.99</td>
<td>14.48</td>
<td>18.46</td>
<td>3.02</td>
</tr>
<tr>
<td>Cxcl10</td>
<td>33.59</td>
<td>37.99</td>
<td>25.45</td>
<td>29.93</td>
<td>2.85</td>
</tr>
<tr>
<td>Gdf15</td>
<td>22.39</td>
<td>24.97</td>
<td>21.83</td>
<td>26.12</td>
<td>2.82</td>
</tr>
<tr>
<td>Ebi3</td>
<td>21.45</td>
<td>25.86</td>
<td>18.91</td>
<td>23.32</td>
<td>2.75</td>
</tr>
<tr>
<td>Ccl3</td>
<td>24.77</td>
<td>29.18</td>
<td>22.24</td>
<td>25.56</td>
<td>2.57</td>
</tr>
<tr>
<td>Lif</td>
<td>17.78</td>
<td>22.19</td>
<td>17.15</td>
<td>21.55</td>
<td>2.40</td>
</tr>
<tr>
<td>Csf2</td>
<td>28.14</td>
<td>32.53</td>
<td>25.00</td>
<td>28.58</td>
<td>2.39</td>
</tr>
<tr>
<td>Tnfsf13/Tnfsf12</td>
<td>21.98</td>
<td>26.39</td>
<td>12.72</td>
<td>17.12</td>
<td>2.14</td>
</tr>
<tr>
<td>Ccl6</td>
<td>29.33</td>
<td>33.73</td>
<td>26.66</td>
<td>31.09</td>
<td>2.03</td>
</tr>
<tr>
<td>Ccl5</td>
<td>11.09</td>
<td>14.60</td>
<td>9.42</td>
<td>13.82</td>
<td>1.79</td>
</tr>
<tr>
<td>Tnfsf10</td>
<td>26.12</td>
<td>30.52</td>
<td>24.21</td>
<td>28.62</td>
<td>1.74</td>
</tr>
<tr>
<td>Ccl6</td>
<td>23.97</td>
<td>28.38</td>
<td>22.48</td>
<td>24.92</td>
<td>1.65</td>
</tr>
<tr>
<td>Lefty1</td>
<td>15.78</td>
<td>20.18</td>
<td>14.38</td>
<td>16.69</td>
<td>1.64</td>
</tr>
<tr>
<td>Tnfsf12/tnfsf13</td>
<td>-0.10</td>
<td>4.30</td>
<td>-2.83</td>
<td>0.15</td>
<td>0.93</td>
</tr>
<tr>
<td>Hmgb1</td>
<td>-21.19</td>
<td>-17.49</td>
<td>-21.53</td>
<td>-15.91</td>
<td>0.80</td>
</tr>
</tbody>
</table>

4.8.2 Transcription factors

The total number of transcription factors classified to each of the four models is shown in Table 4.13. There were 75 genes with nonzero and different treatment effects in response to CpG and LPS. In general, the magnitude of the response of genes to CpG was greater than to LPS, as shown in Figure 4.21A. However, for the most part, the genes with largest magnitude of the response to LPS and CpG were the same as can be seen in Figure 4.21B. However, there were two notable exceptions. Tcf2 was
Table 4.12: Cytokines with nonzero but different treatment effects in response to LPS and CpG stimulation.

<table>
<thead>
<tr>
<th>Gene name</th>
<th>DL_{M1}</th>
<th>DL_{M2}</th>
<th>DL_{M3}</th>
<th>DL_{M4}</th>
<th>ΔT_{CpG}^*</th>
<th>ΔT_{LPS}^*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ifna2</td>
<td>48.86</td>
<td>53.29</td>
<td>45.64</td>
<td>41.11</td>
<td>8.67</td>
<td>2.94</td>
</tr>
<tr>
<td>Ccl1</td>
<td>42.16</td>
<td>46.56</td>
<td>44.69</td>
<td>41.84</td>
<td>5.74</td>
<td>1.37</td>
</tr>
<tr>
<td>Lta</td>
<td>41.32</td>
<td>45.72</td>
<td>41.41</td>
<td>37.07</td>
<td>5.28</td>
<td>1.40</td>
</tr>
<tr>
<td>Tnfsf14</td>
<td>34.95</td>
<td>39.13</td>
<td>41.68</td>
<td>32.74</td>
<td>4.82</td>
<td>0.88</td>
</tr>
<tr>
<td>Wnt1</td>
<td>37.28</td>
<td>41.59</td>
<td>42.26</td>
<td>33.82</td>
<td>4.60</td>
<td>1.13</td>
</tr>
<tr>
<td>Cx3cl1</td>
<td>16.09</td>
<td>20.50</td>
<td>25.82</td>
<td>3.04</td>
<td>4.30</td>
<td>2.72</td>
</tr>
<tr>
<td>Ifna4</td>
<td>20.93</td>
<td>25.31</td>
<td>25.82</td>
<td>18.19</td>
<td>4.28</td>
<td>2.69</td>
</tr>
<tr>
<td>Ifna9</td>
<td>16.79</td>
<td>21.19</td>
<td>24.07</td>
<td>16.50</td>
<td>3.91</td>
<td>1.80</td>
</tr>
<tr>
<td>Ccl9</td>
<td>14.34</td>
<td>18.86</td>
<td>15.20</td>
<td>13.73</td>
<td>2.97</td>
<td>1.89</td>
</tr>
<tr>
<td>Tnfsf10</td>
<td>24.50</td>
<td>28.90</td>
<td>17.24</td>
<td>16.86</td>
<td>2.94</td>
<td>2.99</td>
</tr>
<tr>
<td>Kitl</td>
<td>5.29</td>
<td>9.69</td>
<td>11.03</td>
<td>-18.43</td>
<td>2.27</td>
<td>1.45</td>
</tr>
<tr>
<td>Cntm3</td>
<td>-0.49</td>
<td>3.57</td>
<td>-0.76</td>
<td>-5.50</td>
<td>1.30</td>
<td>0.46</td>
</tr>
<tr>
<td>Grn</td>
<td>-18.72</td>
<td>-17.40</td>
<td>-13.12</td>
<td>-34.46</td>
<td>0.74</td>
<td>0.32</td>
</tr>
<tr>
<td>Grn</td>
<td>-20.32</td>
<td>-19.51</td>
<td>-15.28</td>
<td>-44.46</td>
<td>0.66</td>
<td>0.24</td>
</tr>
</tbody>
</table>

the transcription factor with strongest response to CpG, but had a treatment effect close to zero for LPS. In contrast, the magnitude of the response of Mafb to LPS was much larger than to CpG. It presented a strong downregulation in response to LPS, and a mild upregulation followed by a mild downregulation in response to CpG. Figure 4.22 shows the treatment effect of these and a few other transcription factors with different transcriptional responses to LPS and CpG.

Table 4.13: Number of probe sets that map to genes encoding transcription factors classified to each of the four models

<table>
<thead>
<tr>
<th></th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2716</td>
<td>22</td>
<td>72</td>
<td>75</td>
</tr>
</tbody>
</table>

We had noted in Section 4.6.3 that IRF7 was differentially expressed under both LPS and CpG treatments. Here we show that the treatment effect to both stimulations is the same. In fact, among the transcription factors with the same transcrip-
tional response to the two treatments, IRF7 presents the largest treatment effect. Figure 4.21 shows the common treatment effect of IRF7, as well as that of other transcription factors that present the same response in both experiments such as Stat2, Mxd1, Trim30, Sp100 and Pml.

4.9 Discussion

We have developed methodology to analyze temporal gene expression data. Our method models gene expression with additive Langevin processes, which are Gaussian processes obtained from the Langevin equations. Although using other types of Gaussian processes would be appropriate, we are particularly keen to use Langevin processes for two main reasons. First, the Langevin equations can provide a useful analogy to the process driving gene expression in cells. Second, the covariance function of the LP has properties that are in line with some of our prior expectations about the data. Specifically, this covariance function specifies that the correlation between time points increase over time.

This method was developed specifically to analyze a novel dataset produced by the Kepler laboratory. This dataset consists of gene expression data measured with Affymetrix gene chips over the course of 46 hours (eight time points) after dendritic cells were stimulated with either LPS (which is known to bind to TLR4) or CpG (which is known to bind to TLR9). The goal of the dataset and analysis presented here have been to unravel some of the differences and similarities between the transcriptional responses of these two pathways.

One of the features of the Langevin process is that it is continuous in time, which makes the LP a natural approach to modeling gene expression. Hence, samples from the LP are entire curves, instead of single points. Consequently, it allows us to model the actual underlying transcription process, instead of being limited to making inference about the sampling times. For example, using the LP, we were able
to compute features of the temporal trajectory of genes such as the half-time, the
time at which the trajectory reaches half of its magnitude. Doing this allowed us to
make very interesting observations. For example, we revealed that the transcriptional
response of genes in dendritic cells stimulated with CpG is typically more immediate
than that of cells stimulated with LPS. We also showed that the responses to both
CpG and LPS happen in two waves of transcription, an earlier one that happens
within a couple of hours from the initial stimulus, and a later one that happens
about a day after the stimulation.

Because of the additivity of our model, we are able to separate the contribution to
the observed expression level of a gene due to its response to the intentional stimulus
from that due to unintentional (but nevertheless significant) stimuli. Our results
show that indeed many of the control trajectories are not flat. Therefore, in order
for an accurate representation of a gene’s response to a given stimulus, it is extremely
important that the control effect be estimated and taken into account. In addition
to determining whether or not the control and/or treatment effects are null, we can
also quantify the actual treatment effect, that is, the transcriptional response of a
gene to the intentional stimulus. We have performed these tests of flatness for all
probesets in the dendritic cell dataset, and have identified nearly four times more
genes that are transcriptionally activated in response to CpG stimulation compared
to LPS stimulation.

By taking one step further and analyzing both experimental groups simultane-
ously, we are able to draw direct comparisons between a gene’s response to both
groups. In this simultaneous analysis we consider all four time series available for
the given gene in our dataset, and treat each one of those as independent Langevin
processes. Then, we can ask a number of questions about which components of the
model are null or not. Here, we have considered four models of increasing complexity
and performed a model selection using the description length of the model and the
classification criteria. We have been primarily interested in genes classified as one of two of the models above: Model 3, which determines that the treatment effect in response to CpG and LPS are equivalent, and Model 4, which determines that there is a treatment effect in response to both stimuli, but that the effects are different. By performing this model selection procedure on the entirety of the DC dataset, we have identified thousands of genes that are differentially expressed under the two experiments. By computing the treatment effects for the genes that have different effects under each condition, we have demonstrated that, in general, the response of genes to CpG was more intense than the response of the same genes to LPS.

In conclusion, we have described here a novel approach to the analysis of temporal gene expression data. Our model can be applied to both long and short time series, and can be applied to datasets with variable sampling time points as well as datasets with missing data. Being a Gaussian process, our model does not impose any parametric form on the shapes of the temporal trajectories, as it is able to model virtually any smooth trajectory. This model allows us to ask a number of interesting questions about the response of a gene to one or more stimuli. Importantly, it allows us to test whether or not there is a treatment effect, and if so, allows us to quantify it. Additionally, by considering multiple treatments together, we can directly compare the treatment effects of a gene in response to each given stimulus. By analyzing the dendritic cell dataset with this approach we have been able to identify genes that are unique to one of the treatments, as well as genes that respond to both treatments similarly or differently.
Figure 4.19: Treatment effects of some of the cytokines that respond similarly to LPS and CpG.
Figure 4.20: Comparison of treatment effects in response to LPS and CpG for cytokines that respond differently to the two treatments.
Figure 4.21: Magnitude of the treatment effects of transcription factors in Models 3 and 4. (A) Box plots comparing magnitude of response of genes with similar response to both LPS and CpG (Model 3), and of genes with different responses to LPS and CpG (Model 4). (B) Scatter plot comparing the magnitude of the response to LPS and to CPG for genes that respond differently to the two stimulations.

Figure 4.22: Treatment effect of transcription factors that present different responses to stimulation with LPS and CpG.
Figure 4.23: Treatment effect of transcription factors that present the same response to stimulation with LPS and CpG.
In the previous chapter, we presented a novel dendritic cell temporal gene expression dataset in response to LPS and CpG stimulation (see Section 4.2), and developed methodology to model and analyze these data. There we modeled each gene independently of the others, and addressed questions pertinent to single genes. For example, we were able to perform comparisons among the control and stimulated temporal profiles of a given gene, as well as to compare the treatment effect of that gene in response to LPS and CpG. In this chapter, we continue to explore the response of dendritic cells to these different stimulation by considering entire groups of genes simultaneously. We have achieved this by developing a methodology to cluster temporal gene expression patterns.

Cluster analysis is an important step towards a deep understanding of many types of data, including the dendritic cell dataset investigated here, and aids in the identification of groups of genes with similar temporal transcriptional responses to a given stimulus. The rationale is that this similarity could be a result of genes being co-regulated, sharing related functions, or participating in the same pathway. Hence, this type of analysis has the potential to shed light on aspects of the transcriptional
regulatory networks within the cell. As such, the goal of our clustering analysis is to provide inference on the underlying structure of the temporal gene expression data, including both the numbers and the identities of genes assigned to both common and group-specific components, along with quantification of the uncertainties in the model.

The clustering method presented here builds on the methodology developed in Chapter 4. Here the temporal mRNA expression is still modeled with the Langevin process, but now all genes assigned to the same cluster are assumed to follow the underlying trajectory described by the same Langevin process. We perform clustering using a Dirichlet process prior, described in Section 3.2, which is particularly appealing due to the fact that it allows the number of gene expression patterns (i.e., the number of clusters) to be inferred directly from the data. In sections Section 5.1 and Section 5.2, we will describe our model and some aspects of the posterior simulation, respectively. In Section 5.3 we discuss some relevant issues regarding MCMC simulation, along with some alternatives to address these issues, including our own solution. In Section 5.4, we validate our model using a synthetic dataset, and then use it to analyze the dendritic cell gene expression data. We conclude this chapter in Section 5.5 with a discussion about features and shortcomings of the model, highlighting future directions for further exploration of our model.

5.1 Statistical Model

Let $y_i$ be the vector of length $L$ of log-transformed mRNA intensity measurements of the $i^{th}$ gene from microarray readouts, with $i$ ranging from 1 to $N$, the total number of genes in the dataset. Suppose there is a partition of the genes into $K$ clusters of sizes $\{N_1, \ldots, N_K : \sum_{j=1}^K N_j = N\}$, such that for the $k^{th}$ cluster of genes, we have the $N_K L$-vector (i.e., vector of length $N_K \cdot L$) $y^{(k)} = (y_1, \ldots, y_{N_K})'$ of expression values
of the genes in cluster $k$. Our method makes two key assumptions:

1. Genes in the same cluster, $k$, follow the same underlying trajectory, $x_k$, up to
   additive constants.

2. The number, $K$, of underlying trajectories generating the expression profiles
   in the microarray experiment is unknown and must be inferred from the data.

We are primarily interested in two quantities: the number of clusters, $K$, and
the classification set $c = (c_1, \ldots, c_N)'$ of the data into the $K$ clusters, where $c_i = k$
with $k = \{1, \ldots, K\}$ indicates the cluster to which gene $i$ is assigned to. In this way,
$c_i$ represents which underlying trajectory $k = 1, \ldots, K$ has generated the expression
profile of the $i^{th}$ gene. Take $\theta_i$ to be the vector of parameters defining the underlying
distribution of the $i^{th}$ gene, and $\phi$ to denote the vector of unique values of $\theta$, such that
$\phi = \{\phi_k : k = 1, \ldots, K\}$. Then, the observed trajectory $y_i$ arises from a DP-induced
mixture,

$$y_i|\theta_i \sim F(\theta_i) \tag{5.1}$$

$$\theta_i \sim G \tag{5.2}$$

$$G \sim DP(\alpha G_0). \tag{5.3}$$

As described in Section 3.2.6, we can deconvolve this mixture by introducing the
classification variables $c$,

$$y_i|c_i \sim f(y_i|\theta_{c_i}),$$

$$c_i|\pi \sim Mn(\pi), \tag{5.4}$$

$$\pi \sim DP(\alpha, G_0).$$

Next, we explain in detail the distribution of each of the variables involved in this
model.
Distribution of the Dirichlet process hyperparameters

We use the scheme suggested by Escobar and West (1995) to model the Dirichlet process precision hyperparameter, $\alpha$, where the prior is a Gamma distribution

$$\alpha \sim Ga(a, b)$$

and the resulting posterior is a mixture of Gamma distributions

$$\alpha | \eta, N, K \sim q_\eta Ga(a + K, b - \log(\eta)) + (1 - q_\eta) Ga(a + K - 1, b - \log(\eta))$$

where $\eta$ is a latent variable to aid in the computation of the posterior distribution of $\alpha$. Its conditional distribution is $\eta | \alpha \sim Be(\alpha + 1, N)$. $q_\eta$ is the mixture coefficient and has the form $q_\eta = (a + K - 1)/(a + K - 1 + N[b - \log(\eta)])$.

Distribution of the mixture coefficients

The mixture coefficients $\pi$ follow a Dirichlet process prior

$$\pi_{1:K} | \alpha \sim DP(\alpha).$$

Distribution of the classification variables

The classification variable $c$ can assume values between 1 and $K$ according to a Multinomial prior

$$c_i | \pi_{1:K} \sim Mn(\pi_{1:K})$$

By integrating over the mixing proportions $\pi$, the prior for $c_i$ becomes

$$p(c_i = k) = \frac{N_k}{N - 1 + \alpha}$$

$$p(c_i = K + 1) = \frac{\alpha}{N - 1 + \alpha}$$
with \( k = \{1, \ldots, K\} \) and \( c_i = K + 1 \) implying the creation of a new cluster.

