OPTIMAL MODEL-BASED APPROACHES FOR PREDICTIVE INFERENCE IN BIOLOGY

A Dissertation

by

JASON MATTHEW KNIGHT

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Edward Dougherty
Co-Chair of Committee,	Ivan Ivanov
Committee Members,	Jean-Francois Chamberland
	Robert Chapkin
Head of Department,	Miroslav Begovic

May 2015

Major Subject: Electrical Engineering

ABSTRACT

Predictive modeling of the dynamic, multivariate, non-linear, stochastic systems of biology is a difficult enterprise. High throughput measurement techniques are enabling new approaches to computational biology, but the small number of samples typically available relative to the number of features measured make additional sources of information critical for accurate predictions. In this dissertation, we offer an approach to incorporate biological pathway knowledge into a predictive stochastic model for genetic regulatory networks. In addition, we propose a statistical model for shotgun sequencing and use computational approximation strategies to derive optimal estimators for classification.

We perform comparisons of classifiers trained using this framework to other existing classification rules including non-linear support vector machines. Using both synthetic and real sequencing data, our classifiers delivered lower classification error rates than existing classification techniques. In addition, we demonstrate using prior knowledge to construct the classifier through properly constructed prior distributions and several scenarios where this increases classification performance.

This research establishes a flexible framework to generate optimal estimators with respect to statistical biological models. By demonstrating the role and power of computation in unlocking these estimators, we point future research efforts towards this computationally intensive approach for the computational biology field.

DEDICATION

To my family, both cis and trans.

ACKNOWLEDGEMENTS

I first would like to thank my parents Bud and Debbie Knight. Without their love and support in all its forms, I would not have had the freedom and courage to be who I am today. I also thank Bryan for his persistent flame of curiosity, Michelle for leading me away from hubris, and my beloved Emily for her enormous support and love in our journey and the patience which she will continue to need for the years to come!

I'd also like to thank my advisors: Ed Dougherty for teaching me to aspire to the higher planes of optimal estimation over ad-hockery and to having a sound epistemology; Ivan Ivanov for teaching me wisdom in navigating the often treacherous landscapes of science; and Robert Chapkin for showing me what first class science looks like and the importance of transparency in all things. I also thank Jean-Francois Chamberland for acting as my committee member and reinforcing rigor in all things.

The members of the GSP and Chapkin labs were crucial in raising my standards of quality and productivity, keeping things interesting, and always being there in support, in no particular order I'll list the most notable: Dr. Laurie Davidson, Jennifer Goldsby, Karen Triff, Dr. Tim Hou, Robert Fuentes, Eunjoo Kim, Dr. Roger Zoh, Dr. Manasvi Shah, Jason Xingde Jiang, Dr. Sriram Sridharan, Dr. Mohammad Shahrokh, Dr. Esmaeil Gargari, Dr. Ting Chen, Dr. Chen Zhou, Dr. Youting Sun, Dr. Mohammad Yousefi-Rezaei, Dr. Lori Dalton, Dr. Amin Zollanvari, and Dr. Fang Hsu. And lastly, I thank my friends and workout partners Edwin Eigenbrodt and Edward Talmage for waging war against physiological atrophy and hastening the heat death of the universe.

NOMENCLATURE

PSTG	Probabilistic State Transition Graph
OBC	Optimal Bayesian Classifier
BEE	Bayesian error estimate
MSEBEE	the estimated Mean Square Error of the Bayesian Error Estimator
MCMC	Markov chain Monte Carlo
NF-κB	Nuclear factor-κB

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Biological organisms can be considered high-dimensional, non-linear, stochastic, dynamical systems. Each of these attributes increases the difficulty of making predictions about the behavior of these complex machines. Additionally, the known operational mechanisms of biological systems are only partially catalogued adding another level of difficulty to the task. Despite this, disciplines ranging from medicine to synthetic biology drive the need to develop methodologies to make accurate predictions in the face of this difficult and partially understood domain.

Recent developments in the field of high-throughput biological measurement techniques – most notably shotgun sequencing – have increased our ability to probe the internal workings of the cell. These techniques produce measurements of the entire transcriptional profile of a cell, or sequence an entire genome. But despite rapidly dropping costs of these techniques, the number of samples obtained is typically much smaller than the number of features measured. This positions the analyses of these dataset squarely in the "small sample" domain where common statistical assumptions such as asymptotics cannot be relied upon.

Despite the difficulty of the domain and the small number of samples available, two facets of the problem, if properly leveraged, can enable forward progress. First, for decades biologists have worked to discover knowledge regarding the mechanistic underpinnings of biology in the form of pathways. These pathways describe known relationships between genes, proteins, RNAs, and other functional elements of the cell. Secondly, with the data and pathway knowledge available, we are typically not immediately interested in a full understanding of the biological system itself. Instead, we would often like to make predictions about some limited aspect of the

system. For example, medical practitioners often want answers to questions such as, "what subtype of cancer is my patient suffering from?" and "which treatment will lead to the best prognosis?". Similarly biologists ask targeted questions of their experimental datasets such as, "what mechanism is responsible for this observed shift in phenotype?" and "which experiment should I perform next to discover this mechanism?"

Leveraging those two insights, in this dissertation, we propose techniques to utilize biological pathway knowledge to make predictions about biological systems.

1.1 Contributions

The contributions made in this dissertation result from two primary research projects. First, we developed an algorithm to use biological pathways to construct a stochastic dynamical model for gene regulatory networks. This model can then be used for making predictions of system behavior under unobserved operating conditions. We then applied this technique to build an NF-κB regulatory model using pathways from the literature. We then used additional mouse knockout experiments available in the literature for an external qualitative validation. Secondly, we wished to improve this technique to develop an optimal estimation methodology for the incorporation of prior knowledge with high-throughput data. We achieved this in the realm of classification using Optimal Bayesian Classification and the application of computational approximation techniques.

1.1.1 Markov models for pathway knowledge

The cell is the essential functional unit of life, and an understanding of its internal mechanisms has occupied a large proportion of the productive output of biology. Biological pathways represent a formalization of much of this knowledge in the form of mechanistic dependencies between the functional elements in a cell.