The joint posterior distribution of the classification vector given the data is implicitly specified by the hierarchical model, but cannot be written in closed form. The marginal posterior distribution of an individual classification variable \( c_i \), however, has been shown by Neal (2000) to be

\[
p(c_i = k | y_i, \theta_k, \sigma^2) = B \frac{N_k}{N - 1 + \alpha} p(y_i | \theta_k, \sigma^2, c_i = k) \tag{5.5}
\]

where \( B \) is a normalizing constant ensuring that the probabilities add up to 1.

This prior says that the probability of allocating an observation to an existing cluster should be proportional to the number of observations already allocated to that cluster, and the probability of allocating an observation to a new cluster should be proportional to \( \alpha \). We have also explored other priors that express different beliefs about the probability of allocation of an observation to a given cluster. One such prior that we have explored reflects the belief that observations should be allocated to clusters so to minimize the description length of the clustering configuration. This prior determines that the complexity term of the description length should be

\[
\frac{(N + K - 1)!}{(K - 1)! \prod_{j=1}^{K} n_j !}\]

Distribution of the data

The observed expression values of the \( i^{th} \) gene, \( y_i \), are generated by the underlying trajectory of each cluster \( h_k \) with probability \( \pi_k \) as described by the mixture

\[
y_i \sim \sum_{k=1}^{K+1} \pi_k f(\cdot | \theta_k), \tag{5.6}
\]

with

\[
f(t | \theta_k) = x(t | \theta_k) + \hat{y}_{i0} + \varepsilon \tag{5.7}
\]
where \( \hat{y}_{i0} \) is the expected value of the trajectory of gene \( i \) at time point 0. In the case of the dendritic cell dataset described in Section 4.2, \( \hat{y}_{i0} \) is the expected mRNA level of gene \( i \) prior to stimulation with LPS or CpG, and it is approximated by its likelihood estimator. \( \varepsilon \sim N(0,\sigma^2) \) is the background noise. Note that we treat the noise as common to all of the data, independently from cluster assignment. This choice is purely based on our understanding of the gene expression data and its sources of noise; for other datasets, if deemed appropriate, the model can be easily extended to different noise levels for each cluster by setting \( \varepsilon_k \sim N(0,\sigma^2_k) \). Finally, \( x(t) \) is a Langevin process,

\[
x(t|\theta_k) \sim \mathcal{GP}\left(\mu(t), C(t, t'|\theta_k)\right),
\]

with \( \mu(t) = 0 \) and covariance function as defined in Equation (4.7) in Section 4.3.2, and restated here for clarity,

\[
C(t, t+\tau|\theta_k) = \mathbb{E}\left[\left(x(t) - \mu(t)\right)\left(x(t+\tau) - \mu(t+\tau)\right)\right]
\]

\[
= \frac{\zeta_k^2}{\gamma_k}\left\{\gamma t - (1 - e^{-\gamma t})\left[1 + \frac{1}{2}(1 - e^{-\gamma t})e^{-\gamma \tau}\right]\right\},
\]

so that \( \theta_k = (\gamma_k, \zeta_k) \). Note that the covariance function of the trajectory depends only of the model hyperparameters \( \theta \) and time, and does not depend on the observed expression \( y_i \).

Provided that we know the cluster assignment of the \( i^{th} \) gene, \( c_i \), then the data likelihood is given by

\[
y_i|c_i = k, \theta_k, \sigma^2, \hat{y}_{i0} \sim N(\hat{y}_{i0} \mathbf{1}, \Sigma_k)
\]

where \( \mathbf{1} \) is a column vector of ones, and \( \Sigma_k \) is a covariance matrix with components,

\[
\Sigma_k(t, t') \equiv C(t, t'|\theta_k) + \sigma^2 \delta_{t,t'},
\]

with \( \delta_{t,t'} \) the Kronecker delta.
where \( \delta \) is Kronecker’s delta.

Similarly, the joint distribution of the expression values of all genes in cluster \( k \) is

\[
y^{(k)}|\theta_k, \sigma^2, \hat{y}_0^{(k)} \sim N(\hat{y}_0^{(k)}, \Sigma^{(k)})
\]

where \( \hat{y}_0^{(k)} = (\hat{y}_{10}, \ldots, \hat{y}_{nk0})' \) is a \( NKL \)-vector, with \( \hat{y}_{i0} = \hat{y}_{i0} \mathbf{1} \). The covariance matrix \( \Sigma^{(k)} \) is a \( NKL \)-by-\( NKL \) matrix formed of \( N_K \) identical \( L \)-by-\( L \) matrices \( \Sigma_k \), whose entries are defined in Equation (5.9) above. Note that the size of the covariance matrix of a given cluster grows rapidly with the number of genes assigned to that cluster. Computations, and particularly inversions, of such large matrices would represent a substantial computational burden, and would effectively impose a practical limit to the sizes of potential clusters. The repetitive, blocky nature of these matrices, however, allows for algebraic manipulations that simplify computation and restrict matrix operations to square matrices of dimension \( L \) (Kepler, 2008).

5.2 Posterior simulation

In our model, we have been able to write the full conditional distribution of the individual cluster indicator variables \( c_i \) in closed-form, shown in Equation (5.5), but not their joint posterior distribution. Thus, we have implemented a Gibbs sampler to explore the posterior distribution of our variables of interest. Below we provide some of the relevant details of the posterior simulation:

**Initialization**

The initialization step consists of specifying an initial configuration of the random variables. The algorithm is initialized in one of three ways:

1. Assuming all genes are generated by the same underlying pattern, that is,

\[
c_1^0 = c_2^0 = \ldots = c_N^0 = 0.
\]
2. Assuming each gene is generated by a different underlying pattern, that is, 
\[ c_1^0 = 1, c_2^0 = 2, \ldots, c_N^0 = N. \]

3. Assuming that there are \( K \) underlying patterns governing the expression profiles of the genes in the dataset, but that we don’t know which underlying pattern each gene follows. In this case, the genes are randomly assigned to \( K \) initial clusters.

Unless otherwise specified, we generally start our sampler from configuration (2).

**Updating**

The updating step consists of iteratively sampling the random variables from their full conditionals. For each iteration \( h \) and each gene \( i \):

1. Compute \( p(c_i | \pi_{1:K}, \theta_{i:K+1}, y_i) \), the posterior probability of the \( i^{th} \) gene’s expression profile being generated by each of the possible underlying patterns, according to Equation (5.5).

2. Draw \( c_i^h \) based on \( p(c_i | \pi_{1:K+1}) \)

3. Move gene \( i \) from cluster \( c_i^{h-1} \) to cluster \( c_i^h \). If \( c_i^h \neq c_j^h, \forall i \neq j \), then a new cluster is created. If \( c_i^{h-1} \) contained only gene \( i \), then it is deleted.

4. Update the hyperparameters \( \theta_{c_i^h} \) and \( \theta_{c_i^{h-1}} \) of clusters \( c_i^h \) and \( c_i^{h-1} \), respectively.

The priors on the Langevin covariance parameters, \( \gamma \) and \( \zeta \), as well as the prior for the error variance \( \sigma^2 \) are the same as those described in Section 4.4.4. Because we are not able to obtain a closed form posterior distribution for the covariance function parameters, we update their values by using their maximum likelihood estimators given the genes currently allocated to the cluster they represent.
Termination

The algorithm is monitored over the iterations, and it is stopped after it has converged and generated the desired number of posterior samples. Convergence is determined by both analysis of the traceplots of the random variables samples, and by comparison of independent runs of the Gibbs sampler on the same data but with different initial conditions.

5.3 Post-processing of MCMC sampler output

MCMC samplers are tools for sampling from the posterior distribution of variables of interest, when it is not available in closed form. The posterior samples obtained this way provide an empirical estimation of their posterior distribution. In our clustering problem, we are ultimately interested in providing a single best representation of the data in a clustering configuration, preferably accompanied by quantifications of the uncertainties associated with it. The natural answer to this question would be to output the \textit{maximum a posteriori} (MAP) clustering configuration along with credible intervals. The posterior distributions of clustering configurations, however, tend to have shallow peaks. In other words, the posterior probabilities of very similar clustering configurations are similar, such that the difference in posterior probability between the MAP configuration and the next best one is very small. Hence, this problem of choosing one final representation of the data in a clustering arrangement continues to be an open question.

In addition to choosing the MAP configuration, other approaches have been suggested. One such approach commonly used when clustering with mixtures of simple multivariate Gaussian distributions is to use a number of posterior samples to compute a posterior mixture of normals. Then, input the weights and parameters of the components of this posterior mixture into a mode-search algorithm. Commonly
there are many fewer modes than there are mixture components. These modes are the treated as the final clusters, and an observation is assigned to a given mode based on its probabilities given the mixture components that map to this mode. This is a compelling approach given that it provides a clustering configuration that summarizes the posterior distribution. In our application, performing a mode search in the space of Langevin processes would be less straightforward, and thus, we do not employ this approach here.

Medvedovic and Sivaganesan (2002) suggest another approach, which consists of computing the posterior pairwise probabilities of two genes’ expression patterns being generated by the same underlying profile. The posterior pairwise probability that two genes’ temporal expression profiles are generated by the same underlying process is

\[
p_{ij} = \frac{\text{# of samples in which } c_i = c_j}{\text{total # of posterior samples}}.
\]

Taking \( N \) to be the total number of genes in the data being clustered, these pairwise probabilities are then put together into an N-by-N matrix of such pairwise probabilities, called the posterior pairwise probability matrix (PPPM). This PPPM is then used as a similarity measure inputted into an existing agglomerative ad-hoc clustering algorithm, together with the posterior number of clusters. One advantage of this method is that the final clustering configuration produced represents, in some sense, an average of the posterior distribution of the clustering configurations.

Instead of either of these two approaches, we use a different optimization method to produce our final clustering configuration, as will be described in the following sections. Although we do not use PPPMs for producing the final clustering configuration, we do believe that this representation conveys a great deal of information about the strength of the clusters, as well as about the relationships among different clusters. PPPMs are visualized with the aid of heatmaps (see Figure 5.4 for an
example) sorted according to the chosen clustering arrangement, \( c \). The resulting clusters can be identified along the diagonal of the sorted heatmap. The strength of our belief about the integrity of a cluster is indicated by its colors: lighter clusters contain genes whose expression patterns are strongly believed to be generated by the same process, whereas our certainty about the similarity of genes in darker clusters is lower. Similarly, the sorted heatmap also shows the similarity among genes in different clusters, providing visual information regarding the degree of overlap among different clusters. Thus, we use PPPMs as a tool that provides further insight into the similarity of the transcriptional temporal patterns of different genes, as well as into the stability (or lack thereof) of our final clusters.

5.3.1 Practical issues with the Gibbs sampler

A well-documented aspect of Gibbs samplers is that they have a tendency to become trapped in local modes of the posterior distribution. Celeux et al. (2000) points out that the incremental nature of the Gibbs sampler and its inability to move groups of items simultaneously between clusters are the culprits for its poor mixing. The reasoning is that the intermediate states between modes of the posterior represent states of very low probability, unlikely to be traversed by the Gibbs sampler, such that the sampler stays trapped in a local mode that represents an erroneous clustering configuration of the data.

We have observed this phenomenon with our sampler when clustering artificial data (such as that described in Section 5.4.1). In these datasets, we have noticed that the MCMC sampler often “overfits,” that is, separates true clusters into smaller “sister” subclusters. Each of these subclusters are typically composed of items belonging only to a single true cluster. As such, the underlying trajectory of sister subclusters are very similar to each other. Because of this, the probability of an observation given any of these sister subclusters are relatively similar to each other (and
proportional to the subclusters’ sizes). As a result, the subclusters tend to persist for long periods during the course of the MCMC chain before one of the subclusters entirely absorbs all of the observations assigned to the other sister subclusters.

Before addressing the issue of the slow convergence of our Gibbs sampler, we first revisit our goal and motivation for clustering our data. As we described in the beginning of this chapter, clustering is a way to identify common and distinct patterns in the data by imposing an artificial structure to it. Here, we are interested in identifying groups of genes with similar overall responses to a given experimental condition. While obtaining full posterior distributions and inference on the data arrangement into clusters would be ideal, it is not necessary. For our purposes, a single clustering configuration of the data that is optimal in some sense suffices.

Here we present an algorithm that provides an arrangement of the data into clusters that is optimal in that it provides the clustering configuration with the minimal description length. This algorithm relies on the observed fact that our sampler tends to split clusters into subclusters, and that these subclusters are pure; in other words, these subclusters only contain items from a single true cluster. The basic premise of this procedure, then, is that the given clustering configuration or one obtained by merging sets of clusters of this configuration is an optimal clustering configuration of the data, and the goal is to find this optimal clustering arrangement. We emphasize that this algorithm provides a local optimum not guaranteed to be the global optimum, and that it will only find the minimal DL configurations that are nested within the input configuration. Thus, only configurations produced by the sampler after it has converged to this local near-optimal configuration should be considered as input to the post-clustering merge algorithm.
5.3.2 Post-clustering merge algorithm

Let \( c = (c_1, \ldots, c_N)' \), where \( c_i = k \) and \( k = \{1, \ldots, K\} \), represent the current arrangement of the data into clusters. There exist \( \binom{K}{2} \) distinct pairs of clusters in \( c \). Take \( p \) and \( q \) to denote two different clusters in \( c \), and let \( \phi_{pq} \equiv (p, q) \) denote a pair of clusters in this configuration. Let \( c^{(0)} \) be the starting from configuration, then each iteration \( h \) of the merge algorithm proceeds as follows:

1. For each pair of clusters, \( \phi_{pq} \), let \( c^{(h)}_{pq} \) be a clustering configuration equal to \( c^{(h)} \) except for the fact that cluster \( p \) and \( q \) are merged into a single cluster.

   Compute the change in the description length, \( \Delta DL_{pq} \equiv DL_{c^{(h)}} - DL_{c^{(h)}_{pq}} \).

2. Let \( DL_{p'q'} \equiv \min_{\phi_{pq}} \Delta DL_{pq} \).

3. If \( DL_{p'q'} < 0 \), then:

   (a) Merge clusters \( p' \) and \( q' \), such that the size of the new configuration \( c^{(i)} \) of the data is \( K^{(i)} = K^{(i-1)} - 1 \).

   (b) Repeat step 1.

4. Otherwise, return the current clustering configuration, \( c^{(h)} \).

5.4 Results

5.4.1 Synthetic dataset results

We first tested our method on an artificial dataset. These data were constructed in the following manner. First, we sampled trajectories from a Langevin process with covariance function parameters \( \gamma = 1/3 \) and \( \zeta = 2 \), and using an error variance \( \sigma^2 = 0.1 \). The curves were obtained for the time interval (the Langevin process input) between zero and nine. Three of the sampled curves (shown in Figure 5.1A) were selected for our artificial data, such that each of them was used as the underlying
trajectory for one cluster in the artificial data. We then subsampled each of these three curves, retrieving their values at 10 time points, from zero to nine, using increments of one. The vector containing the ten subsampled values of a given underlying trajectory was then used as the mean of a ten-dimensional multivariate normal distribution with variance matrix, $\tau I$, where $\tau = 0.1$. The data assigned to each cluster were then sampled from these multivariate Gaussian distributions. The final dataset consists of three clusters of sizes 20, 25, and 30, shown in Figure 5.1B. The data in Cluster 1 follow a trajectory that is constantly increasing, representing genes that are upregulated during the entire measured interval. Both Clusters 2 and 3 have underlying trajectories that start with a positive slope, reach a peak and then decrease again. The overall shape of these two trajectories are similar, but they differ in that the scale of the change in Cluster 2 is larger than, and peaks prior to, the underlying trajectory of Cluster 3.

![Figure 5.1: Artificial Langevin data. (A) Underlying trajectories of the three clusters comprising the artificial dataset. (B) Complete artificial dataset.](image)

The optimal clustering configuration, i.e, the clustering configuration that minimizes the DL, obtained by clustering these data with the DP Gibbs sampler and performing the post-clustering merge step is shown in Figure 5.2. This cluster con-
configuration is exactly equal to the configuration known to be true, with the same number of clusters (three) and the same assignment of items into clusters. Additionally, the inferred trajectories for each of the three clusters are very similar to the true ones used to generate the artificial dataset. Figure 5.3 compares the underlying trajectories used to create the data to those inferred by our clustering algorithm.

**Figure 5.2**: Langevin artificial data inferred minimal DL cluster configuration. Underlying trajectory of, and data assigned to, (A) Cluster 1, (B) Cluster 2, and (C) Cluster 3.

**Figure 5.3**: Comparison of underlying trajectory used to construct the artificial data and the inferred trajectory of the corresponding cluster in the minimal DL clustering arrangement obtained by our clustering method. (A) Cluster 1, (B) Cluster 2, and (C) Cluster 3.