Unfortunately, the information inherent in these pathways is incomplete and often conflicting due to cellular context including epigenetics and internal or environmental conditions for the cell. Additionally, the information is univariate and therefore does not typically provide enough information alone to make accurate predictions in light of the cellular context.

We address that problem in Section 2 by developing an algorithm which takes as an input a set of pathways and outputs a Markov chain model of the system which evolves consistently with the pathways. This requires a procedure to deal with inconsistencies in the pathway information that may arise due to differing cellular context when the pathway was originally discovered.

We then use this Markov chain to make predictions regarding the steady state distribution of the gene/protein expression. These steady state distributions can then be used to predict phenotypes based on known gene actions in the cell.

The NF-κB network is of prime interest in translational medicine and biology as it acts as a hub network of the cellular inflammatory response mechanism. Chronic inflammation is linked to many diseases such as autoimmune disorders, the progression of cancer, and heart disease. Therefore, it is of great interest in translational medicine to produce accurate predictions about the activation of NF-κB given a set of conditions (or treatments) surrounding the upstream signaling network.

We obtained 28 pathways from the NF-κB literature and transformed them into a binary vector valued Markov chain model of the NF-κB network. We then compared the predictions of this Markov chain under seven network perturbations to the literature where a seven analogous mice knockout models were performed. This qualitative validation of the network model demonstrates that pathways can encode enough information regarding the system when intelligently combined, and the Markov chain model maintains this information in a consistent manner enabling predictions under

perturbation which match biological observations under similar perturbations.

1.1.2 Optimal Bayesian classification for non-Gaussian sequencing datasets

While the approach taken in Section 2 was seen to produce acceptable predictions, it still remains essentially heuristic. In Section 3, by adopting a cost function and an optimization approach, we find optimal estimators for predictions of biological systems. Specifically, we consider the prediction problem of classification. In addition, instead of considering pathway information, we utilize a new statistical model (this time of the data generation process) in order to operate on labeled sample data and prior distributions to train an optimal classifier.

This work builds on previous work by Dalton and Dougherty [15, 16, 17, 18] where they discovered MMSE classifiers for Gaussian and multinomial distributions. However, we wished to apply these methodologies to sequencing data which is a widespread biological measurement technique and does not conform to Gaussian distributional assumptions. This required a statistical feature-label distribution with considerably more complexity than multivariate Gaussian or multinomial distributions alone. We therefore proposed a hierarchical Poisson model to encapsulate the known processes that sequencing data undergoes from the biology to the resulting measurements.

Utilizing a statistical model that aligns closely with the underlying measurement process provides several advantages over simpler phenomenological statistical models:

- Placing prior distributions over the parameters of the model is more straightforward as the parameters of the model relate to measurable, real world, quantities.

- The inferred parameters of the model are easier to interpret and troubleshoot should problems arise.

- The flexible construction of the model allows the addition or subtraction of complexity as the data warrant (or require) it.

The downsides to these complex models is the loss of analytical tractability. We therefore developed a computational approximation strategy to arrive at the optimal estimators using tools such as Markov chain Monte Carlo and Monte Carlo integration.

We validated the performance of these models and subsequent optimal classifiers on a variety of synthetic datasets against other classification techniques. We also used a real dataset from The Cancer Genome Atlas to classify subtypes of lung cancer sequencing data using the same set of available classifiers. The optimal classifier exhibited superior performance in nearly all cases.

We also applied the same statistical model in a feature selection study to detect groups of genes which well separate phenotypes of interest. In this capacity, we utilized two additional optimal estimators surrounding the prediction problem: the Bayesian error estimate, and the mean square error of the Bayesian error estimate. These two estimates give a salient measure of the separation of the phenotypes and a quantification of the uncertainty in that estimate.

We then applied the three estimators (including the optimal classifier itself) to a dietary animal model dataset to discover gene pairs and triplets that were not individually differentially expressed, but together well separated the groups with low error estimates and low uncertainties around those estimates. This led to novel biological insights to the system which were not available using widely available to the differential expression analysis techniques common in sequencing data analysis.

1.2 Organization

Section 2 introduces a novel algorithm to produce binary vector valued Markov chain models of genetic regulatory networks consistent with biological pathway knowledge. Section 3 explains an optimal estimation framework to use prior knowledge and data for sample classification. Section 4 extends the work in Section 3 for feature selection of sequencing datasets using an additional pair of optimal estimators. And Section 5 concludes the dissertation with summarizing remarks and future work.

2. FROM BIOLOGICAL PATHWAYS TO PREDICTIVE MODELS*

2.1 Introduction

Biological regulatory network models offer the promise of one day applying systems based approaches for cancer diagnosis and therapy [27, 21]. Consequently, it is not surprising that the systems biology literature contains many algorithms to infer such regulatory networks from (time-course) microarray data [46, 72, 24, 1, 45]. Inferring such networks is inherently difficult because of the limited availability of the data and the fact that most of these algorithms do not include mechanisms for incorporating prior knowledge, which could potentially reduce the data requirement. Consequently, most of these network models have not been validated, thereby hindering their use in translational science and medicine.

Before the advent of high throughput measurement techniques such as microarrays and shotgun sequencing, biological experimentation often focused on uncovering (mostly univariate) relationships between genes and proteins in the production of what is usually referred to as *pathway knowledge*. This pathway knowledge is based on empirical observations across different experiments that have acquired some degree of validity through the peer review process. While not all pathways in the literature are accurate, and some pathways may in fact be conflicting, we believe that they offer an excellent foundation for the network construction process especially when combined with high throughput data for model refinement and validation.

In the absence of any pathway knowledge, one would have to assume that each protein behaves randomly. In other words, with no knowledge of the interactions

*Parts of this section are reproduced with permission from Knight, J.M.; Datta, A.; Dougherty, E.R. "Generating Stochastic Gene Regulatory Networks Consistent with Pathway Information and Steady-State Behavior", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 59(6), 1701-1710 2012. doi:10.1109/TBME.2012.2192117

between proteins we cannot predict if a certain protein will be expressed more or less than average over the long run. With available pathway information, however, one can refine the random model by using the knowledge to guide the behavior of the model when the contextual information of the pathway is satisfied. By requiring the model to obey the pathway information, we can be sure that the model reflects the pathway knowledge that is available. By using a significant amount of pathway information we can reduce the data requirement and generate models that produce predictions that are more meaningful than those with little or no prior information.