The results of clustering this artificial dataset provide a good example of the type of problems caused by the incremental nature of the Gibbs sampler, described in Sec-
tion 5.3.1, and how they can be addressed by a post-clustering merge step. We ran the Gibbs sampler for approximately 40000 iterations, starting with 75 clusters, each containing one observation. The chain gradually decreased the number of clusters, reaching the neighborhood of five to four clusters near part way through the 40000 iterations. For the rest of the duration of the chain, the sampler explored the space of four clusters. Two out of these four clusters were very stable and mapped exactly to true clusters 1 and 2, whereas the other two clusters continued to exchange observations among themselves. This can be seen in Figure 5.4, which shows the pairwise posterior probability matrix (PPPM) computed based on the last 800 iterations of the Gibbs sampler. Notice that the 20 observations from Cluster 1 were grouped together in every one of these last configurations sampled by the Gibbs sampler. Similarly, all 25 observations from Cluster 2 clustered together. Observations from Cluster 3, however, were separated into two subclusters, each of which had a core group of observations that remained together, with most of the observations from the original Cluster 3 visiting both of these two subclusters with varying frequency. This can be seen in greater detail in Figure D.1, which shows a PPPM restricted for observations belonging to the true Cluster 3 only, and in Figure 5.5, which shows the fluctuation in the number of observations assigned to either one of these two extra clusters over the last 500 iterations of the Gibbs sampler. This issue seems to be caused by the interplay between two opposing forces. On one hand, larger clusters have larger prior weight from the DP viewpoint, and a higher complexity penalty from the encoding viewpoint. On the other hand, because we use the MLE of the covariance function parameters for each cluster, the likelihood tends to favor small clusters. As such, neither of the two subclusters is able to completely dominate the other one, such that the sampler remains trapped in this local optimum.

Although the Gibbs sampler tends to become trapped in these types of local modes with extra subclusters, in all of our tests the “overfitting” subclusters have
Figure 5.4: Pairwise posterior probability matrix (PPPM) for Langevin data computed based on the last 800 iterations output by the Gibbs sampler.

Figure 5.5: Number of observations assigned to the two subclusters containing observations from true Cluster 3 over the last 500 Gibbs sampler iterations.
been pure, that is, all observations in any given subcluster come from the same true original cluster. Additionally, when we compare the description length of the overfitted clustering configurations output by the Gibbs sampler with those obtained by merging the correct subclusters, the merged configuration presents a smaller overall description length than the overfitted configuration. In the case of the Langevin data, the merge procedure produced a single merge which merged the two subclusters referred to in Figure 5.5 mapping to true Cluster 3, and improved the DL of the system by about 8.5%.

5.4.2 Transcription factor response to LPS

In Section 4.6.3, we performed tests of the treatment effect of transcription factors in response to LPS stimulation. The tests described a total of 81 transcription factors as being differentially expressed in response to this treatment. Here we perform a cluster analysis of these 81 genes. We ran the MCMC chain for 40000 iterations, which converged to five distinct clusters. The post-clustering merge procedure did not merge any of the five clusters. A traceplot of the number of clusters over the course of the MCMC chain and a PPPM with the relationship among the 81 genes are shown in Figure D.2 and Figure D.3, respectively.

Figure 5.6 shows the data and inferred trajectories of four of the five produced clusters. In general, we observe a different temporal transcriptional pattern for each of these clusters, with two clusters representing upregulated genes and two clusters representing downregulated genes. The fifth cluster contains 45 probesets and the inferred trajectory is flat. The fact that the trajectory of this cluster is flat does not mean that the genes allocated to this cluster have a null treatment effect, i.e., are non-responsive to LPS stimulation. Instead it suggests that the genes assigned to it represent a wide variety of temporal expression profiles, such that no substantial subset of it has the same trajectory. Thus, in order for the sampler to produce
clusters whose trajectories were a relatively accurate representation of the temporal profile of the genes assigned to them, the genes assigned to this large cluster would have to be separated into a very large number of subclusters, each containing a small handful of genes. From the viewpoint of minimizing the description length of the clustering configuration, this solution with many small clusters is not advantageous, and hence the genes are clustered together.

![Image of trajectory profiles and data of clusters obtained for transcription factors stimulated with LPS in the dendritic cell dataset.](image)

**Figure 5.6**: Trajectory profiles and data of clusters obtained for transcription factors stimulated with LPS in the dendritic cell dataset. (A) Cluster 1 with one gene (IRF7), (B) Cluster 2 with five genes, (C) Cluster 3 with 17 genes, and (D) Cluster 4 with 13 genes.

The inferred trajectories of the two clusters that represent downregulated transcription factors differ primarily in the rate of down regulation (i.e., the slope of
the descent): the mRNA levels of genes in Cluster 2 seem to decrease more rapidly than those of genes in Cluster 3. Five probesets were allocated to Cluster 2, two of them mapping to the transcription factor Mafb, and the other three mapping to the transcription factors Cebpα, Pparg, and Nr1h3. Seventeen probesets were allocated to Cluster 3, three of them mapping to Dnmt3a, a DNA methyltransferase thought to regulate transcription by promoting *de novo* methylation of DNA. The other 14 probeset map to Hey1, Gata6, Egr2, Gtf2i, Fli1, Sox18, Klf7, Rxra, Asb2, Ppp1r12b, Braf, Nfia, Zfp295, and Hivep3. Notably, Hey1 participates in the Notch signaling pathway and has been recently shown to inhibit transcription of IL-12p70 (Alvarez et al., 2011) and its transcription is, in turn, inhibited by IFN-γ. Both of these cytokines are transcriptionally responsive to LPS as shown in Figure 4.19. Interestingly, the transcription factor Fli1 has been shown to be downregulated in human macrophages after LPS stimulation, and provides a self-regulatory feedback loop that limits the expression of metalloproteases, attenuating the extent of tissue destruction associated with inflammatory responses (Ho and Ivashkiv, 2010).

Clusters 1 and 4, shown in Figure 5.6A and Figure 5.6B respectively, represent upregulated transcription factors. A single gene, IRF7, was assigned to Cluster 1. As described in previous sections, IRF7 is strongly upregulated in dendritic cells in response not only to LPS but also to CpG. No other transcription factor produces as strong of a response as IRF7, and thus, this gene is allocated to a cluster by itself. Cluster 4 represents genes that are upregulated in response to LPS, though to a lesser extent than IRF7. There are 13 probes in this cluster that map to seven unique genes: Adar, Pml, Stat1, Stat2, Sp100, Trim30, and Usf1. All of these transcription factors, except Usf1, are known to be induced by at least one type of interferon and participate in the activation of transcription of genes in the interferon response. As already mentioned in Section 4.6.3, Trim30 has been demonstrated to inhibit NF-κB activation, and thus, negatively regulate TLR signaling (Shi et al., 2008). Pml is
another member of the Trim family of transcription factors, and has been shown to be activated by type I, II, and III interferons, and to positively regulate IFN-γ signaling.

5.4.3 Transcription factor response to CpG

We also performed a clustering analysis of transcription factors in response to CpG. The 100 probes with largest magnitude of treatment effect among the ones determined to be differentially expressed in response to CpG were selected for this analysis. The final clustering configuration is shown in Figure 5.7, and the number of probesets assigned to each of these clusters is shown in Table 5.1, along with their corresponding genes. This final configuration contains nine clusters, representing a wide variety of transcriptional responses to CpG stimulation. One of the clusters (Cluster 9) presents a nearly flat trajectory. As discussed in Section 5.4.2, this can happen when the alternative is to separate a cluster into too many subclusters, which penalizes the DL. Among the remaining eight clusters, four of them represent upregulated genes. The other four clusters represent downregulated genes, and can be separated into two groups: one in which the genes are downregulated through the entire course of the measurements, and another in which the initial downregulation of the genes is followed by an upregulation.

The four upregulated clusters present trajectories that differ both in the magnitude and rate of upregulation. Most notably, the trajectory of Cluster 1 is upregulated by almost ten-fold (in log scale!). This cluster is exclusive to a single transcription factor, Tcf2, which, as mentioned in Section 4.8.2, has never been implicated in the immune response. The other three clusters represent more moderate levels of upregulation, though still substantial. Figure 5.8A compares the half-time of these four trajectories. Recall that the half-time of a trajectory is defined as the time at which the treatment effect reaches half of its total amplitude (as described
Figure 5.7: Trajectory profiles and data of clusters obtained for transcription factors stimulated with CpG in the dendritic cell dataset. (A) Cluster 1 with one probeset (Tcf2), (B) Cluster 2 with four probesets (mapping to three TFs: Batf, Tcf2, and Nfkbiz), (C) Cluster 3 with 13 probesets, (D) Cluster 4 with 21 probesets, (E) Cluster 5 with nine probesets, (F) Cluster 6 with 16 probesets, (G) Cluster 7 with six probesets, (H) Cluster 8 with 17 probesets, and (I) Cluster 9 with 12 probesets.
Table 5.1: Summary of the nine-cluster configuration of the transcription factors differentially expressed in response to CpG. Note that in several instances multiple probesets map to the same gene, in which cases the number of genes in the third column will not match the number of probesets shown in the second column. The 100 probesets clustered here map to 82 unique genes.

<table>
<thead>
<tr>
<th>Cluster ID</th>
<th>Number probesets</th>
<th>Genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Tcf2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Batf, Tcf2, Nfkbiz</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>Jun, Irf6, Lhx2, Nr4a3, Mxd1, Foxh1, Tpb, Grhl1, Hsf2, Zbtb38, Casz1</td>
</tr>
<tr>
<td>4</td>
<td>21</td>
<td>Elf3, Irf7, Mybl2, Trim30, Elf, Adar, Batf2, Zbtb32, Pml, Tbx21, Stat1, Stat2, Pou3f1, Sp100, Zbtb5</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>Nfe2l3, Lyl1, Fli1, E2f8, Mxd3, Bhlhb8, Hes5,Asb1, Zfp710</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>Irt3, Vdr, Cebpa, Bmyc, Mxd4, Patz1,Tbx6, Zfp2, Mafb, Runx1, Rfxank, Zfp710, Egr3, Zhx3, Zbtb16</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>Nks1, X1110051B16Rik, Glis3, Zbtb16, Lhx9, X1110051B16Rik</td>
</tr>
<tr>
<td>8</td>
<td>17</td>
<td>Mycn, Tcf21, Fosl1, Zfhx4, Rhox5, Gata6, Rorc, Hoxa3, Zbtb16, Egr2, Nfkb2, Prrx1, Prrx2, Hr, Tead2, Nr3c2, Tbx4</td>
</tr>
<tr>
<td>9</td>
<td>13</td>
<td>AFFX.r2.Bs.dap.3.at, Relb, Irf4, Esrrg, Eomes, Gli1, Cdkn2a, Klf13, X4931423N10Rik, Adar, Zfp667, Sp100, Mxd1</td>
</tr>
</tbody>
</table>

in Section 4.6.1). The half-time of Clusters 1, 2 and 3 are very close to each other, with the main difference among the trajectories of these clusters being the amplitude of the response. The upregulation of genes in Cluster 4 is delayed in comparison to the other three upregulated clusters. Similarly we have compared the half-times of downregulated clusters, shown in Figure 5.8B. The downregulation of genes in Cluster 7 is faster than that of genes in Cluster 8, but while the former levels off at about time point 6 (which maps to about 6 hours post-stimulation), the latter continues to be down regulated through the entire time course. Finally, Figure 5.8C shows the half-times of transiently downregulated clusters. Here, we provide both half-times of the curves, in order to be able to compare both the down- and up-regulated parts of the trajectories. In general, the transcriptional response of genes in Cluster 6 is
delayed in comparison to those in Cluster 5. Furthermore, the total amplitude of the trajectory of Cluster 5 is larger than that of Cluster 6.

5.5 Discussion

In this chapter, we have continued to explore the dendritic cell gene expression data presented in Chapter 4. In the previous chapter, we presented a method based on the Langevin process to model the temporal transcriptional profiles of individual genes, being able to perform comparisons among the control and stimulated temporal profiles of a given gene, as well as to compare the treatment effect of that gene in response to LPS and CpG. In the present chapter, we have built upon this model, taking it one step further. Instead of modeling one gene at a time, here we have extended our Langevin model to consider the trajectories of multiple genes simultaneously and have used it to perform clustering analysis of our data. Although clusters are entirely artefactual structures that we impose on the data, they are very useful tools for identifying common expression patterns among large numbers of genes, such that genes grouped together could be functionally related and/or transcriptionally coregulated.

In our model, each cluster is associated with one Langevin process trajectory, which represents an underlying temporal expression profile. All genes assigned to a given cluster are assumed to follow the underlying trajectory of the cluster, up to additive constants and given measurement noise. We take an optimization approach to the clustering problem, and search via MCMC sampling for arrangements of the data into clusters that minimize the description length. Different from the majority of clustering methods applied to gene expression, the number of clusters is not a pre-requisite in our clustering approach. Instead, our sampler explores the space of clustering configurations with a variable number of clusters, and by doing so, infers the number of clusters from the data directly.
Figure 5.8: Half max of trajectories of clusters obtained by clustering transcription factors stimulated with CpG. Inferred trajectories and corresponding half max points for upregulated clusters (A), constantly downregulated clusters (B), and transiently downregulated clusters (C). In C we also show the second occurring half max point in the trajectory. The first one provides insight into the rate of the initial downregulation of the gene, whereas the second one provides insight into the rate of the subsequent upregulation. The half-time of a trajectory is the time at which the treatment effect reaches half of its total amplitude.
The utility of our clustering approach was demonstrated on an artificial dataset, in which we were able to recover the clusters from which the data were originally generated. In addition to recovering the correct associations among the artificial genes, the inferred trajectories of the final clusters were close approximations of the Langevin curves used to create the data. We then moved on to analyze parts of the dendritic cell dataset. Here, we focused our attention on transcription factors, obtaining some interesting results. We identified four transcriptional patterns among the transcription factors differentially expressed in response to LPS. As suspected in Chapter 4, IRF7 was the transcription factor with strongest response to LPS and no other TF presented changes in mRNA levels with the same magnitude as IRF7. Another seven transcription factors were upregulated in response to LPS, with most of them being interferon-inducible genes, such as Stat1 and 2, Sp100 and two members of the Trim family, Trim30 and Trim19. Many more transcription factors were differentially expressed in response to the CpG treatment. This phenomenon has not been restricted to transcription factors; rather, our CpG stimulation has in general elicited a stronger and broader transcriptional response than the LPS stimulation. We were able to identify eight distinct patterns of gene expression among the 100 transcription factors we have clustered. There were four groups of genes that were upregulated, with Tcf2 presenting a remarkable and unique upregulation of its mRNA level. We compare the dynamics of temporal profiles representing each of these groups via half-time. We also identified two continuously downregulated and two transiently downregulated groups of transcription factors.

The two types of analysis, described in this and previous chapters, are complementary to each other, and together allowed us to explore at varying depths the transcriptional response of dendritic cells to both LPS and CpG. Our clustering approach was developed to have a number of features that are desirable in the type of analysis we have performed, such as no constraints on the number of clusters and al-
following genes in the same cluster to have different initial conditions. The utility of our approach has been emphasized in this chapter both by validation with artificial data and by its application to the transcription factor dataset. In developing and applying this model, however, we have encountered a number of interesting challenges imposed by various aspects of the model. Here, we describe some of the most relevant of these challenges, discussing our approach to addressing them, as well as some alternative ideas. Additionally, we highlight some of the aspects of the model that leave the most room for improvements, pointing out directions for further developments.

Gaussian processes are a compelling approach to modeling gene expression time series. One significant issue with these models is the computational burden imposed by the matrix operations they entail. Matrix inversion, in particular, is arguably the major bottleneck in performing analysis with Gaussian processes. This issue is especially relevant for clustering problems, where the dimension of the covariance matrix of a given cluster grows with the number of time points and the number of genes assigned to it. In practical terms, computations required for matrix inversions effectively impose a limit on the size of clusters. Our model has been designed such that the covariance matrix of any cluster is composed of smaller block matrices, which have the same dimension as the number of measured time points. Kepler (2008) has described mathematical manipulations of these types of block matrices obtained by “stacking” Langevin processes, such that we are able to restrict matrix inversion operations to these smaller square matrices. Although this stacking property of our covariance matrices make the use of the Langevin process within a clustering framework possible, this method is still very computational intensive. Performance improvements could be obtained both by performing the matrix inversion computations with highly parallelized graphical processing units as described by Suchard (2010), as well as by streamlining of matrix operations (Brahim-Belhouari and Bermak, 2004) and restricting the type of allowed matrix operations (Mackay
Considerations of computational efficiency in Langevin process models are intrinsically related to how the hyperparameters of the Langevin process, $\gamma$ and $\zeta$, are modeled. Ideally, we would assign reasonable prior distributions to these variables and update them using an MCMC algorithm to sample directly from their posterior distributions. The problem, however, is that we are not able to obtain closed form posterior distributions for either of these parameters. A reasonable alternative, from our optimization approach to clustering, has been to update $\gamma$ and $\zeta$ by using their maximum likelihood estimators. These are obtained via iterative optimization algorithms, such as the Nelder-Mead method. Matrix inversions are performed in each iteration of the optimization, which has to be performed each time the probability of a gene being assigned to a cluster is computed. While this strategy works, other strategies for updating the covariance function parameters could prove to be much more efficient both in the number of computations required and in exploring the space of these variables. The lack of closed form posterior distributions is a rather common problem encountered in these types of applications. As such, the literature is ripe with approaches to address it. An alternative to using MLEs is to explore the posterior distribution of these variables via another MCMC sampler nested within the original one. It is not clear whether this solution would require more or less computations than our current approach, but it is very possible that it would provide better estimates of these variables, allowing the sampler to converge to the posterior distribution of clustering configurations at a faster rate.