Based on the preceding discussion, it is clear that the long run behavior of the model is a function of the amount of pathway information available about the system and the initial conditions. If we know too little about the system, then the long run behavior will reflect the unknown, random evolution and no conclusions can be made. However, if the long run behavior differs from the stochastic background level, then the pathway knowledge and initial conditions are sufficient to make qualitative predictions about that system. This is what we will be demonstrating in this section by using a model which captures the behavior of the pathways relating to the NF-κB system.

A few key assumptions underlie our model development. The first is the discrete state and discrete time approximation of protein behavior. This is a large assumption, but the most important one utilized in this section. It has been validated in many biological contexts [53, 22], and provides an important simplification that enables large scale network modeling with pathway data. Indeed, as pointed out in [63], such discrete-time discrete-state modeling avoids the need for making continuous-time measurements of protein concentrations and facilitates the accommodation of genes/proteins which exhibit ON/OFF switch-like behavior. Moreover, discrete time systems are easier to analyze, model and control in real time [60].

The second assumption that we make is that there is no prior knowledge about the initial state of the cell and only the presence or absence of external stimuli is known. In other words, we assign a uniform initial distribution to all possible states and allow the pathway constrained model to stabilize into attractor cycles. This assumption has the effect of distributing the resulting probability mass of the cell states to the states that are in the attractor cycles. Furthermore, these attractor cycles have a total probability that is proportional to the size of the basin of attraction. A similar conclusion was reached by Huang [49] but from a biological argument of cellular homeostasis in the presence of continual perturbations of the inter and intra cellular environments.

In this section, we propose a new method for generating networks from pathway information. We then apply this method to the set of proteins that compose the NF-κB regulatory network to build the transition probability matrix of a discrete-state, discrete-time Markov chain that produces predictions that agree with the literature. The NF-κB system was chosen due to the prevalence of associated pathway information in the literature as well as due to its biological importance in cancer and the innate immune system.

2.2 Obtaining the pathways

In any network inference procedure, the first step consists of selecting a specific biological system and choosing the specific agents for inclusion in the model. In this section, this task was performed manually, although future work could include using selection techniques such as statistical tests on high-throughput data to identify the most relevant molecules for state based modeling.

The model generated in this section consists of protein species with the exception of one lipoglycan (lipopolysaccharide). In general, the species in the pathways and

the resulting model can be a mixture of any types of biological entities as long as the pathways are accurately reflect the relationships being studied. Throughout this section, we will refer to the elements in the model as proteins with the tacit understanding that sometimes other biological entities would also be admissible.

To obtain pathway data for this section, we manually reviewed the biological literature relevant to the NF-κB system and recorded a pathway when significant biological evidence was available and the molecules involved in the pathway were chosen to be significant. For interactions that included non-significant species, the pathway was either ignored or extended upstream and downstream until it included significant species. The resulting pathways and the references used to arrive at them are summarized in Table 2.1. A full description of the NF-κB system appears later in section 2.6.

Here each pathway description consists of two parts, the predicate and the subject and are separated by the implication sign, $\Longrightarrow$. The information that the pathway contains can be understood as: "when the predicate is true, the subject is implied to occur in the future." The timing with which this dependence occurs is not known, but in this section we assume that the dependence relationship is implemented at the next time step as in [63].

Using the pathway data, one can determine which proteins are upstream of each other and also determine the set of predictor proteins, i.e. the proteins whose activity status collectively determines the time course updates of a given protein. In the general case, it is possible that one could have this information without having any pathway knowledge about the specific behavior of the regulation. While the algorithm presented below can handle this case equally well, in the NF-κB model considered here we did not have any knowledge of this type and therefore the predictor sets were derived directly from the pathway knowledge.

Table 2.1: Pathways comprising the NF-κB system. Reproduced with permission from [56].

Pathway	Reference
RIP1 $= 1 \implies$ NEMO $= 1$	[43]
A20 $= 1 \implies$ NEMO $= 0$	[43]
LTβR $= 1 \implies$ NEMO $= 1$	[43]
A20 $= 1$ and RIP1 $= 1 \implies$ NEMO $= 0$	[43]
RIP1 $= 0$ and LTβR $= 0 \implies$ NEMO $= 0$	[43]
TNFR $= 1 \implies$ AP-1 $= 1$	[43]
TNFR $= 0 \implies$ AP-1 $= 0$	[43]
TNFR $= 1 \implies$ RIP1 $= 1$	[44]
TNFR $= 0 \implies$ RIP1 $= 0$	[44]
LPS $= 1 \implies$ TNFR $= 1$	[54]
TNF$\alpha = 1 \implies$ TNFR $= 1$	[54]
LPS $= 0$ and TNF$\alpha = 0 \implies$ TNFR $= 0$	[54]
IKK$\alpha = 1 \implies$ p52 $= 1$	[43]
IKK$\alpha = 0 \implies$ p52 $= 0$	[43]
LTβR $= 1 \implies$ IKK$\alpha = 1$	[43]
NEMO $= 1 \implies$ IKK$\alpha = 1$	[43]
NEMO $= 0$ and LTβR $= 0 \implies$ IKK$\alpha = 0$	[43]
NEMO $= 1 \implies$ IKK$\beta = 1$	[90]
LPS $= 1 \implies$ IKK$\beta = 1$	[44]
NEMO $= 0$ and LPS $= 0 \implies$ IKK$\beta = 0$	[90, 44]
p65 $= 1 \implies$ IκB $= 1$	[43, 88]
p65 $= 1 \implies$ A20 $= 1$	[43, 88]
p65 $= 0 \implies$ A20 $= 0$	[43, 88]
IKK$\beta = 0 \implies$ IκB $= 1$	[43]
IKK$\beta = 1 \implies$ IκB $= 1$	[43]
IκB $= 1 \implies$ p65 $= 0$	[43]
IκB $= 0 \implies$ p65 $= 1$	[43]
IKK$\alpha = 1 \implies$ p65 $= 0$	[61]