Like in any modeling decision, choices of how to stack the Langevin process lead to different implications and assumptions. As described earlier, in our model genes in the same cluster follow a single Langevin trajectory, differing only by the vertical offset (the initial condition) and background noise. As such, the covariance matrix induced by the Langevin process of this cluster $k$ will be formed by $N_k^2$ blocks, where
$N_k$ is the number of genes assigned to cluster $k$ and all blocks are represented by the same matrix. A direct consequence of this construction is that for a given time point, the measurements of genes assigned to the same cluster are treated as exchangeable. Figure 5.6D provides a good visual example of this phenomenon. This assumption implied by our model deserves careful consideration and further exploration. One general solution would be to assume that observed mRNA levels of genes in a given cluster are the result of two additive Langevin process: one common to all genes in the cluster and one specific to each gene. The resulting covariance from this model would still be formed by smaller blocks, but now the blocks on its main diagonal would differ from the off-diagonal blocks and from each other. This raises several issues related to modeling choices, as well as to computational tractability. Exploring these issues and other covariance structures would be a worthwhile exercise, and could potentially lead to significant improvements in clustering.

The Dirichlet process priors provide a compelling approach for performing inference on the number of clusters without requiring a pre-specification of the number of clusters; as such, they are attractive as priors in clustering problems and have become increasingly more popular. The DP prior puts higher probability on larger clusters, leading to one of its most prominent properties, commonly referred to in the literature as the “rich get richer” phenomenon. While this feature of the DP works well for some datasets, it certainly does not apply to all data. For example, if we were clustering the time series of all 45000 probesets in the dendritic cell dataset, we would expect to have one very large cluster with genes that do not respond to the stimulation provided, and a few smaller clusters representing the different transcriptional responses elicited by the stimulation. On the other hand, when clustering only a subset of genes that we have reason to believe are transcriptionally responsive to the treatment, we do not want to make this type of assumption about sizes of the clusters. With this concern in mind, we have explored other priors for the
number of clusters. From the information theoretic point of view, an appropriate choice of prior on the clustering structure would be one that leads to the minimum description length code. In this sense, the multinomial coefficient, referred to here as the Mn prior, provides a sensible prior. It is somewhat similar to the DP prior in that it favors clusters of variable sizes. Another choice we have explored is what we refer to as the “naïve” encoding, with uniform probability over clusters and favoring clusters of similar sizes. In exploring these two priors as well as the DP prior, we have found that the naïve encoding leads to slower convergence rates, penalizing the creation of clusters too strongly. We have found that performing simulated annealing when using this naïve prior improves the clustering results. The DP and the Mn priors provided very similar results for the synthetic Langevin data presented in Section 5.4.1, as well as to other test datasets not presented here. For example, with the Langevin data, both samplers converged to configurations with four clusters that when post-processed with the merge procedure lead to the configuration of the data known to be true. A more systematic comparison of these and other priors could provide further insight into this model, and lead to improvements in the exploration of larger and more optimal sets of clustering configurations.

We have discussed above some of the open questions of our methodology and possible directions for future investigations, but there are many other aspects of the model deserving of attention. For example, in Section 5.1 we talked about our choice to model the error variance as common to all clusters. Although we believe that this is a more appropriate choice than the alternative of considering different error levels for each cluster, we have observed that the choice of error structure affects the outcome, and gaining further insight into this aspect of the model is likely to be important. Another open question relevant, but not specific, to this model is how to treat the posterior samples obtained from the MCMC algorithm. Here we have proposed using a merge procedure to obtain a single and optimal (in some sense)
clustering configuration. This solution has been adequate for our purposes but it is limited and we have suggested in Section 5.3 two other solutions to this problem. These and other approaches to this problem could be explored in future studies. In summary, the method we have developed here has proved useful in our analysis of the dendritic cell temporal gene expression data. This method has a number of desirable properties and features for this type of analysis, and here we have explored and highlighted several of them. There remain, however, numerous open questions and possibilities for further development of this methodology that could lead to entire dissertations.
6

Prediction of peptide-MHCI binding

6.1 Introduction

Prediction of peptide binding to Major Histocompatibility Complex class I (MHCI) is a crucial step in the development of subunit vaccines. The peptide-MHCI complexes are required for T cell activation and thus for the initiation of the adaptive immune response. Although MHCI binding does not alone determine the immunogenicity of peptides, it plays an important part, being a major bottleneck that separates immunogenic peptides from non-immunogenic ones. Hence, the ability to predict the binding between peptides and MHCI molecules would greatly reduce costs and accelerate the experimental process of identifying immunogenic peptides, which can then be used in the development of vaccines and therapies against neoplastic, infectious, and autoimmune diseases.

Myriad approaches have been applied to the prediction of peptide-MHCI binding. These methods can be divided into two broad categories: 1) MHCI structure-based methods, which use crystallized structures of MHCI molecules to develop computational models of the interaction between MHCI and peptides (Zhang et al., 1998);
and 2) peptide sequence-based methods, which infer the physico-chemical preferences of a particular MHCI allele by analyzing the amino acid sequence of peptides with known affinity to it, where peptides with dissociation constants lower than a certain threshold, typically 500 nM (Peters et al., 2006), are classified as binders, and otherwise as non-binders. Earlier prediction methods used the amino acids frequencies in each position of MHCI-eluted peptides to derive binding motifs and position specific scoring matrices (PSSMs), among which SYFPEITHI (Rammensee et al., 1999) and BIMAS (Parker et al., 1994) have been publicly available and used extensively by the experimental community. As the number of peptides in the MHCI databases increased, so did the number of different machine learning methods that were applied to this problem, which include artificial neural networks (Gulukota et al., 1997), support vector machines (Donnes and Elofsson, 2002), hidden Markov models (Yu et al., 2002), Gibbs sampling (Nielsen et al., 2004), and classification trees (Segal et al., 2001; Zhu et al., 2006; Peters et al., 2003).

Our primary goal in this chapter is to develop methodology to guide experimental research in identifying potential vaccine epitopes. In a given microbial genome, there are tens of thousands of peptides and the experimental assessment of the affinity between each peptide and an MHCI molecule represents a significant cost in terms of time and resources. The investigator has to consider the benefits of identifying binders versus the cost associated with experimentally testing non binders in order to decide which and how many peptides will be tested in the laboratory. This type of concern can be best addressed by the use of decision-theoretic approaches. Here we formalize such approach to training decision trees to differentiate binders from non binders and show how costs that reflect this experimental tradeoff can be incorporated into the training of classifiers to increase their utility.
6.2 The problem of unbalanced datasets

While some of the existing methods mentioned above achieve outstanding performance in predicting binding between peptides and certain MHCI alleles, all of them suffer from the fact that the available data for training is heavily enriched for one class of peptides (either binders or non-binders). There is a vast literature on the impact of class distribution of training sets on the performance of the prediction algorithms (for further readings see Chawla et al., 2004), and although there is not a straightforward answer to the question of what the ideal class distribution of training datasets is, it has been suggested that a balanced distribution or the estimated distribution of the target population should be used. Moreover, it is a well known phenomenon that highly unbalanced datasets are detrimental to classifier performance. The imbalance in the peptide-MHCI binding data results from the experimental methods used to produce them: either elution assays, in which case all of the data consists of binders, or binding assays in which peptides are tested for binding or affinity to a particular MHCI allele, leading to a dataset enriched in non-binders since most peptides do not bind a given allele. Datasets generated in different laboratories using different assays and conditions are often inconsistent with each other and thus combination of datasets can be very difficult.

Here we investigate how best to use unbalanced datasets to train algorithms for the prediction of peptide-MHCI binding. Although there is no universally agreed upon method for dealing with unbalanced data, several techniques have been proposed to deal with this issue and have been demonstrated to improve prediction accuracy depending on the context in which they are used (Chawla et al., 2004). Elkan (2001) showed how to make a standard learning algorithm yield cost-sensitive results when trained with an unbalanced dataset. Another successful strategy is referred to as cost-sensitive methods, in which weights are used to compensate for
the imbalance in the ratio of the two classes. Other methods pre-process the data to achieve a balanced class distribution. In particular two methods stand out: 1) under-sampling, where random cases of the majority class are deleted until both classes have the same number of cases; and 2) oversampling, where random cases of the minority class are duplicated until both classes have the same number of cases.

Our primary goal is to determine whether or not the accuracy of peptide-MHCI binding prediction can be improved by the use of methods that compensate for the training data imbalance, such as under-sampling and cost-sensitive methods. We do so by addressing the following questions: First, can under-sampling procedures improve the utility of classifiers used in the context of peptide-MHCI binding? Second, can the imbalance in the data be overcome, at least partially, by cost-sensitive methods that weight more heavily peptides belonging to the minority class and thus improve the classifier’s performance? If so, we would like to determine the relationship between data imbalance and training costs and provide it as a guideline to training other classifiers.

6.3 Methods

6.3.1 Approach

The development of subunit vaccines is a multi-step process; at each stage, the investigators must decide whether a particular peptide warrants further investment or should be omitted from further experimentation. These decisions must be informed, either explicitly or implicitly, by consideration of the costs incurred in continuing the experiments and of the potential reward for a positive discovery. One must also estimate the probability that a given decision will be erroneous, either as a false positive (continuing to invest in a peptide that will prove to be unsuitable) or a false negative (discontinuing tests on a peptide that would have worked). Let the cost of misclassifying a binder be denoted $\kappa_2$ (for type 2 error) and that for misclassifying
a non-binder, $\kappa_1$. We refer to $\kappa_2$ as the “real-world” cost, as it can be interpreted as the number of non-binders an investigator is willing to test in the laboratory in order to find one binder. Finally, suppose that a family of classifier can be parametrized with the continuous vector $\theta$. Then the cost incurred in making a decision using the classifier $T(\theta)$ is

$$K(\phi|\theta) = \tau_+(\phi)\kappa_2 c_-(\phi|\theta) + \tau_-(\phi)\kappa_1 c_+(\phi|\theta),$$

(6.1)

where $\tau_+$ and $\tau_-$ are indicators of true class and $c_+ (\cdot | \theta)$ and $c_- (\cdot | \theta)$ are indicators of the classification induced by $T(\theta)$.

The expected cost over all peptides is

$$EK(\theta) = \pi \kappa_2 \epsilon_2(\theta) + (1-\pi) \kappa_1 \epsilon_1(\theta)$$

(6.2)

where $\pi$ is the proportion of binders in the population and $\epsilon_i$ is the expected rate of type $i$ errors. We would like to find the classifier $T^*$ that minimizes this expected cost, given the difficulty that only finite datasets are ever available. We will define a loss function of the same form as Equation (6.1),

$$L(D|\theta) = n \{ p \lambda_2 \epsilon_2(\theta) + (1-p) \lambda_1 \epsilon_1(\theta) \}$$

(6.3)

$$+ \sum_{\phi \in D} \{ \tau_+(\phi) \lambda_2 \delta c_-(\phi|\theta) + \tau_-(\phi) \delta c_+(\phi|\theta) \}$$

where the $\lambda_i$ is the misclassification cost used at the training level and the residual type 2 (1) classification error is defined on positive (negative) peptides as

$$\delta c_-(+) (\phi|\theta) \equiv c_-(+)(\phi|\theta) - \epsilon_2(1)(\theta)$$

(6.4)

This expression may be further abbreviated to

$$L(D|\theta) = n \{ p \lambda_2 \epsilon_2(\theta) + (1-p) \lambda_1 \epsilon_1(\theta) \} + R(D; \theta, \lambda).$$

(6.5)
The expected cost is minimized at \( \hat{\theta} \) where
\[
0 = \frac{\partial}{\partial \theta} EK(\hat{\theta}) = \pi \kappa_2 \frac{\partial \epsilon_2}{\partial \theta} (\hat{\theta}) + (1 - \pi) \kappa_1 \frac{\partial \epsilon_1}{\partial \theta} (\hat{\theta}).
\] (6.6)

The value of \( \theta \) that minimizes the loss function is given by
\[
0 = \frac{\partial}{\partial \theta} L(D|\theta^*) = p\lambda_2 \frac{\partial \epsilon_2}{\partial \theta} (\theta^*) + (1 - p)\lambda_1 \frac{\partial \epsilon_1}{\partial \theta} (\theta^*) + \frac{\partial R}{\partial \theta} (D, \theta^*, \lambda).
\] (6.7)

Denote the value of \( \theta \) that minimizes the expected cost by \( \hat{\theta} \), and the value that minimizes the Loss function by \( \hat{\theta} + 1/n \delta \theta \). Now differentiation and Taylor expansion yield the sufficient condition for the minimum of the loss function to approach \( \hat{\theta} \) as \( R(\theta)/n \to 0 \):
\[
\lambda_2^B = \frac{\pi}{1 - \pi} \frac{1 - p \kappa_2}{p \kappa_1} \lambda_1.
\] (6.8)

This expression defines what we refer to as the “balancing cost”, \( \lambda_2^B \). Furthermore,
\[
\delta \theta = -\frac{\partial R}{\partial \theta} \left[ p\lambda_2 \frac{\partial^2 \epsilon_2}{\partial \theta^2} + (1 - p)\lambda_1 \frac{\partial^2 \epsilon_1}{\partial \theta^2} \right]^{-1},
\] (6.9)

with the right-hand side evaluated at \( \hat{\theta} \), which provides a first-order correction for finite datasets.

Figure 6.1 displays the above results graphically and can serve as a guideline of what weights to assign to peptides of different classes given the class distribution in the training set and the relative importance of positives versus negatives in the real-world application.

6.3.2 Cost adjustments

Seven training sets for each MHCI allele studied were generated, such that all training sets for a given allele had the same number of observations but varying proportions
Figure 6.1: Theoretical relationship between the training false negative cost ($\lambda_2$) that minimizes the expected cost of a classifier, $\mathcal{E}(\theta)$, for a given type 2 error cost ($\kappa_2$). The dotted lines represent one standard deviation from the mean.

of positives, namely 5%, 10%, 25%, 50%, 75%, 90% and 95%. These training sets were created as follows. First, 25% of the binders and 25% of the non-binders were randomly selected and set aside as a testing set. The remaining 75% of the binders and of the non-binders formed the “training superset”, from which the peptides for the various training sets were sampled. The total number of peptides in each of the seven training sets was fixed and equal to the number of peptides of the minority class in the training superset. The minority class was the positive for all tested alleles. Finally, the training sets were formed by randomly sampling without replacement positive and negative peptides from the training superset such that the described
class distribution was reached. The numbers of binders and non-binders in the resulting training sets are shown Table 6.1.

Table 6.1: Number of binders (B) and non-binders (NB) in training sets.

<table>
<thead>
<tr>
<th>%pos</th>
<th>A0203</th>
<th>A1101</th>
<th>A3101</th>
<th>B0702</th>
<th>B1501</th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td>24</td>
<td>456</td>
<td>26</td>
<td>495</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>48</td>
<td>432</td>
<td>52</td>
<td>469</td>
<td>32</td>
</tr>
<tr>
<td>25</td>
<td>120</td>
<td>360</td>
<td>130</td>
<td>391</td>
<td>80</td>
</tr>
<tr>
<td>50</td>
<td>240</td>
<td>240</td>
<td>261</td>
<td>261</td>
<td>160</td>
</tr>
<tr>
<td>75</td>
<td>360</td>
<td>120</td>
<td>391</td>
<td>130</td>
<td>240</td>
</tr>
<tr>
<td>90</td>
<td>432</td>
<td>47</td>
<td>469</td>
<td>52</td>
<td>288</td>
</tr>
<tr>
<td>95</td>
<td>456</td>
<td>24</td>
<td>495</td>
<td>26</td>
<td>304</td>
</tr>
</tbody>
</table>

The goal of this set of experiments was two-fold: 1) to investigate the relationship between class distribution and classifier performance, and 2) to learn how misclassification costs can be used to improve prediction accuracy for a given class distribution of the training set. We emphasize that our goal is not to improve upon existing computational methods, but rather to show that the performance of a single classifier can be improved with the use of cost-sensitive techniques. Misclassification costs were used as weights with the purpose of artificially changing the class distribution of the training dataset. The false negative cost \(\lambda_2\) can be interpreted as the weight given to the peptides in the positive class, and similarly false positive cost \(\lambda_1\) is the weight given to the negative class. The overall scale of the loss function Equation (6.5) is arbitrary, so we have fixed \(\lambda_1 = 1\) and varied \(\lambda_2\) between 1/20 and 20 in order to investigate the relationship between costs and class distribution.

Weiss and Provost (2003) have suggested that among the best class distributions for learning is the balanced distribution, one in which all classes are equally represented. We assume that given an unbalanced training set, a balancing misclassification cost can be used to achieve an artificially balanced class distribution. The
balancing cost $\lambda^B_2$, defined in Equation (6.8) can be interpreted to be the $\lambda_2$ that weights the positive peptides to be the same number as the negatives and therefore compensates for the imbalance ratio of the two classes. Consider the simplest scenario, where $\kappa_1 = 1$, $\kappa_2 = 1$ and $\pi = 0.5$, then the balancing cost reduces to

$$\lambda^B_2 = \frac{(1 - p)}{p}$$

We are particularly interested in how classifiers trained with this simplified balancing cost perform compared to the best classifiers for a given allele, as well as compared with classifiers trained with unit costs ($\lambda_1 = 1$ and $\lambda_2 = 1$).

6.3.3 Under-sampling

The under-sampling method consists of randomly eliminating peptides of the majority class from the training set until both classes have the same number of examples. The training sets were constructed in a similar manner to the cost-modifying experiment. First, 25% of binders and non-binders were set aside into the testing set. The remaining binders were put into the training set together with the same number of non-binders, which were randomly sampled without replacement from the non-binders training super set. One of the issues concerning under-sampling is the loss of information that results from the process, which can be aggravated when particularly important elements are removed from the training set. To get around this problem, we used 10-fold crossvalidation and the results presented here are the average of the 10 experiments.

6.3.4 Decision trees

The present study applies tree-based models to the peptide-MHCI binding prediction problem. We have chosen to use decision trees for the simplicity in their interpretation and also because they have not been thoroughly explored in the context of
peptide-MHCI binding. Moreover, decision and classification trees have become the canonical method for comparison of techniques used to deal with unbalanced datasets in the machine learning community.

Tree generation

Breiman (1993) provides an excellent and detailed description of classification and regression trees. Briefly, given a dataset in which each object is represented by a \((\tau, \mathbf{x})\) pair, where \(\mathbf{x}\) is a vector containing attributes of the object and \(c\) is a binary response variable, a tree-based classifier recursively partitions the data’s attribute space into sub-regions, called nodes, in which the response variable is increasingly more homogeneous. These trees are created in two steps: (1) induction of a large tree; and (2) pruning of the large tree into gradually smaller subtrees (here we use the cost-complexity pruning of Breiman, 1993). Finally, one subtree must be chosen from the sequence of subtrees generated by the pruning process. In the present study, we chose the tree that minimizes the loss function Equation (6.5) when applied to the test set.

The construction of a tree requires (1) a set of splits, which are binary questions with mutually exclusive and exhaustive outcomes used to partition the data, where the questions are coined in terms of the attributes of the objects in the dataset; and (2) a split function used to quantify the goodness of a split, by measuring the change in the homogeneity of the response variable in the tree due to splitting a node into two subsets based on the given split.

Splits and split function

In the problem at hand, the training dataset consists of peptides \(\phi\), where \(\tau(\phi)\) is the class of the peptide (either binder or non-binder) and \(\mathbf{x}(\phi)\) is the linear sequence of amino acids of the peptide, with \(x_j\) being the \(j^{th}\) amino acid from the amino
terminal end of the peptide. The binary questions about the sequence of peptides can be phrased in several distinct ways, and each one of them generates a different class of splits, called motifs, that can be used in the construction of trees. We used motifs based on the anchor positions, which are represented by a single amino acid with a fixed position in the peptide. The amino acids, in turn, can be represented in one of two ways: 1) by the traditional amino acid single-letter code. For example, alanine is represented by “A”, arginine by “R” and so forth; and 2) by their physicochemical properties, namely molecular weight, hydropathicity, volume, isoelectric point, polarity, ability to form hydrogen bonds and chain type (aliphatic, aromatic) as previously shown by Ray and Kepler (2007).

The split function used was the misclassification cost which is related to Equation (6.5) and is given by

$$L(\phi|\theta) = \tau_+(\phi)c_-(\phi)\lambda_2 + \tau_-(\phi)c_+(\phi)\lambda_1.$$  

6.4 Datasets

The peptide binding data used to train and test the decision trees were obtained from a publicly available database published by Peters et al. (2006), where the peptide affinity to a particular MHCI molecule is measured by one of two assays, and classified as binder when its dissociation constant is less or equal to 500 nM, and non-binder otherwise. Decision trees were constructed for each one of 32 alleles in the dataset. The cost-sensitive and under-sampling experiments described above were performed for only 5 alleles: A0203, A1101, A3101, B0702 and B1501. The numbers of peptides in the datasets for these five alleles are shown in Table 6.2.
Table 6.2: Number of binders and non-binders in Peters et al. (2006) datasets for 5 alleles.

<table>
<thead>
<tr>
<th>Allele</th>
<th>Binders</th>
<th>Non-binders</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0203</td>
<td>639</td>
<td>804</td>
</tr>
<tr>
<td>A1101</td>
<td>695</td>
<td>1290</td>
</tr>
<tr>
<td>A3101</td>
<td>427</td>
<td>1442</td>
</tr>
<tr>
<td>B0702</td>
<td>210</td>
<td>1052</td>
</tr>
<tr>
<td>B1501</td>
<td>179</td>
<td>799</td>
</tr>
</tbody>
</table>

6.5 Results

6.5.1 Cost adjustments

The first goal of this set of experiments was to investigate the relationship between class distribution and classifier performance. The results suggest that for a fixed training set size, decision trees perform best when trained with datasets of nearly balanced class distribution. Figure 6.2 shows the performance of classifiers trained with datasets of the same size but different class distributions and training costs for alleles A1101 and B0702 (results for other alleles are provided in Table E.1). Note that as the proportion of positives in the training set increases, the false negative rate decreases and the false positive rate increases as can be seen by the subtle shift in the curves from left to right.

Our second goal was to determine whether or not prediction accuracy of a given classifier can be improved by the use of cost-sensitive techniques and, if so, to establish the relationship between classifier performance and training costs. Our results demonstrate that misclassification costs can be used to improve prediction accuracy. In fact, for each one of the alleles we tested there was a cost $\lambda_2$ that performed significantly better than the unit cost, as can be seen by the increase in AUC shown in Table 6.3 for the five alleles, and in Table E.1 for all 35 tested alleles. Although our goal is not to improve upon the performance of existing methods, we also show in Table 6.3 and Table E.1 the AUC for four other methods as described in Peters.
Figure 6.2: Comparison of the performance of classifiers built with training sets of same size but different proportions of positives for alleles A1101 and B0702. Each point in a curve represents a classifier constructed with a different false negative training cost. The classifier constructed with the unit cost ($\lambda_1 = 1$) in each curve is marked with a solid circle and that constructed with the balancing cost is marked with a star. The curve for the perfect classifier would lie on the dotted line. The $y$-axis shows the total error rate of a classifier, which is the same as the classifier cost ($K$) when the type 1 and type 2 misclassification costs are identical ($\kappa_1 = \kappa_2 = 1$). FNR: false negative rate. FPR: false positive rate.

et al. (2006) for purposes of comparison.

Note in Figure 6.2 that for the training sets enriched for non-binders, $\lambda_2^B$ consistently reduced the total error rate as compared to the unit cost ($\lambda_2 = 1$). The impact of $\lambda_2^B$ on the performance of classifiers trained with binders-enriched datasets was not consistent, being better than unit cost for some classifiers and worse for others. In addition to representing an improvement over the unit cost, in a few cases $\lambda_2^B$ coincided with the minimizing cost, that is, the most accurate classifier for a given
allele and training set was the one trained with $\lambda_2^B$. However, in most cases, the balancing cost over-compensated for the imbalance in the class distribution, such that it was larger than the minimizing cost (see Figure 6.3).

We then compared the performance of trees trained with the complete dataset using either the unit cost or $\lambda_2^B$ (the red and green ROC curves in Figure 6.4, respectively). The use of $\lambda_2^B$ resulted in AUC at least as large as those for unit cost, such that $\lambda_2^B$ improved the ROC curves as compared to the unit cost in the majority of the cases. One interesting feature of the use of $\lambda_2^B$ is that it consistently shifts the ROC curve toward increasing sensitivity at the price of decreasing specificity, which is a desirable tradeoff when binders are rare. Thus, even in the cases when the increase in AUC is not substantial, the use of $\lambda_2^B$ can still represent an improvement over unit cost due to the shift it causes to the ROC curve.

6.5.2 Under-sampling

The results obtained using the balanced under-sampled training sets did not represent an improvement over those using the complete unbalanced training sets (see Figure 6.4 and Figure E.2). For alleles A0203, A1101 and B0702, the ROC curves for the trees trained with the entire dataset and those trained with the under-sampled dataset were indistinguishable from one another, whereas for alleles A3101 and B1501, the use of under-sampling severely damaged the accuracy of the trees.

6.5.3 Real-world costs versus training costs

We built decision trees with the training data described in Table 6.2 using different values of false negative cost ($\lambda_2$), and evaluated them on a test set using the “real-world” cost $\kappa_2$. We call $\hat{\lambda}_2$ the training cost that minimizes the total cost of a classifier on the test set for a given $\kappa_2$. Figure 6.5 shows the relationship between $\hat{\lambda}_2$
Table 6.3: AUC values for different classifiers. The second and third columns correspond to the decision trees described in the present work. Note the improvement in the performance of the trees with the use of training costs. *Values extracted from table 2 of Peters et al. (2006).

<table>
<thead>
<tr>
<th>allele</th>
<th>Trees, unit $\lambda_2$</th>
<th>Trees, best $\lambda_2$</th>
<th>ARB*</th>
<th>SMM*</th>
<th>ANN*</th>
<th>External Tool*</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0203</td>
<td>.746</td>
<td>.781</td>
<td>.884</td>
<td>.916</td>
<td>.921</td>
<td>.788</td>
</tr>
<tr>
<td>A1101</td>
<td>.841</td>
<td>.859</td>
<td>.918</td>
<td>.948</td>
<td>.951</td>
<td>.869</td>
</tr>
<tr>
<td>A3101</td>
<td>.825</td>
<td>.833</td>
<td>.909</td>
<td>.930</td>
<td>.928</td>
<td>.829</td>
</tr>
<tr>
<td>B0702</td>
<td>.887</td>
<td>.911</td>
<td>.952</td>
<td>.964</td>
<td>.965</td>
<td>.942</td>
</tr>
<tr>
<td>B1501</td>
<td>.759</td>
<td>.823</td>
<td>.900</td>
<td>.952</td>
<td>.941</td>
<td>.816</td>
</tr>
</tbody>
</table>

Figure 6.3: Comparison of balancing cost (solid black line) and the minimizing costs (symbols) for each one of the five alleles.
Figure 6.4: ROC curves for alleles A1101 (left panel) and B0702 (right panel) comparing the results of trees constructed with the oversampled training set (black curve), the under-sampled training set (red curve), and the full training set without training costs, that is, $\lambda_1 = \lambda_2 = 1$ (green curve) and with the balancing training cost, that is, $\lambda_1 = 1$ and $\lambda_2 = (1-p)/p$ (blue curve). The ROC curves were constructed by varying the threshold used to label a node from 0 to 1 and evaluating its sensitivity and specificity at each threshold.

and $\kappa_2$. Note that although the results are relatively noisy, in general the same trend shown in theory can be observed from this empirical data (see Figure 6.1). The $\lambda_2$ increases with $\kappa_2$ and as the proportion of positives in the training set increases, the line shifts to the right, indicating that for a particular value of $\kappa_2$, the suggested $\lambda_2$ decreases.

6.6 Discussion

Prediction of peptide-MHCI binding is of great importance to the development of subunit vaccines, as it can accelerate and reduce the cost of this process. One of the issues concerning the prediction of MHC-peptide binding is that binders are much less abundant than non-binders, and thus much harder to find experimentally. This
circumstance typically leads to highly unbalanced training sets, which can hinder the performance of algorithms trained with them. In fact, such training sets lead to significant increase of type 2 errors and thus more difficult still to find binders.

Our results show that highly unbalanced training sets do indeed reduce the accuracy of predictions made with decision trees and that these predictions improve as the training sets become more balanced. We have examined two approaches that aim at improving classifier accuracy by compensating for the imbalance in the class distribution of the training sets: under-sampling and a cost-sensitive method. Overall, under-sampling did not improve the performance of the decision trees. In fact,
in several cases classifiers trained with under-sampled training sets performed much worse than those trained with the full dataset. This could have been caused by the loss of information relevant to the training process. For this reason, under-sampling methods may only be appropriately used with datasets in which the majority class contains a lot of redundancy, in which circumstance under-sampling has been shown to outperform other random re-sampling methods in four distinct datasets Drummond and Holte (2003). Another potential drawback of under-sampling, and in broader terms of random re-sampling methods, is that they may yield noisy results due to the variability introduced in the process by the randomness of the sampling procedure.

In contrast to under-sampling, using misclassification costs as a means to artificially counterbalance data bias led to significant improvements in the performance of the decision trees in the majority of the cases. Although cost-sensitive procedures do not add any extra information to the training set, they seem to be more advantageous than random re-sampling techniques because they do not lead to the loss of information as does under-sampling and do not have the extra variability introduced by the random sampling process. Several other studies have shown cost-modifying methods to be advantageous. For example, Japkowicz and Stephen Japkowicz and Stephen (2002) performed a systematic comparison of these methods in both artificially-generated and real-world domains, showing that cost-modifying methods yield better results than re-sampling techniques. In addition to showing that peptide-MHCI binding predictions can be improved by the use cost-sensitive decision trees, we have investigated the use of the balancing cost, $\lambda_B^2$, as a rule-of-thumb to train classifiers. We have showed that although $\lambda_B^2$ is not always the $\lambda_2$ that minimizes the total cost of the classifier, it consistently outperforms the unit cost ($\lambda_2 = \lambda_1$) when the training set in enriched for non-binders. Moreover, we have showed that the use of $\lambda_B^2$ shifts the ROC curves towards areas of higher sensitivity.
in relation to ROC curves generated with unit cost, which can be highly desirable in situations such as epitope discovery projects.

Thus, although the relationship between training costs and class imbalance is relatively noisy, and further studies should be conducted before a complete guideline of what training costs should be used for a particular peptide-MHCI binding dataset, our results allow us to suggest that a balancing cost should be used for datasets enriched for non-binders, and the unit cost should be used for binders-enriched training sets.
Appendix A

Data description

The dendritic cell gene expression dataset was generated by Dr. Feng Feng, a post-doctoral researcher in the Kepler laboratory. Feng has written and kindly provided the description of the experimental procedures described below.

A.1 Mice

C57BL/6 male mice were obtained from Jackson Laboratories. Mice were maintained at the animal facility of Duke University and used at 8-10 weeks of age. All animals were housed and handled according to the Duke University’s Institutional Animal Care and Use Committees (IACUC) approved protocols.

A.2 Mouse spleen-derived dendritic cells

Spleen-derived DCs were isolated from C57BL/6 mice essentially as described in Björck (2001) and Maraskovsky et al. (1996). Mice were mobilized with recombinant Flt3L (Chinese hamster ovary cell derived) at a concentration of 10 µg/mouse per day in phosphate-buffered saline (PBS) by intra-peritoneal injection for nine consecutive
days under an IACUC-approved protocol. Spleens were isolated, and a single-cell suspension was made by being treated with 100U/mL collagenase (Worthington, Lakewoods, NJ) and passing the spleen through a nylon cell strainer (Falcon; Becton Dickinson, Franklin Lakes, NJ). After lysis of red cells by ammonium chloride solution, CD11c (Miltenyi Biotec) beads were added according to the manufacturer’s instructions. After incubation for 20 minutes at 4C, cells were washed and passed over a MACS column. Positively selected cells were isolated and suspended in appropriate buffer. Purity was checked routinely by FACS and was found to be greater than 97% (not shown). CD11c+ DCs were plated on non-tissue culture-treated plastic in complete DMEM medium containing 10% (vol/vol) FBS (HyClone Laboratories), 2 mM L-glutamine, 100 IU/mL of penicillin and 100 g/mL of streptomycin (all from Cellgro, Mediatech). DCs were stimulated for various times with high-purity LPS (1 ug/ml; Invitrogen) or oCpG (1uM/ml, Midland, TX).

A.3 Microarray analysis and data preparation

Total RNA was isolated with RNeasy Micro Kit (Qiagen) and overall quality was analyzed with an Agilent 2100 Bioanalyzer. Sample mRNA was amplified and labeled with the Affymetrix One-Cycle Eukaryotic Target Labeling Assay protocol and reagents. Biotinylated cRNA was hybridized to an Affymetrix GeneChip Mouse Genome 430 2.0 array with standard protocols and reagents from Affymetrix. Probe intensities were measured with the Affymetrix GeneChip Scanner 3000 and were processed into image analysis (.CEL) files with Affymetrix GeneChip operating software. Probe intensities were processed into CEL files using the Affymetrix GeneChip Operating Software. Probe intensities were background- adjusted, normalized, and probeset-summarized using the Robust Multi-chip Average (RMA) method using the software Bioconductor.
Appendix B

Gene ontology term lists

B.1 Transcription factors

The list of transcription factors was obtained via the gene ontology annotations provided by Affymetrix. Probesets associated with the biological process containing the string “transcription” and molecular function containing the string “DNA binding” were retrieved into the transcription factor list. As such, this list contains probesets 2886 and represents a superset of molecules that function as transcription factors.

B.2 Cytokines

The list of cytokines was obtained by the union of two lists. The first list was obtained via the gene ontology annotations provided by Affymetrix. A total of 64 probesets associated with the molecular functions “cytokine activity” (term GO:0005125) were retrieved into this list. We then obtained a second list by searching for the same GO term in the Mouse Genome Informatics Gene Ontology Browser (http://www.informatics.jax.org) and were able to increase the set of probesets by 215 items. The final list of cytokines used here contains 269 probesets.
Appendix C

Flatness tests additional results

C.1 Treatment effect parameters

Table 4.6 shows there is a four-fold increase in the number of probe sets with a nonzero treatment effect in the CpG experiment in comparison to the same number in the LPS experiment. Here we provide some additional information about these two sets of genes, comparing some of the model parameters and the changes in description length between the null treatment model and the treatment model. Recall that under the null treatment model,

\[ y_{ijkt} = \xi(t) + \epsilon_{ijkt}, \]

and under the not null treatment model,

\[ y_{ijkt} = \xi(t) + \delta_{k1} \tau(t) + \epsilon_{ijkt}, \]

where \( \xi \) and \( \tau \) are Langevin processes characterized by \( (\gamma_{\xi}, \zeta_{\xi}) \) and \( (\gamma_{\tau}, \zeta_{\tau}) \) respectively. Figure C.1 contrasts the parameters of the treatment effect model for the probe sets that were classified as differentially expressed under each treatment. Figure C.2 contrasts the error variance, the change in DL, and the magnitude of the
treatment effect for the probe sets that were classified as differentially expressed under each treatment. Although the distributions of the parameters are very similar between the two experiments, there are few differences that provide some insight into the data. Remarkably, the distribution of $\zeta$ for the CpG treatment is shifted to the left in relation to that for LPS treatment. This indicates that the control time series for the CpG experiment are in general flatter than those for the LPS experiment. Notice also that the magnitude of the treatment effect is generally larger under CpG stimulation than under LPS treatment.