Let us now demonstrate this algorithm on a part of the NF-κB system. From Table 2.1, we consider the pathways related to the behavior of NEMO. These are the first five entries in that table and are given by:

$$\text{RIP1} = 1 \implies \text{NEMO} = 1 \tag{2.1}$$

$$\text{A20} = 1 \implies \text{NEMO} = 0 \tag{2.2}$$

$$\text{LT}\beta\text{R} = 1 \implies \text{NEMO} = 1 \tag{2.3}$$

$$\text{A20} = 1 \text{ and } \text{RIP1} = 1 \implies \text{NEMO} = 0 \tag{2.4}$$

$$\text{RIP1} = 0 \text{ and } \text{LT}\beta\text{R} = 0 \implies \text{NEMO} = 0 \tag{2.5}$$

The pathways listed above mandate the following relationships: when RIP1 is activated, it activates NEMO; when A20 is activated, it deactivates NEMO; when LTβR is activated, it activates NEMO; when both A20 and RIP1 are activated, NEMO is deactivated; and when RIP1 and LTβR are both inactive, NEMO is deactivated. From these, we can infer that a reasonable predictor set for NEMO is $\{\text{A20}, \text{LT}\beta\text{R}, \text{RIP1}\}$. One can similarly arrive at predictor sets for the other biological entities in Table 2.1.

2.3 Creating Karnaugh maps from pathways

Having identified the predictor set for each protein, we can use a Karnaugh map [52] to determine the update rule for that protein. The method used is an extension of the one developed in [63]. For each protein, using its predictor set, we initialize a Karnaugh map with every entry in the map containing the unknown 'x'. This is consistent with the observation that to start with we have no information about how the current value of the prediction proteins affects the update value of the predicted

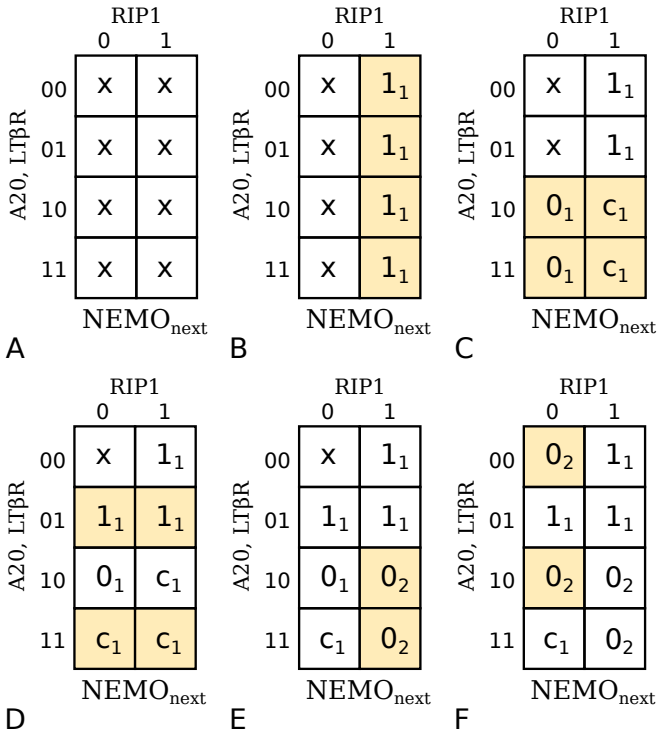

Figure 2.1: The process of developing the Karnaugh map using the pathways associated with NEMO. For each step in the process, we evaluate each pathway in turn. In each pathway, the predicate specifies certain locations in the Karnaugh map and these are shaded in yellow in the corresponding table. Reproduced with permission from [56].

protein at the next time step. The Karnaugh map entries can then be updated by incorporating the pathway information. For instance, consider the update of NEMO. The initial blank Karnaugh map for it is shown in Fig. 2.1A. We will refer to the locations in the Karnaugh map using the vector [A20, LTβR, RIP1]. Thus 000 would correspond to the square in the upper left corner of the map and so on. Now, using the first pathway: RIP1 = 1 $\implies$ NEMO = 1, we can fill all the entries that correspond to RIP1=1 with ones, and this results in the Karnaugh map shown in Fig. 2.1B.

We next proceed to fill the locations 100 and 110 with zeros to satisfy the requirements of the second pathway: A20 = 1 $\implies$ NEMO = 0, but are faced with a conflict when RIP1=1 and A20=1. At the two conflicting locations we replace the ones with c^1. The letter c indicates a conflict while the superscript 1 indicates that the conflict comes from pathways which contain only one protein in the predicate. The reasoning for this notation will become clear later in this section. This results in the table in Fig. 2.1C.

We next consider pathway three: LTβR = 1 $\implies$ NEMO = 1, which mandates that the 'x' at location 010 be replaced by a 1. Since location 110 contains a 0, so we replace it with a c^1 to indicate that there is a conflict. Finally, location 111 already has a c^1 and because this pathway only contains one protein in its predicate, we must leave the c^1 in place. This leads to the table shown in Fig. 2.1D.

Next we look at the fourth pathway: A20 = 1 and RIP1 = 1 $\implies$ NEMO = 0. The predicate here applies to locations 101 and 111. Both of these contain c^1 conflicts, but because this pathway contains two proteins in its predicate, we acknowledge this pathway has more specific information regarding this particular experimental scenario and can override the c^1 conflicts. Therefore we fill the locations 101 and 111 with zeroes and obtain the table shown in Fig. 2.1E.

For the final pathway: RIP1 = 0 and LTβR = 0 $\implies$ NEMO = 0, the predicate applies to the locations 000 and 100. The former has an x, so we replace it with a 0, and the latter is already a 0. Therefore we are now finished with NEMO's pathway information and are left with the Karnaugh map of Fig. 2.1F. In this map, we see there is only one uncertainty condition at the location 110.

The procedure demonstrated on the NEMO example above can be generalized by proceeding through the pathways in order from the least specific predicates to the most specific ones and filling in the Karnaugh map entries if information is available or invalidating information already provided if there are conflicts in the pathway information. This general procedure is presented in Algorithm 1. The key difference between the algorithm presented in [63] and the one presented here is that in [63], one attempts to resolve the conflicting entries in the Karnaugh maps by suitably altering the timings of some of the pathways whereas here the conflicts in the Karnaugh maps are retained. Consequently, the state transitions following a conflict will not be unique and by assuming that all the subsequent states are equiprobable, we can come up with probabilistic state transition graphs which are introduced next.

2.4 Probabilistic state transition graphs

The Karnaugh maps generated using the procedure of the last section can be used to produce the probabilistic state transition graph (PSTG) of the system. A PSTG is a directed graph that describes the evolution of the biological system through time. It consists of k^n nodes that correspond to the states of the system where k is the number of quantization levels associated with the activity state of each protein (assumed throughout the rest of this section to be two for a binary discretization) and n is the number of proteins in the system. Additionally, each directed edge indicates a viable transition between states as allowed by the pathway information.

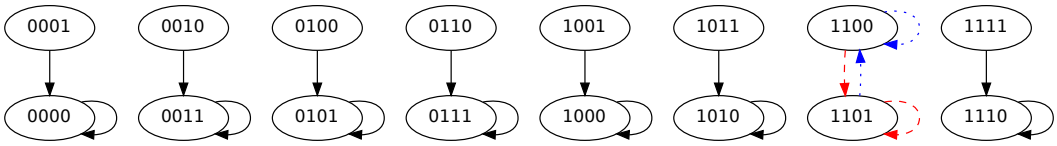

Figure 2.2: A simplified PSTG for the NEMO protein with predictors that are all static for the purposes of illustration. Here the binary value of the state should be interpreted as [A20, LTβR, RIP1, NEMO]. The red, dashed edges represent one possible configuration of the network, while the blue, dotted edges represent the other configuration. Reproduced with permission from [56].

For example, using the NEMO pathways and the Karnaugh map generated in the last section we obtain a simplified but illustrative example of a PSTG for this system as shown in Fig. 2.2. We assume here that the predictor proteins have no predictors themselves and therefore exhibit static behavior. This is seen in Fig. 2.2 where the edges of the PSTG, or the allowed transitions, are between states with identical values for the predictor proteins and only differing in the values of NEMO. Also, based on the Karnaugh map generated from the NEMO pathways, we expect uncertainty at the state 110x and indeed, both the states 1100 and 1101 have two outgoing edges each. One of these creates a self loop, and the other directs to the corresponding state with the value of NEMO flipped.

The PSTG is actually a compact representation for the class of networks which the uncertain Karnaugh maps such as the one in Fig. 2.1F generate. To use the example above, we could also represent the uncertainty of NEMO at 110x by creating two separate networks with two corresponding state transition diagrams. One network would contain the blue, dotted edges in Fig. 2.2 and predict that NEMO should equal 0 when A20=1, LTβR=1, and RIP1=0, while the other would have the red, dashed edges and predict that NEMO should equal one for the same set of predictor values.

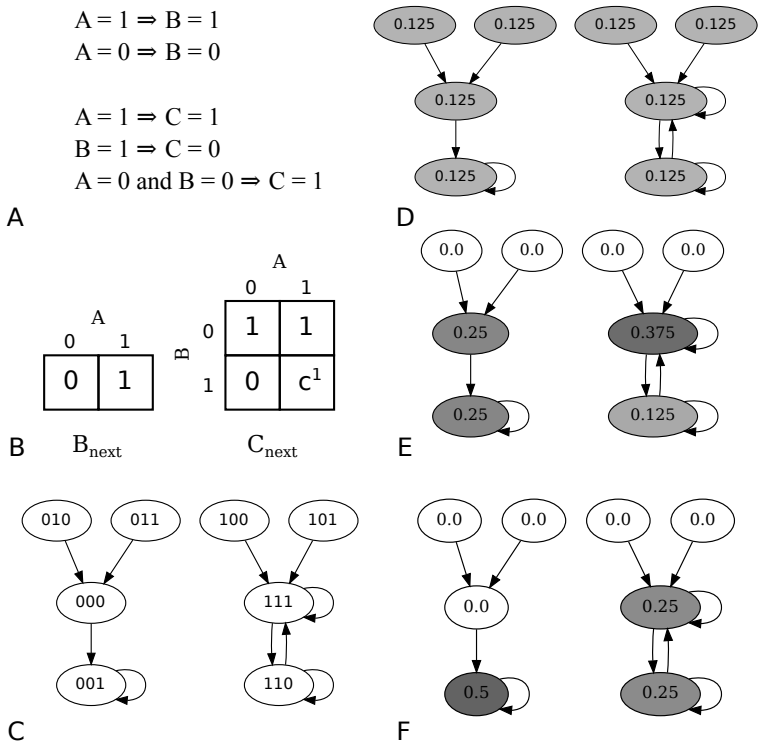

Figure 2.3: A synthetic example of the complete process from pathways to long run probabilities. (A) Pathways for the three proteins: protein A has no predictors, B is predicted by A, and C is predicted by A and C. (B) The resulting Karnaugh maps with one final conflict obtained using the method described in Algorithm 1. (C) The resulting PSTG shows two separate basins. The left basin has a single attractor state 001 while the right basin has two states in the attractor, 111 and 110. (D-F) The probability mass in each state at the (D) initial stage (E) after the first state transition, and (F) after many state transitions have taken place. (G) The resulting protein B knockout (B= 0) PSTG. The state space is halved as the states with B=1 are no longer considered in the possible transitions. Reproduced with permission from [56].

Let us now demonstrate the process of converting Karnaugh maps to PSTGs. We will use the synthetic example in Fig. 2.3A where we are given five pathways for the three proteins A, B, and C. These produce the two Karnaugh maps in Fig. 2.3B and the resulting PSTG is shown in Fig. 2.3C. As can be seen, protein A has no predictors and thus the state space can be partitioned into two basins according to its activity.