### C.2 Cytokines

Table C.1: Summary of results of treatment flatness tests for cytokines classified as not flat for both LPS and CpG treatments.

<table>
<thead>
<tr>
<th>Probe ID</th>
<th>Gene name</th>
<th>LPS</th>
<th>CpG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$DL_{treat}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$DL_{treat}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range</td>
<td>Range</td>
</tr>
<tr>
<td>1417266_at</td>
<td>Ccl6</td>
<td>17.87</td>
<td>6.16</td>
</tr>
<tr>
<td>1418219_at</td>
<td>Il15</td>
<td>17.67</td>
<td>9.65</td>
</tr>
<tr>
<td>1418345_at</td>
<td>Tnfsf13</td>
<td>14.23</td>
<td>12.36</td>
</tr>
<tr>
<td>1418930_at</td>
<td>Cxcl10</td>
<td>18.45</td>
<td>21.69</td>
</tr>
<tr>
<td>1419697_at</td>
<td>Cxcl11</td>
<td>26.11</td>
<td>26.77</td>
</tr>
<tr>
<td>1419698_at</td>
<td>Cxcl11</td>
<td>22.02</td>
<td>21.30</td>
</tr>
<tr>
<td>1420249_s_at</td>
<td>Ccl6</td>
<td>20.24</td>
<td>11.68</td>
</tr>
<tr>
<td>1422305_at</td>
<td>Ifnb1</td>
<td>27.41</td>
<td>31.10</td>
</tr>
<tr>
<td>1422408_at</td>
<td>Ifna4</td>
<td>6.85</td>
<td>15.82</td>
</tr>
<tr>
<td>1423635_at</td>
<td>Bmp2</td>
<td>8.21</td>
<td>12.69</td>
</tr>
<tr>
<td>1427429_at</td>
<td>Csf2</td>
<td>15.62</td>
<td>15.86</td>
</tr>
<tr>
<td>1459913_at</td>
<td>Tnfsf10</td>
<td>11.24</td>
<td>16.46</td>
</tr>
</tbody>
</table>
Figure C.1: Histograms of (A) $\gamma_\zeta$, (B) $\xi_\zeta$, (C) $\gamma_\tau$, and (D) $\zeta_\tau$, for the treatment effect model for all probes classified as differentially expressed (i.e., all probes for which DL_{t+}^{\text{treat}} < DL_{t-}^{\text{treat}}).
Table C.2: Summary of results of treatment flatness tests for cytokines classified as not flat for LPS treatment and as flat for CpG treatment.

<table>
<thead>
<tr>
<th>Probe ID</th>
<th>Gene name</th>
<th>LPS $DL_{treat}$</th>
<th>LPS $DL_{+}$</th>
<th>Range</th>
<th>CpG $DL_{treat}$</th>
<th>CpG $DL_{+}$</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1417932_at</td>
<td>Il18</td>
<td>13.15</td>
<td>7.79</td>
<td>2.59</td>
<td>2.59</td>
<td>6.30</td>
<td>0.00</td>
</tr>
<tr>
<td>1418652_at</td>
<td>Cxcl9</td>
<td>21.33</td>
<td>20.02</td>
<td>1.81</td>
<td>19.80</td>
<td>21.16</td>
<td>1.66</td>
</tr>
<tr>
<td>1421207_at</td>
<td>Lif</td>
<td>9.79</td>
<td>9.68</td>
<td>1.85</td>
<td>13.90</td>
<td>15.20</td>
<td>3.17</td>
</tr>
<tr>
<td>1423792_a_at</td>
<td>Cmtm6</td>
<td>-0.35</td>
<td>-1.69</td>
<td>1.77</td>
<td>0.05</td>
<td>3.67</td>
<td>0.17</td>
</tr>
<tr>
<td>1449288_at</td>
<td>Gdf3</td>
<td>13.08</td>
<td>12.70</td>
<td>1.75</td>
<td>9.22</td>
<td>11.30</td>
<td>0.84</td>
</tr>
<tr>
<td>1452543_a_at</td>
<td>Scgb1a1</td>
<td>7.58</td>
<td>4.96</td>
<td>1.43</td>
<td>13.25</td>
<td>14.57</td>
<td>1.57</td>
</tr>
</tbody>
</table>
Figure C.2: Histograms of (A) the error variance ($\sigma^2$), (B) the improvement in DL of the treatment effect model over the null effect model ($\Delta DL$), and (C) the magnitude of the treatment effect ($\Delta T$). The improvement in DL is defined as $\Delta DL = DL_{treat}^+ - DL_{treat}^-$, where $DL_{treat}^+$ and $DL_{treat}^-$ are described in Section 4.5.3. The magnitude of the treatment effect is $\Delta T = \max_t (s_{treat}(t)) - \min_t (s_{treat}(t))$, where $s_{treat}(t)$ is defined in Equation (4.41).
Table C.3: Summary of results of treatment flatness tests for cytokines classified as not flat for CpG treatment and as flat for LPS treatment.

<table>
<thead>
<tr>
<th>Probe ID</th>
<th>Gene name</th>
<th>LPS $DL_{treat}$</th>
<th>LPS $DL_{treat}$</th>
<th>Range</th>
<th>CpG $DL_{treat}$</th>
<th>CpG $DL_{treat}$</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1415855_at</td>
<td>Kitl</td>
<td>-0.64</td>
<td>3.14</td>
<td>0.00</td>
<td>6.49</td>
<td>5.10</td>
<td>2.37</td>
</tr>
<tr>
<td>1416176_at</td>
<td>Hmgb1</td>
<td>-11.77</td>
<td>-8.50</td>
<td>0.40</td>
<td>-6.07</td>
<td>-7.88</td>
<td>0.94</td>
</tr>
<tr>
<td>1417391_a_at</td>
<td>Il16</td>
<td>13.33</td>
<td>15.94</td>
<td>0.81</td>
<td>12.40</td>
<td>10.68</td>
<td>4.13</td>
</tr>
<tr>
<td>1417574_at</td>
<td>Cxcl12</td>
<td>10.46</td>
<td>14.16</td>
<td>0.17</td>
<td>7.67</td>
<td>7.23</td>
<td>1.18</td>
</tr>
<tr>
<td>1417936_at</td>
<td>Ccl9</td>
<td>3.02</td>
<td>8.31</td>
<td>1.89</td>
<td>12.20</td>
<td>5.70</td>
<td>2.96</td>
</tr>
<tr>
<td>1418097_a_at</td>
<td>Tpte2</td>
<td>-1.60</td>
<td>2.17</td>
<td>0.00</td>
<td>-6.43</td>
<td>-6.83</td>
<td>0.47</td>
</tr>
<tr>
<td>1418126_at</td>
<td>Ccl5</td>
<td>12.79</td>
<td>16.03</td>
<td>1.06</td>
<td>-1.01</td>
<td>-7.24</td>
<td>1.42</td>
</tr>
<tr>
<td>1418480_at</td>
<td>Cxcl7</td>
<td>-2.78</td>
<td>1.00</td>
<td>0.00</td>
<td>11.96</td>
<td>7.79</td>
<td>2.39</td>
</tr>
<tr>
<td>1418718_at</td>
<td>Cxcl16</td>
<td>2.18</td>
<td>4.87</td>
<td>2.09</td>
<td>1.31</td>
<td>-0.81</td>
<td>2.15</td>
</tr>
<tr>
<td>1418949_at</td>
<td>Gdf15</td>
<td>16.42</td>
<td>18.17</td>
<td>1.67</td>
<td>10.08</td>
<td>9.51</td>
<td>2.41</td>
</tr>
<tr>
<td>1419282_at</td>
<td>Ccl12</td>
<td>22.05</td>
<td>22.22</td>
<td>5.51</td>
<td>19.61</td>
<td>17.84</td>
<td>2.67</td>
</tr>
<tr>
<td>1419412_at</td>
<td>Xcl1</td>
<td>13.56</td>
<td>17.33</td>
<td>0.00</td>
<td>15.62</td>
<td>12.15</td>
<td>2.40</td>
</tr>
<tr>
<td>1419530_at</td>
<td>Il12b</td>
<td>14.72</td>
<td>18.50</td>
<td>0.00</td>
<td>19.69</td>
<td>18.75</td>
<td>3.87</td>
</tr>
<tr>
<td>1419561_at</td>
<td>Ccl3</td>
<td>13.23</td>
<td>16.64</td>
<td>0.72</td>
<td>16.55</td>
<td>16.31</td>
<td>2.96</td>
</tr>
<tr>
<td>1420353_at</td>
<td>Lta</td>
<td>16.94</td>
<td>20.51</td>
<td>0.65</td>
<td>28.46</td>
<td>20.48</td>
<td>5.00</td>
</tr>
<tr>
<td>1420412_at</td>
<td>Tnfsf10</td>
<td>16.44</td>
<td>18.95</td>
<td>1.31</td>
<td>15.08</td>
<td>12.40</td>
<td>2.09</td>
</tr>
<tr>
<td>1421134_at</td>
<td>Areg</td>
<td>24.97</td>
<td>29.20</td>
<td>0.94</td>
<td>20.77</td>
<td>18.89</td>
<td>6.06</td>
</tr>
<tr>
<td>1421578_at</td>
<td>Ccl4</td>
<td>14.01</td>
<td>17.54</td>
<td>1.00</td>
<td>26.05</td>
<td>25.43</td>
<td>4.52</td>
</tr>
<tr>
<td>1421588_at</td>
<td>Tnfsf14</td>
<td>11.62</td>
<td>15.40</td>
<td>0.00</td>
<td>27.42</td>
<td>22.05</td>
<td>6.83</td>
</tr>
<tr>
<td>1421688_a_at</td>
<td>Ccl1</td>
<td>16.06</td>
<td>19.84</td>
<td>0.01</td>
<td>29.72</td>
<td>21.93</td>
<td>4.98</td>
</tr>
<tr>
<td>1422080_at</td>
<td>Il7</td>
<td>10.46</td>
<td>12.66</td>
<td>0.77</td>
<td>13.35</td>
<td>12.76</td>
<td>1.87</td>
</tr>
<tr>
<td>1422406_at</td>
<td>Ifna9</td>
<td>-0.29</td>
<td>3.44</td>
<td>0.11</td>
<td>14.85</td>
<td>12.12</td>
<td>3.93</td>
</tr>
<tr>
<td>1422924_at</td>
<td>Tnfsf9</td>
<td>16.44</td>
<td>20.22</td>
<td>0.00</td>
<td>17.85</td>
<td>17.02</td>
<td>3.41</td>
</tr>
<tr>
<td>1423028_at</td>
<td>Ifna2</td>
<td>15.54</td>
<td>15.71</td>
<td>2.45</td>
<td>33.35</td>
<td>20.85</td>
<td>8.85</td>
</tr>
<tr>
<td>1425377_at</td>
<td>Wnt1</td>
<td>3.47</td>
<td>7.25</td>
<td>0.00</td>
<td>26.93</td>
<td>18.03</td>
<td>5.15</td>
</tr>
<tr>
<td>1425947_at</td>
<td>Ifng</td>
<td>20.73</td>
<td>21.61</td>
<td>2.36</td>
<td>26.89</td>
<td>18.40</td>
<td>4.44</td>
</tr>
<tr>
<td>1425958_at</td>
<td>Il1f9</td>
<td>11.98</td>
<td>15.66</td>
<td>0.27</td>
<td>19.42</td>
<td>15.12</td>
<td>5.10</td>
</tr>
<tr>
<td>1426152_a_at</td>
<td>Kitl</td>
<td>6.79</td>
<td>10.56</td>
<td>0.00</td>
<td>8.93</td>
<td>6.72</td>
<td>2.74</td>
</tr>
<tr>
<td>1427624_s_at</td>
<td>Il22/Iltifb</td>
<td>18.14</td>
<td>20.15</td>
<td>2.12</td>
<td>22.56</td>
<td>18.26</td>
<td>3.11</td>
</tr>
<tr>
<td>1427964_at</td>
<td>Cmtm8</td>
<td>5.91</td>
<td>6.40</td>
<td>1.59</td>
<td>8.11</td>
<td>5.12</td>
<td>2.17</td>
</tr>
<tr>
<td>1428934_at</td>
<td>Cmtm2b</td>
<td>11.98</td>
<td>15.76</td>
<td>0.00</td>
<td>3.85</td>
<td>1.64</td>
<td>1.38</td>
</tr>
<tr>
<td>1437271_at</td>
<td>Clcf1</td>
<td>-0.23</td>
<td>3.54</td>
<td>0.00</td>
<td>12.93</td>
<td>11.72</td>
<td>3.55</td>
</tr>
<tr>
<td>1439680_at</td>
<td>Tnfsf10</td>
<td>15.43</td>
<td>16.33</td>
<td>1.83</td>
<td>17.15</td>
<td>8.77</td>
<td>3.14</td>
</tr>
<tr>
<td>144117_at</td>
<td>Kitl</td>
<td>8.83</td>
<td>12.61</td>
<td>0.00</td>
<td>12.76</td>
<td>12.61</td>
<td>2.09</td>
</tr>
<tr>
<td>1448148_at</td>
<td>Grn</td>
<td>-7.64</td>
<td>-4.59</td>
<td>0.15</td>
<td>-13.53</td>
<td>-15.35</td>
<td>0.68</td>
</tr>
<tr>
<td>1448316_at</td>
<td>Cmtm3</td>
<td>-8.22</td>
<td>-6.87</td>
<td>0.40</td>
<td>5.79</td>
<td>0.04</td>
<td>1.40</td>
</tr>
<tr>
<td>1448818_at</td>
<td>Wnt5a</td>
<td>10.04</td>
<td>13.82</td>
<td>0.00</td>
<td>10.25</td>
<td>9.83</td>
<td>2.40</td>
</tr>
</tbody>
</table>

174
### C.3 Transcription factors

Table C.4: Summary of results of treatment flatness tests for transcription factors classified as not flat for both LPS and CpG treatments.