To produce the PSTG in Fig. 2.3C, we begin by listing all the possible 2^3 states as unconnected nodes in a graph. Then, for each node we use the Karnaugh maps to update the status of each gene and then concatenate this information to determine the next state to which the model will transition to. For example, for the state 000, protein A has no predictors and therefore no Karnaugh map so we assume it remains in the state 0. Using protein B's Karnaugh map, A is currently 0, so $\mathrm{B}_{\mathrm{next}} = 0$. Finally, proteins A and B being zero imply $\mathrm{C}_{\mathrm{next}} = 1$. Combining all this information creates a transition edge from state 000 to 001 in the PSTG.

As another example, consider the state 111. Following the above logic, $\mathrm{A}_{\mathrm{next}} = 1$ and $\mathrm{B}_{\mathrm{next}} = 1$ but $\mathrm{C}_{\mathrm{next}}$ is a conflict, so 111 will progress to 11x which can be expanded to 110 and 111. This is shown in Fig. 2.3C, where the node 111 has two outgoing edges, a self loop and an edge linking to 110.

The PSTG, as in Fig. 2.3C, can also be interpreted as the state transition graph of a Markov chain. Thus, the Karnaugh maps can be converted into a $2^n \times 2^n$ ($n = 3$ here) transition probability matrix where each entry $[p_{i,j}]_{2^n \times 2^n}$ represents the probability of the model transitioning from state i to state j in one time step. This matrix is stochastic, and is a row normalized sum of the individual deterministic state transition matrices of the resulting class of Boolean networks that would be required to accommodate the uncertainties associated with the different Karnaugh maps. The PSTG is a compact, sparse representation of this transition probability

matrix, and this is what motivates us to use this framework to simulate the long run behavior of the model.

2.5 Simulating long run behavior

A chief objective of the developed model is to obtain some useful measure of the long run behavior of the system akin to the steady state distribution for an ergodic Markov chain. Unfortunately, the PSTG generated using the above method does not necessarily provide an irreducible or aperiodic state space. Therefore, we must approximate its long run behavior as explained in the following subsections.

2.5.1 A synthetic example

For illustrative purposes we start with the synthetic example in Fig. 2.3. Since we do not have any prior knowledge as to whether a given cell exhibits activated protein A, we make no assumptions about the cell initially being in either basin. In general this would apply to all the proteins in our model and ,therefore, we initialize the states in our network with uniform initial probabilities of 0.125 each as shown in Fig. 2.3D.

We can think of this initial probability as our initial belief of the state of the system. Then the algorithm can be understood as an application of the knowledge incorporated in our model to refine the initial belief. We will refer to this belief as the probability mass.

For transitions between states we again utilize uniformity by assuming that the transition probability from one state to the chronologically next one is equal to the inverse of the outgoing degree from the state of origin. So for a state with two outgoing edges such as 110 in Fig. 2.3C, the probability to transition to any of its children (110 and 111) will be 0.5.

Now our model dictates the evolution of the states, so any cell in state 010 will

progress in one time step to state 000, so we can move the probability mass in state 010 to state 000. The same applies to state 011 as its mass will “flow" to state 000. Now, state 110 has two outgoing edges, so our model is ambiguous about whether a cell in state 110 will stay in state 110 or if it will transition to 111. Therefore, we split the probability mass in state 110 and place half of it in state 110 and half in state 111 for the next time step. The result of the first run of the process is seen in Fig. 2.3E.

Through iteration of this process, the probability in each transient state tends towards 0.0. This can be seen in the example as after the first step of the algorithm, the states along the top are transient because once we leave them, we never return. This is clear from Fig. 2.3E where the probability mass has already been depleted from these transient states and will remain at 0.0 for the remaining lifetime of the model under these conditions. At the next stage, we arrive at Fig. 2.3F, from which we can see that after the second iteration of the algorithm, state 000 has no remaining probability mass and will remain that way for the remaining simulation because the state is transient and has no return path.

In simple attractor cycles such as the one consisting of the nodes 111 and 110 or singleton attractors such as 001 in Fig. 2.3F, we can calculate the long run probability mass intuitively. The resulting mass in a singleton attractor is obtained by summing all the initial masses in its basin of attraction. For simple attractor cycles, we can sum all of the initial mass in the basin of attraction for that cycle and divide it by the number of nodes in the cycle. In this way, we can get an approximation about how often we could expect the biological system to exist in certain states.

2.5.2 The general method

The intuitive reasoning described above where we iterate over the nodes and add their masses to the downstream nodes applies when the PSTG is acyclic and a preorder traversal through the graph exists. However, with the possibility of cycles, no such preorder traversal must exist and we are forced to introduce an accumulation buffer to describe a general algorithm that works with cycles for any node traversal order.

Let each node have a probability mass and an accumulation buffer. For each node we first initialize the mass of each node to be uniformly distributed over the entire graph, and set the accumulation buffer to zero. Then for each node divide its current probability mass by the number of its outgoing edges and then add that amount to the temporary accumulation buffers for each of its child nodes. Then for each node, set the probability mass to be the value in its accumulation buffer and reset the accumulation buffer to zero.

Repeating this algorithm will result in the probability mass accumulating in the attractors of the system. In the case of aperiodic attractors, the masses will converge to a limiting probability mass, but we must be careful about handling the possibility of periodic attractors. Repeatedly using the algorithm in this case might result in the propagation of an unbalanced mass around the cycle, analogous to oscillatory behavior in undamped systems. To overcome this problem, it is necessary to first run the algorithm a sufficient number of times to eliminate all the transient states and then go through several runs of the algorithm and the final probability mass in the attractor cycle states can then be taken to be an average of the probability masses from the final runs. This essentially smoothens out any oscillations in the probability mass distribution in the attractor cycles.

2.5.3 The steady state activity vector

Now, from a biological point of view, it is more relevant to determine how often a particular protein is active instead of determining how much time is spent in a particular state. Accordingly, we apply a transformation to the long run probability approximation, and define a steady state activity (SSA) vector with n components, indexed by 0 to $n-1$, corresponding to the n proteins in the model. The i-th component of the SSA vector, characterizing the activity of the i-th protein, is computed as:

$$\mathrm{SSA}(i) = \sum_{j=0}^{2^n-1} \mathrm{PA}(j) z_i(j)$$

where $z_i(j)$ is the binary value of the i-th protein in the j-th decimal state, and $\mathrm{PA}(j)$ is the j-th entry of the Probability Approximation vector resulting from the algorithm presented in the previous subsection. The resulting $\mathrm{SSA}(i)$ takes values in the interval $[0, 1]$ according to the probability that the model is likely to exist in states where protein i is active. For example, an $\mathrm{SSA}(i)$ value of 1 indicates that the i-th protein is active in every attractor state.

Applying this to our example, in Fig. 2.3F, the SSA for protein A, i.e. $\mathrm{SSA}(0)$, is calculated by considering the two attractor states 111 and 110 with active protein A as shown in the right hand basin in the figure. Therefore, $\mathrm{SSA}(0) = 0.25 + 0.25 = 0.5$. Similarly, SSA(1) (B) is also 0.5 while SSA(2) (C) is 0.75, the latter being due to the fact that all attractors have protein C active except the state 110 which has a final mass of 0.25.

We next further justify the need for using the SSA vector instead of the state vector. Consider the network that we have constructed by applying Algorithm 1 to the set of 28 pathways involving NF-κB. With the external stimuli set to TNF$\alpha = 1$,

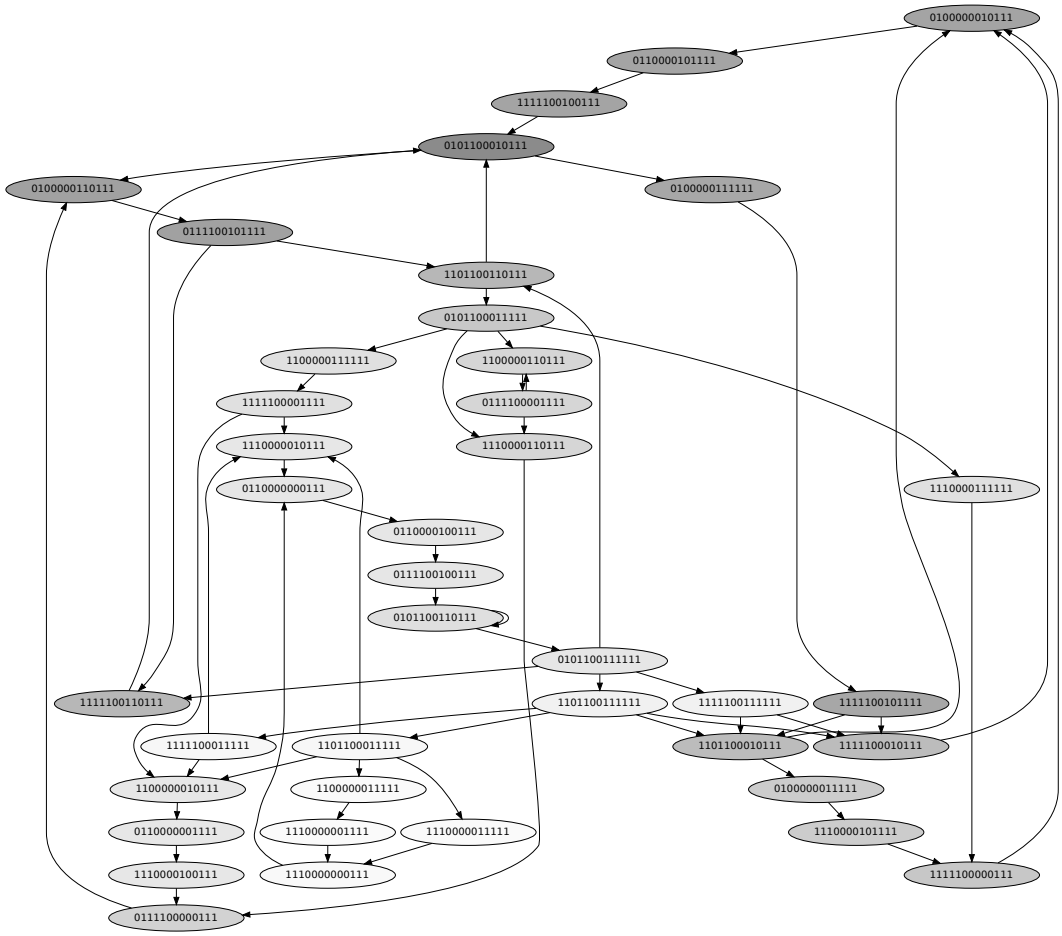

Figure 2.4: The resulting communicating class of states for the full NF-κB PSTG when the stimuli conditions are set to TNF=1, LPS=0, and LTβR=0. The thirteen bit binary vector can be read as [A20, AP-1, IκB, IKKα, IKKβ, LPS, LTβR, NEMO, p52, p65, RIP1, TNFα, TNFR]. To give a walk through of one state transition, starting at the upper right state, 0100000010111, many things occur in one transition: IκB is inactive and thus at the next state p65 translocates to the nucleus to become active; NEMO is activated by RIP1; p52 is deactivated as IKKα is not activating it; and IκB becomes active, as constitutive expression allows it to repopulate the cytoplasm in the absence of activated IKKβ. All of these changes results in the model evolving to state 0110000101111. Reproduced with permission from [56].

LPS= 0, and LTβR= 0, the only communicating set of states for the resulting PSTG is displayed in Fig. 2.4. Due to the size and complexity of the resulting set of states, it would be difficult, if not impossible, to compare the behavior of this PSTG with that obtained from any of the knockout experiments. The concept of the SSA vector was introduced precisely to ameliorate this problem and aids in carrying out qualitative comparisons with the experimental data. This will be demonstrated in the next section.

2.6 The NF-κB system

Nuclear factor-κB (NF-κB) is a protein dimer from the rel family of transcription factors that promote the expression of over 100 genes, primarily in the immune system [37]. The NF-κB system's primary role in the immune system is in the production of inflammatory cytokines, small signaling proteins used extensively in cell to cell communication. NF-κB also has both proapoptotic and antiapoptotic effects on the cell and the balance of these responses can be adjusted by the stimulus context.