<table>
<thead>
<tr>
<th>Probe ID</th>
<th>Gene name</th>
<th>LPS</th>
<th></th>
<th></th>
<th>CpG</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DL\textsubscript{treat}</td>
<td>DL\textsubscript{treat}</td>
<td>Range</td>
<td>DL\textsubscript{treat}</td>
<td>DL\textsubscript{treat}</td>
<td>Range</td>
</tr>
<tr>
<td>1415999_at</td>
<td>Hey1</td>
<td>12.67</td>
<td>8.06</td>
<td>2.01</td>
<td>12.76</td>
<td>8.88</td>
<td>1.81</td>
</tr>
<tr>
<td>1416751_a_at</td>
<td>Ddx20</td>
<td>-10.31</td>
<td>-11.67</td>
<td>0.99</td>
<td>-3.46</td>
<td>-5.16</td>
<td>1.44</td>
</tr>
<tr>
<td>1417244_a_at</td>
<td>Irf7</td>
<td>26.81</td>
<td>20.07</td>
<td>4.57</td>
<td>21.64</td>
<td>12.68</td>
<td>3.97</td>
</tr>
<tr>
<td>1417961_a_at</td>
<td>Trim30</td>
<td>15.26</td>
<td>10.39</td>
<td>3.66</td>
<td>13.79</td>
<td>9.69</td>
<td>3.22</td>
</tr>
<tr>
<td>1418982_at</td>
<td>Cebpa</td>
<td>18.06</td>
<td>12.34</td>
<td>2.69</td>
<td>13.80</td>
<td>9.69</td>
<td>3.22</td>
</tr>
<tr>
<td>1420715_a_at</td>
<td>Pparg</td>
<td>21.61</td>
<td>14.36</td>
<td>3.41</td>
<td>10.40</td>
<td>9.48</td>
<td>1.01</td>
</tr>
<tr>
<td>1420865_at</td>
<td>Zfp161</td>
<td>-2.90</td>
<td>-2.91</td>
<td>0.71</td>
<td>4.85</td>
<td>-2.27</td>
<td>1.15</td>
</tr>
<tr>
<td>1420915_at</td>
<td>Stat1</td>
<td>5.33</td>
<td>5.03</td>
<td>0.22</td>
<td>1.46</td>
<td>-0.44</td>
<td>1.29</td>
</tr>
<tr>
<td>1421079_at</td>
<td>Nra43</td>
<td>8.19</td>
<td>6.08</td>
<td>1.75</td>
<td>13.81</td>
<td>9.69</td>
<td>3.22</td>
</tr>
<tr>
<td>1421322_a_at</td>
<td>Isgf3g</td>
<td>7.06</td>
<td>6.03</td>
<td>1.05</td>
<td>1.00</td>
<td>-0.87</td>
<td>1.09</td>
</tr>
<tr>
<td>1421911_at</td>
<td>Stat2</td>
<td>14.26</td>
<td>7.35</td>
<td>2.70</td>
<td>15.37</td>
<td>11.17</td>
<td>3.97</td>
</tr>
<tr>
<td>1422675_at</td>
<td>Smarce1</td>
<td>-2.57</td>
<td>-5.15</td>
<td>0.72</td>
<td>3.40</td>
<td>-2.32</td>
<td>1.48</td>
</tr>
<tr>
<td>1423063_at</td>
<td>Dnmt3a</td>
<td>-1.68</td>
<td>-3.60</td>
<td>0.67</td>
<td>0.19</td>
<td>-2.24</td>
<td>0.84</td>
</tr>
<tr>
<td>1423066_at</td>
<td>Dnmt3a</td>
<td>0.78</td>
<td>0.53</td>
<td>0.69</td>
<td>-1.86</td>
<td>-4.10</td>
<td>0.74</td>
</tr>
<tr>
<td>Probe ID</td>
<td>Gene name</td>
<td>LPS</td>
<td>CpG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>---------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \text{DL}_{\text{treat}} )</td>
<td>( \text{DL}_{\text{treat}} )</td>
<td>( \text{DL}_{\text{treat}} )</td>
<td>( \text{Range} )</td>
<td>( \text{DL}_{\text{treat}} )</td>
<td>( \text{DL}_{\text{treat}} )</td>
</tr>
<tr>
<td>1423100_at</td>
<td>Fos</td>
<td>3.07</td>
<td>-0.96</td>
<td>0.83</td>
<td>6.30</td>
<td>1.37</td>
<td>1.05</td>
</tr>
<tr>
<td>1423773_at</td>
<td>Gpbp1</td>
<td>-9.70</td>
<td>-10.03</td>
<td>0.38</td>
<td>-6.28</td>
<td>-8.87</td>
<td>0.88</td>
</tr>
<tr>
<td>1424531_a_at</td>
<td>Tce3</td>
<td>8.26</td>
<td>7.20</td>
<td>2.12</td>
<td>10.44</td>
<td>10.34</td>
<td>1.53</td>
</tr>
<tr>
<td>1425405_a_at</td>
<td>Adar</td>
<td>12.63</td>
<td>6.78</td>
<td>1.98</td>
<td>12.85</td>
<td>10.00</td>
<td>2.64</td>
</tr>
<tr>
<td>1425464_a_at</td>
<td>Gata6</td>
<td>5.27</td>
<td>4.39</td>
<td>1.33</td>
<td>11.19</td>
<td>8.51</td>
<td>3.15</td>
</tr>
<tr>
<td>1427682_a_at</td>
<td>Egr2</td>
<td>2.27</td>
<td>1.83</td>
<td>1.48</td>
<td>11.35</td>
<td>11.26</td>
<td>2.83</td>
</tr>
<tr>
<td>1427713_x_at</td>
<td>Pou2f2</td>
<td>-0.66</td>
<td>-1.38</td>
<td>1.41</td>
<td>6.28</td>
<td>4.56</td>
<td>1.57</td>
</tr>
<tr>
<td>1429722_at</td>
<td>Zbtb4</td>
<td>-2.01</td>
<td>-2.66</td>
<td>0.83</td>
<td>0.04</td>
<td>-2.87</td>
<td>1.05</td>
</tr>
<tr>
<td>1431415_a_at</td>
<td>Tbp1</td>
<td>-6.50</td>
<td>-7.02</td>
<td>0.67</td>
<td>-4.90</td>
<td>-5.05</td>
<td>0.49</td>
</tr>
<tr>
<td>1431676_x_at</td>
<td>Gtf2i</td>
<td>6.95</td>
<td>4.62</td>
<td>1.78</td>
<td>5.79</td>
<td>5.66</td>
<td>1.03</td>
</tr>
<tr>
<td>1433512_at</td>
<td>Fli1</td>
<td>9.70</td>
<td>7.17</td>
<td>1.32</td>
<td>10.01</td>
<td>8.34</td>
<td>2.33</td>
</tr>
<tr>
<td>1434268_at</td>
<td>Adar</td>
<td>7.37</td>
<td>6.87</td>
<td>1.35</td>
<td>7.47</td>
<td>2.69</td>
<td>2.35</td>
</tr>
<tr>
<td>1434457_at</td>
<td>Sp100</td>
<td>18.29</td>
<td>16.22</td>
<td>2.86</td>
<td>14.68</td>
<td>9.60</td>
<td>2.88</td>
</tr>
<tr>
<td>1434786_at</td>
<td>Ppp1r12b</td>
<td>3.01</td>
<td>2.07</td>
<td>1.05</td>
<td>5.36</td>
<td>4.08</td>
<td>1.07</td>
</tr>
<tr>
<td>1435168_at</td>
<td>Zfp710</td>
<td>10.59</td>
<td>7.59</td>
<td>1.50</td>
<td>10.96</td>
<td>7.10</td>
<td>2.26</td>
</tr>
<tr>
<td>1437484_at</td>
<td>Zbtb5</td>
<td>-0.34</td>
<td>-1.95</td>
<td>1.39</td>
<td>8.94</td>
<td>4.71</td>
<td>2.39</td>
</tr>
<tr>
<td>1439842_at</td>
<td>Zfp287</td>
<td>-5.46</td>
<td>-5.50</td>
<td>0.83</td>
<td>7.45</td>
<td>5.83</td>
<td>2.10</td>
</tr>
<tr>
<td>1440481_at</td>
<td>Stat1</td>
<td>15.72</td>
<td>15.31</td>
<td>2.41</td>
<td>12.16</td>
<td>9.04</td>
<td>2.80</td>
</tr>
<tr>
<td>1440878_at</td>
<td>Runx1</td>
<td>1.49</td>
<td>0.44</td>
<td>0.99</td>
<td>4.59</td>
<td>2.93</td>
<td>1.09</td>
</tr>
<tr>
<td>1447213_at</td>
<td>Sp100</td>
<td>16.67</td>
<td>16.02</td>
<td>3.00</td>
<td>11.74</td>
<td>11.52</td>
<td>2.52</td>
</tr>
<tr>
<td>1448757_at</td>
<td>Pml</td>
<td>14.72</td>
<td>14.47</td>
<td>2.07</td>
<td>12.32</td>
<td>3.41</td>
<td>2.39</td>
</tr>
<tr>
<td>1449266_at</td>
<td>MeCP2</td>
<td>-4.84</td>
<td>-5.10</td>
<td>0.47</td>
<td>2.51</td>
<td>2.43</td>
<td>1.82</td>
</tr>
<tr>
<td>1450033_a_at</td>
<td>Stat1</td>
<td>12.78</td>
<td>12.39</td>
<td>1.83</td>
<td>9.81</td>
<td>2.69</td>
<td>2.38</td>
</tr>
<tr>
<td>1450403_at</td>
<td>Stat2</td>
<td>15.30</td>
<td>8.85</td>
<td>2.59</td>
<td>13.48</td>
<td>4.37</td>
<td>2.99</td>
</tr>
<tr>
<td>1450889_at</td>
<td>HlTF</td>
<td>-3.66</td>
<td>-4.08</td>
<td>0.68</td>
<td>1.59</td>
<td>-2.52</td>
<td>1.21</td>
</tr>
<tr>
<td>1451685_at</td>
<td>Mllt6</td>
<td>4.72</td>
<td>3.93</td>
<td>0.60</td>
<td>9.30</td>
<td>8.33</td>
<td>1.66</td>
</tr>
<tr>
<td>1451716_at</td>
<td>Mafb</td>
<td>22.17</td>
<td>17.15</td>
<td>3.68</td>
<td>11.53</td>
<td>4.78</td>
<td>2.44</td>
</tr>
<tr>
<td>1451821_a_at</td>
<td>Sp100</td>
<td>13.83</td>
<td>12.97</td>
<td>1.82</td>
<td>9.77</td>
<td>5.51</td>
<td>1.96</td>
</tr>
<tr>
<td>1451860_a_at</td>
<td>Trim30</td>
<td>5.59</td>
<td>0.95</td>
<td>1.84</td>
<td>0.83</td>
<td>-6.88</td>
<td>1.31</td>
</tr>
<tr>
<td>1453256_at</td>
<td>Polr3c</td>
<td>-6.55</td>
<td>-6.67</td>
<td>0.65</td>
<td>-0.51</td>
<td>-3.10</td>
<td>0.98</td>
</tr>
<tr>
<td>1453372_at</td>
<td>Dnajc1</td>
<td>-6.68</td>
<td>-6.77</td>
<td>0.80</td>
<td>6.04</td>
<td>3.52</td>
<td>1.38</td>
</tr>
<tr>
<td>1454773_at</td>
<td>Rxra</td>
<td>8.98</td>
<td>8.63</td>
<td>1.28</td>
<td>5.95</td>
<td>0.87</td>
<td>1.01</td>
</tr>
<tr>
<td>1455667_at</td>
<td>Preb</td>
<td>0.24</td>
<td>-1.51</td>
<td>0.85</td>
<td>7.59</td>
<td>5.44</td>
<td>2.01</td>
</tr>
<tr>
<td>1456494_a_at</td>
<td>Trim30</td>
<td>17.85</td>
<td>11.58</td>
<td>4.41</td>
<td>15.73</td>
<td>1.79</td>
<td>2.73</td>
</tr>
<tr>
<td>1458802_at</td>
<td>Hivep3</td>
<td>6.16</td>
<td>5.37</td>
<td>2.41</td>
<td>9.36</td>
<td>8.87</td>
<td>1.65</td>
</tr>
<tr>
<td>1460038_at</td>
<td>Pou3f1</td>
<td>10.89</td>
<td>7.25</td>
<td>3.39</td>
<td>21.11</td>
<td>16.75</td>
<td>5.22</td>
</tr>
</tbody>
</table>

Table C.5: Summary of results of treatment flatness tests for transcription factors classified as not flat for LPS and as flat for CpG treatments.
Table C.6: Summary of results of treatment flatness tests for transcription factors classified as flat for LPS treatment and as not flat for CpG treatment. For brevity, here we only show 50 out of the 367 TFs not flat in CpG. These are the 50 TFs with highest values of $|DL_{treat}^{LPS} - DL_{treat}^{CpG}|$.

| Probe ID     | Gene name | LPS | CpG | | | | |
|--------------|-----------|-----|-----| | | | |
| 1418544_at   | Kcnip3    | 11.25 | 10.89 | 2.74 | 15.39 | 18.35 | 0.79 |
| 1420945_at   | Atrx      | 2.73  | 0.96 | 1.99 | 10.10 | 12.61 | 0.85 |
| 1421027_a_at | Mef2c     | 7.25  | 7.22 | 0.54 | 8.00  | 11.71 | 0.00 |
| 1422566_at   | Tcefb     | 5.41  | 4.59 | 1.06 | 3.76  | 6.77  | 0.69 |
| 1423064_at   | Dmmt3a    | 3.06  | 2.05 | 0.95 | -5.21 | -4.59 | 0.50 |
| 1423319_at   | Hhex      | 7.95  | 7.28 | 0.96 | 7.60  | 7.77  | 1.25 |
| 1423579_a_at | Trpc4     | 2.21  | -0.35 | 1.62 | -4.68 | -0.97 | 0.00 |
| 1424852_at   | Mef2c     | 7.96  | 7.42 | 0.81 | 6.50  | 8.90  | 1.94 |
| 1425874_at   | Hoxc13    | 7.02  | 5.59 | 2.61 | 8.06  | 11.77 | 0.00 |
| 1426007_at   | Sox17     | 9.37  | 8.61 | 1.93 | 15.87 | 16.90 | 1.63 |
| 1426314_at   | Scy12     | -6.27 | -6.38 | 0.47 | 7.14  | 10.85 | 0.00 |
| 1428444_at   | Atp2c1    | 17.73 | 8.18 | 2.50 | -10.90 | -9.49 | 0.22 |
| 1429177_at   | Hoxc13    | 7.02  | 5.59 | 2.61 | 8.06  | 11.77 | 0.00 |
| 1434786_at   | Ppp1r12b  | 3.01  | 2.07 | 1.05 | 7.21  | 10.93 | 0.00 |
| 1435478_at   | Wdr26     | -0.11 | -0.21 | 0.67 | 4.39  | 8.51  | 1.43 |
| 1436097_at   | Arghap9   | 11.33 | 9.99 | 1.45 | 0.53  | 2.29  | 0.94 |
| 1436314_at   | Scy12     | -6.27 | -6.38 | 0.47 | 7.14  | 10.85 | 0.00 |
| 1438475_at   | AB124611  | 17.73 | 8.18 | 2.50 | -10.90 | -9.49 | 0.22 |
| 1442742_at   | Atp2c1    | 6.06  | 4.54 | 1.41 | 9.14  | 12.39 | 0.49 |
| 1447946_at   | Adam23    | 7.59  | 5.72 | 1.93 | 2.25  | 5.08  | 1.30 |
| 1449143_at   | Rtp4      | 21.11 | 15.4 | 3.78 | 8.87  | 9.35  | 1.62 |
| 1449455_at   | Hck       | 4.57  | 4.22 | 2.59 | -1.65 | -0.27 | 0.54 |
| 1450454_at   | Tor3a     | 9.86  | 9.12 | 1.70 | 3.69  | 7.40  | 0.00 |
| 1451716_at   | Mafb      | 22.17 | 17.15 | 3.68 | 11.25 | 14.08 | 1.69 |
| 1454837_at   | Chn6      | 5.13  | 2.81 | 1.16 | -4.63 | -3.44 | 0.80 |
| 1455500_at   | D11Erd759e| 12.48 | 10.47 | 1.93 | -8.64 | -6.72 | 0.33 |
| 1456150_at   | A630082K20Rik| 8.59 | 7.08 | 1.33 | 10.17 | 12.11 | 1.09 |
| 1456223_at   | C77713    | 1.34  | 1.10 | 1.48 | 8.10  | 11.21 | 0.64 |
| 1457585_at   | Tnem24    | 17.33 | 16.99 | 3.75 | 2.58  | 4.16  | 0.74 |
| 1459868_at   | Il11ra1   | 1.00  | 0.66 | 0.92 | 8.13  | 11.84 | 0.00 |
| 1459900_at   | C79468    | 6.53  | 5.06 | 1.63 | 6.88  | 10.67 | 1.12 |
| 1460242_at   | Cd55      | 11.67 | 5.49 | 3.20 | -4.91 | -4.12 | 0.89 |
| 1460334_at   | Dbnl      | -2.91 | -4.31 | 1.07 | 2.30  | 3.28  | 0.83 |

<p>| Probe ID | Gene name | LPS | CpG | | | | |
|----------|-----------|-----|-----| | | | |
| 1416019_at | Dr1       | 7.29 | 0.04 | 1.70 | -12.22 | -10.02 | 0.25 |
| 1416029_at | Klf10     | 5.29 | -4.85 | 1.36 | -8.62 | -7.75 | 0.68 |
| 1416302_at | Ebf1      | 7.85 | 3.21 | 2.16 | 1.65  | -3.75 | 0.03 |
| 1416916_at | Elf3      | 16.07 | 11.26 | 2.71 | 8.17  | 14.47 | 2.08 |
| 1416957_at | Pou2af1   | 1.38 | -2.55 | 0.80 | -1.73 | 2.04 | 0.00 |
| 1417000_at | Abtb1     | 7.29 | 3.22 | 1.22 | 1.52  | 5.28  | 0.06 |
| 1417165_at | Mbd2      | -3.25 | -8.74 | 0.78 | -10.91 | -7.41 | 0.28 |
| 1417394_at | Klf4      | 7.77 | 1.97 | 1.23 | -2.33 | 1.44  | 0.00 |
| 1417395_at | Klf4      | 9.65 | 5.68 | 1.23 | -0.96 | 3.06  | 0.58 |
| 1417517_at | Plag2     | 2.02 | -3.47 | 1.11 | -2.57 | 1.55  | 0.12 |
| 1417519_at | Plag2     | 7.05 | 2.55 | 1.55 | 1.67  | 4.22  | 1.62 |</p>
<table>
<thead>
<tr>
<th>Gene</th>
<th>Expression</th>
<th>Z-score</th>
<th>p-value</th>
<th>Adj. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mybl2</td>
<td>13.26</td>
<td>8.68</td>
<td>2.31</td>
<td>-7.93</td>
</tr>
<tr>
<td>Mtf2</td>
<td>-6.06</td>
<td>-10.65</td>
<td>1.15</td>
<td>-14.60</td>
</tr>
<tr>
<td>Batf</td>
<td>23.24</td>
<td>14.83</td>
<td>4.04</td>
<td>10.54</td>
</tr>
<tr>
<td>Ehf</td>
<td>18.19</td>
<td>11.30</td>
<td>2.74</td>
<td>11.32</td>
</tr>
<tr>
<td>Zbtb16</td>
<td>14.28</td>
<td>4.36</td>
<td>2.19</td>
<td>10.65</td>
</tr>
<tr>
<td>Pegf2</td>
<td>0.98</td>
<td>-2.64</td>
<td>0.94</td>
<td>-3.23</td>
</tr>
<tr>
<td>Tcf2</td>
<td>24.72</td>
<td>19.91</td>
<td>3.93</td>
<td>8.89</td>
</tr>
<tr>
<td>Mta3</td>
<td>8.67</td>
<td>4.68</td>
<td>2.13</td>
<td>3.44</td>
</tr>
<tr>
<td>Nkx1-2</td>
<td>17.22</td>
<td>13.66</td>
<td>2.68</td>
<td>9.36</td>
</tr>
<tr>
<td>Pias1</td>
<td>-0.38</td>
<td>-4.56</td>
<td>0.77</td>
<td>-3.42</td>
</tr>
<tr>
<td>Tpb1g</td>
<td>13.16</td>
<td>8.89</td>
<td>3.47</td>
<td>1.93</td>
</tr>
<tr>
<td>Creh1</td>
<td>3.14</td>
<td>-7.90</td>
<td>2.03</td>
<td>3.22</td>
</tr>
<tr>
<td>Myst4</td>
<td>2.52</td>
<td>-1.29</td>
<td>1.46</td>
<td>-3.00</td>
</tr>
<tr>
<td>Grhl1</td>
<td>17.24</td>
<td>12.80</td>
<td>4.40</td>
<td>-2.41</td>
</tr>
<tr>
<td>Zbtb1</td>
<td>-4.11</td>
<td>-8.90</td>
<td>0.85</td>
<td>-12.07</td>
</tr>
<tr>
<td>Mbd2</td>
<td>7.95</td>
<td>0.11</td>
<td>1.78</td>
<td>-2.35</td>
</tr>
<tr>
<td>Nrl1d1</td>
<td>4.10</td>
<td>-1.86</td>
<td>1.75</td>
<td>-7.67</td>
</tr>
<tr>
<td>Jarid1c</td>
<td>2.18</td>
<td>-4.52</td>
<td>1.41</td>
<td>2.35</td>
</tr>
<tr>
<td>Jarid1b</td>
<td>0.65</td>
<td>-2.92</td>
<td>1.24</td>
<td>-2.10</td>
</tr>
<tr>
<td>Tshz1</td>
<td>8.77</td>
<td>4.09</td>
<td>1.35</td>
<td>5.91</td>
</tr>
<tr>
<td>Tshz1</td>
<td>7.22</td>
<td>1.02</td>
<td>1.51</td>
<td>4.07</td>
</tr>
<tr>
<td>Csde1</td>
<td>-10.50</td>
<td>-18.06</td>
<td>0.57</td>
<td>-20.05</td>
</tr>
<tr>
<td>Nrkb1</td>
<td>6.63</td>
<td>1.32</td>
<td>1.36</td>
<td>4.47</td>
</tr>
<tr>
<td>Cypf2</td>
<td>5.03</td>
<td>1.15</td>
<td>1.72</td>
<td>7.42</td>
</tr>
<tr>
<td>Btbd11</td>
<td>7.24</td>
<td>3.02</td>
<td>1.77</td>
<td>0.21</td>
</tr>
<tr>
<td>Bmyc</td>
<td>12.15</td>
<td>8.05</td>
<td>2.29</td>
<td>3.79</td>
</tr>
<tr>
<td>D1Ertd161e</td>
<td>3.04</td>
<td>-1.16</td>
<td>1.26</td>
<td>-5.52</td>
</tr>
<tr>
<td>1110051B16Rik</td>
<td>22.77</td>
<td>17.66</td>
<td>5.65</td>
<td>20.80</td>
</tr>
<tr>
<td>Nrkb2</td>
<td>5.00</td>
<td>0.41</td>
<td>2.56</td>
<td>4.31</td>
</tr>
<tr>
<td>Nfkc1</td>
<td>4.11</td>
<td>-2.76</td>
<td>1.20</td>
<td>-6.61</td>
</tr>
<tr>
<td>Tigd2</td>
<td>2.68</td>
<td>-1.28</td>
<td>1.83</td>
<td>-2.25</td>
</tr>
<tr>
<td>Nfkbie</td>
<td>0.96</td>
<td>-3.20</td>
<td>1.81</td>
<td>2.62</td>
</tr>
<tr>
<td>Zbtb32</td>
<td>14.73</td>
<td>4.24</td>
<td>2.82</td>
<td>13.48</td>
</tr>
<tr>
<td>Ches1</td>
<td>-3.30</td>
<td>-11.36</td>
<td>0.66</td>
<td>-16.86</td>
</tr>
<tr>
<td>Zfp692</td>
<td>1.94</td>
<td>-1.65</td>
<td>1.58</td>
<td>-6.15</td>
</tr>
<tr>
<td>Trps1</td>
<td>4.16</td>
<td>-3.44</td>
<td>1.18</td>
<td>-9.62</td>
</tr>
<tr>
<td>Mxd4</td>
<td>12.21</td>
<td>8.08</td>
<td>2.37</td>
<td>6.01</td>
</tr>
<tr>
<td>Mxd4</td>
<td>14.02</td>
<td>8.89</td>
<td>2.27</td>
<td>5.92</td>
</tr>
<tr>
<td>Phf2</td>
<td>-3.87</td>
<td>-8.48</td>
<td>0.66</td>
<td>-10.10</td>
</tr>
<tr>
<td>Nfic</td>
<td>3.05</td>
<td>-1.73</td>
<td>1.05</td>
<td>-4.86</td>
</tr>
<tr>
<td>ZBTB45</td>
<td>5.29</td>
<td>-1.45</td>
<td>2.18</td>
<td>3.98</td>
</tr>
<tr>
<td>Nr3c2</td>
<td>15.19</td>
<td>8.55</td>
<td>3.38</td>
<td>7.22</td>
</tr>
<tr>
<td>E2f7</td>
<td>2.88</td>
<td>-1.07</td>
<td>1.58</td>
<td>0.81</td>
</tr>
<tr>
<td>Fosl2</td>
<td>4.32</td>
<td>-4.09</td>
<td>1.21</td>
<td>-13.18</td>
</tr>
<tr>
<td>Fem1c</td>
<td>0.09</td>
<td>-3.71</td>
<td>0.85</td>
<td>-5.79</td>
</tr>
<tr>
<td>Zbtb1</td>
<td>5.20</td>
<td>0.47</td>
<td>1.50</td>
<td>2.74</td>
</tr>
<tr>
<td>Mill5</td>
<td>1.57</td>
<td>-2.00</td>
<td>1.47</td>
<td>2.83</td>
</tr>
<tr>
<td>Zbtb16</td>
<td>18.13</td>
<td>5.15</td>
<td>3.76</td>
<td>12.43</td>
</tr>
<tr>
<td>Edf1</td>
<td>-7.29</td>
<td>-11.50</td>
<td>0.67</td>
<td>-13.18</td>
</tr>
<tr>
<td>Gene Symbol</td>
<td>Gene Name</td>
<td>At Position</td>
<td>Value 1</td>
<td>Value 2</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>1442963_at</td>
<td>Zhx3</td>
<td></td>
<td>10.53</td>
<td>3.75</td>
</tr>
<tr>
<td>1445710_x_at</td>
<td>1110051B16Rik</td>
<td></td>
<td>26.94</td>
<td>19.17</td>
</tr>
<tr>
<td>1446846_at</td>
<td>Patz1</td>
<td></td>
<td>8.92</td>
<td>-0.06</td>
</tr>
<tr>
<td>1447958_at</td>
<td>Patz1</td>
<td></td>
<td>10.75</td>
<td>6.22</td>
</tr>
<tr>
<td>1449306_at</td>
<td>Hsf2</td>
<td></td>
<td>13.68</td>
<td>8.04</td>
</tr>
<tr>
<td>1449868_at</td>
<td>Tbx6</td>
<td></td>
<td>17.53</td>
<td>13.43</td>
</tr>
<tr>
<td>1450034_at</td>
<td>Stat1</td>
<td></td>
<td>2.93</td>
<td>-2.01</td>
</tr>
<tr>
<td>1451375_at</td>
<td>Ehf</td>
<td></td>
<td>17.58</td>
<td>12.76</td>
</tr>
<tr>
<td>1451687_a_at</td>
<td>Tcf2</td>
<td></td>
<td>41.00</td>
<td>30.12</td>
</tr>
<tr>
<td>1451899_a_at</td>
<td>Gtf2ird1</td>
<td></td>
<td>1.71</td>
<td>-2.30</td>
</tr>
<tr>
<td>1452246_at</td>
<td>Ostf1</td>
<td></td>
<td>1.83</td>
<td>-4.56</td>
</tr>
<tr>
<td>1456010_x_at</td>
<td>Hes5</td>
<td></td>
<td>16.64</td>
<td>12.22</td>
</tr>
<tr>
<td>1456103_at</td>
<td>Pml</td>
<td></td>
<td>15.24</td>
<td>9.86</td>
</tr>
<tr>
<td>1457404_at</td>
<td>Nfkbiz</td>
<td></td>
<td>9.04</td>
<td>4.85</td>
</tr>
<tr>
<td>1458299_s_at</td>
<td>Nfkbie</td>
<td></td>
<td>-3.31</td>
<td>-8.92</td>
</tr>
<tr>
<td>1459838_s_at</td>
<td>Btbd11</td>
<td></td>
<td>6.66</td>
<td>-2.88</td>
</tr>
<tr>
<td>1460006_at</td>
<td>Atbfi1</td>
<td></td>
<td>1.92</td>
<td>-1.82</td>
</tr>
<tr>
<td>1460228_at</td>
<td>Usf2</td>
<td></td>
<td>9.85</td>
<td>4.33</td>
</tr>
<tr>
<td>1460559_at</td>
<td>Ankrd25</td>
<td></td>
<td>7.16</td>
<td>1.37</td>
</tr>
<tr>
<td>AFFX-r2-Bs-dap-3_at</td>
<td></td>
<td></td>
<td>16.69</td>
<td>13.09</td>
</tr>
</tbody>
</table>
Appendix D

Clustering additional results

Figure D.1: Pairwise posterior probability matrix (PPPM) for Langevin data computed based on the last 800 iterations outputted by the Gibbs sampler restricted to the observations belonging to the true Cluster 3.
Figure D.2: Trace of number of clusters over the MCMC chain for clustering transcription factors under LPS stimulation.

Figure D.3: Pairwise posterior probability matrix (PPPM) for the transcription factors differentially expressed in response to LPS. These probabilities were computed based on the last 2000 iterations outputted by the Gibbs sampler.
Appendix E

Prediction of peptide-MHCI binding additional results

Table E.1: AUC values for different classifiers. The second and third columns correspond to the decision trees described in the present work. Note the improvement in the performance of the trees with the use of training costs. *Values extracted from table 2 of Peters et al. (2006).

<table>
<thead>
<tr>
<th>allele</th>
<th>Trees, unit $\lambda_2$</th>
<th>Trees, best $\lambda_2$</th>
<th>ARB*</th>
<th>SMM*</th>
<th>ANN*</th>
<th>External Tool*</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0101</td>
<td>.903</td>
<td>.951</td>
<td>.964</td>
<td>.980</td>
<td>.982</td>
<td>.955</td>
</tr>
<tr>
<td>A0201</td>
<td>.796</td>
<td>.842</td>
<td>.934</td>
<td>.952</td>
<td>.957</td>
<td>.922</td>
</tr>
<tr>
<td>A0202</td>
<td>.770</td>
<td>.770</td>
<td>.875</td>
<td>.899</td>
<td>.900</td>
<td>.793</td>
</tr>
<tr>
<td>A0203</td>
<td>.746</td>
<td>.781</td>
<td>.884</td>
<td>.916</td>
<td>.921</td>
<td>.788</td>
</tr>
<tr>
<td>A0206</td>
<td>.732</td>
<td>.747</td>
<td>.872</td>
<td>.914</td>
<td>.927</td>
<td>.735</td>
</tr>
<tr>
<td>A0301</td>
<td>.763</td>
<td>.763</td>
<td>.908</td>
<td>.940</td>
<td>.937</td>
<td>.851</td>
</tr>
<tr>
<td>A1101</td>
<td>.841</td>
<td>.859</td>
<td>.918</td>
<td>.948</td>
<td>.951</td>
<td>.869</td>
</tr>
<tr>
<td>A2301</td>
<td>.742</td>
<td>.782</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>A2402</td>
<td>.685</td>
<td>.748</td>
<td>.718</td>
<td>.780</td>
<td>.825</td>
<td>.770</td>
</tr>
<tr>
<td>A2403</td>
<td>.673</td>
<td>.846</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>A2601</td>
<td>.606</td>
<td>.811</td>
<td>.907</td>
<td>.931</td>
<td>.956</td>
<td>.736</td>
</tr>
<tr>
<td>A2902</td>
<td>.783</td>
<td>.847</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>A3001</td>
<td>.741</td>
<td>.861</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>A3002</td>
<td>.777</td>
<td>.810</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>A3101</td>
<td>.825</td>
<td>.833</td>
<td>.909</td>
<td>.930</td>
<td>.928</td>
<td>.829</td>
</tr>
<tr>
<td>A3301</td>
<td>.636</td>
<td>.827</td>
<td>.892</td>
<td>.925</td>
<td>.915</td>
<td>.807</td>
</tr>
<tr>
<td>Code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>A6801</td>
<td>.756</td>
<td>.761</td>
<td>.840</td>
<td>.885</td>
<td>.883</td>
<td>.772</td>
</tr>
<tr>
<td>A6802</td>
<td>.699</td>
<td>.714</td>
<td>.865</td>
<td>.898</td>
<td>.899</td>
<td>.643</td>
</tr>
<tr>
<td>A6901</td>
<td>.614</td>
<td>.813</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>B0702</td>
<td>.887</td>
<td>.911</td>
<td>.952</td>
<td>.964</td>
<td>.965</td>
<td>.942</td>
</tr>
<tr>
<td>B0801</td>
<td>.547</td>
<td>.835</td>
<td>.936</td>
<td>.943</td>
<td>.955</td>
<td>.766</td>
</tr>
<tr>
<td>B1501</td>
<td>.759</td>
<td>.823</td>
<td>.900</td>
<td>.952</td>
<td>.941</td>
<td>.816</td>
</tr>
<tr>
<td>B1801</td>
<td>.745</td>
<td>.833</td>
<td>.573</td>
<td>.853</td>
<td>.838</td>
<td>.779</td>
</tr>
<tr>
<td>B2705</td>
<td>.753</td>
<td>.892</td>
<td>.915</td>
<td>.940</td>
<td>.938</td>
<td>.926</td>
</tr>
<tr>
<td>B3501</td>
<td>.712</td>
<td>.771</td>
<td>.851</td>
<td>.889</td>
<td>.875</td>
<td>.792</td>
</tr>
<tr>
<td>B4001</td>
<td>.587</td>
<td>.897</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>B4002</td>
<td>.718</td>
<td>.778</td>
<td>.541</td>
<td>.842</td>
<td>.754</td>
<td>.775</td>
</tr>
<tr>
<td>B4402</td>
<td>.588</td>
<td>.762</td>
<td>.533</td>
<td>.740</td>
<td>.778</td>
<td>.783</td>
</tr>
<tr>
<td>B4403</td>
<td>.647</td>
<td>.804</td>
<td>.461</td>
<td>.770</td>
<td>.763</td>
<td>.698</td>
</tr>
<tr>
<td>B4501</td>
<td>.679</td>
<td>.824</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>B5101</td>
<td>.664</td>
<td>.792</td>
<td>.822</td>
<td>.868</td>
<td>.866</td>
<td>.820</td>
</tr>
<tr>
<td>B5301</td>
<td>.795</td>
<td>.811</td>
<td>.871</td>
<td>.882</td>
<td>.899</td>
<td>.861</td>
</tr>
<tr>
<td>B5401</td>
<td>.654</td>
<td>.796</td>
<td>.847</td>
<td>.921</td>
<td>.903</td>
<td>.799</td>
</tr>
<tr>
<td>B5701</td>
<td>.756</td>
<td>.936</td>
<td>.428</td>
<td>.871</td>
<td>.826</td>
<td>.767</td>
</tr>
<tr>
<td>B5801</td>
<td>.815</td>
<td>.864</td>
<td>.899</td>
<td>.964</td>
<td>.961</td>
<td>.899</td>
</tr>
</tbody>
</table>
Figure E.1: Comparison of the performance of classifiers built with training sets of same size but different proportions of positives for alleles A0203, A3101 and B1501. Compare to Figure 6.2. Each point in a curve represents a classifier constructed with a different false negative training cost. The classifier constructed with the unit cost ($\lambda_1 = 1$) in each curve is marked with a solid circle and that constructed with the balancing cost is marked with a star. The curve for the perfect classifier would lie on the dotted line. The y-axis shows the total error rate of a classifier, which is the same as the classifier cost ($K$) when the type 1 and type 2 misclassification costs are identical ($\kappa_1 = \kappa_2 = 1$). FNR: false negative rate. FPR: false positive rate.
Figure E.2: ROC curves for alleles (A) A0203, (B) A3101, and (C) B1501 comparing the results of trees constructed with the oversampled training set (black curve), the undersampled training set (red curve), and the full training set without training costs, that is, $\lambda_1 = \lambda_2 = 1$ (green curve) and with the balancing training cost, that is, $\lambda_1 = 1$ and $\lambda_2 = (1 - p)/p$ (blue curve). Compare to figure 4. The ROC curves were constructed by varying the threshold used to label a node from 0 to 1 and evaluating its sensitivity and specificity at each threshold.
Bibliography


Biography


In August of 2004, Ana Paula enrolled at Duke University in the Bioinformatics and Genome Technologies Ph.D. program, which became the Program in Computational Biology and Bioinformatics. In 2008, she published an article entitled “Improving peptide-MHC class I binding prediction for unbalanced datasets” (Sales et al., 2008). While at Duke, she has completed the requirements for, and successfully defended, her Master of Science degree in Statistical Science on May 19, 2011. She will receive this degree concurrently with her Ph.D. in Computational Biology and Bioinformatics in September of 2011.

She has accepted a position as a Postdoctoral Researcher in the Applied Statistics Group at Lawrence Livermore National Laboratory in Livermore, California.