The NF-κB transcription factor is a key element in the inflammation stress response pathway. The general architecture of this system is typical of several stress response pathways [84]. The transcription factor NF-κB is sequestered in the cytosol by the "sensor" which in this case is IκB and when degraded by the "transducer" of IKKβ, it allows for a rapid downstream response without the lag associated with *de novo* protein synthesis. As discussed in [84], this combination of a transducer, sensor, and transcription factor is a common motif seen in stress response pathways and forms the backbone of the NF-κB system.

The mammalian NF-κB family consists of p65 (RelA), RelB, c-Rel, NF-κB1 (p52/p100), and NF-κB2 (p50/p105). NF-κB is constitutively expressed, but sequestered in the cytosol by a family of IκB inhibitor proteins which include the p100

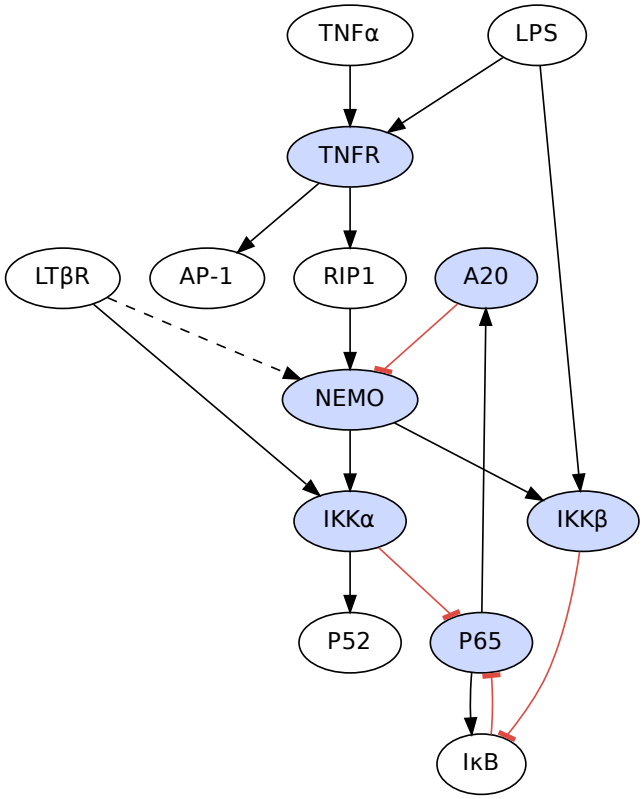

Figure 2.5: The pathway structure of the NF-κB system. Blue proteins are those that are knocked out in the validation portion of this section. The presence of a directed edge indicates that a pathway exists that shows the upstream protein causes a change in the activity of the downstream protein. Inhibitory pathways are marked red and terminated with a filled dot. One thing to note here is that the LPS induced autocrine production of TNFα would seemingly imply that an excitatory connection should be made between LPS and TNFα. However, because we want to exogenously control TNFα, LPS, and LTβR in our knockout simulations, we consider TNFα to be an exogenous stimulus, thereby allowing us to control that level in simulations without affecting the autocrine feedback loop of LPS. The second thing to note is the dotted connection from LTβR to NEMO which indicates this is a pathway with unknown mechanism but described in [43]. Reproduced with permission from [56].

and p105 precursors to p52 and p50, respectively, along with IκBα, IκBβ, IκBϵ, IκBγ, and BCL-3 [43]. These IκBs prevent the NF-κB dimers from reaching their binding sites in the nucleus.

The IκB kinase complex (IKK) consists of the IKKα, IKKβ, and IKKγ (NEMO) subunits. The NEMO (NF-κB essential modulator) subunit is a regulator and maintains the IKKα and IKKβ subunits in inactive states.

The signaling pathways involved in the NF-κB system are shown in Fig. 2.5. NF-κB activation is generally considered to occur through two separate cascade pathways. The canonical pathway is primarily activated by the proinflammatory cytokine Tumor Necrosis Factor-α (TNFα). When TNFα binds to TNF receptor protein (TNFR), it begins a signaling cascade that through the receptor interacting protein 1 (RIP1) activates the NEMO subunit of IKK which activates both the IKKα and IKKβ subunits [81]. The IKKβ subunit then proceeds to phosphorylate IκB proteins which leads to their destruction through polyubiquitination and allows NF-κB dimers, primarily p65 heterodimers and homodimers to translocate to the nucleus and bind to promoter regions [43].

Bacterial lipopolysaccharide (LPS) is a component of bacterial cell walls that provides an activating stimulus for Toll-like receptors 2 and 4 (TLR2 and TLR4). These receptors also activate the canonical pathway but through MyD88 and Trif intermediary proteins [11] that directly activate the IKKβ subunit without activating NEMO [81]. Also, the LPS dependent pathway indirectly activates the canonical pathway through autocrine stimulation through the production of TNFα.

The alternative pathway is activated through CD40 and LTβR and through the NIK protein directly activates the IKKα subunit which through phosphorylation, processes p100 into p52 which activates the nuclear localization segment (NLS) which allows the p52 dimer to translocate into the nucleus. The alternative pathway also

activates the canonical pathway through an unknown mechanism [43].

NF-κB activates two genes in particular that produce IκB and A20. Both of these act as negative feedback to dampen the response of the canonical pathway. A20 binds to NEMO and impairs the activation of the IKKβ and IKKα subunits by RIP1 [90]. Additionally, NF-κB activates antiapoptotic genes such as cFLIP (not shown in Fig. 2.5) that counteracts the TNFα induced activation of the proapoptotic AP-1 family such as c-Jun [76]. Consequently, NF-κB knockouts are likely to exhibit apoptotic behavior when subjected to TNFα stimulation. This will be borne out by some of the knockout studies considered later in the next section.

2.7 Towards model validation using knockout studies

The model developed by us in this section was designed to preserve the biological state transitions and stable state attractor cycles. Accordingly, it seems reasonable to validate our model using the experimentally observed long run behavior of biological systems. We will specifically focus on animal knockout models. This kind of model validation is appropriate given that a long term goal of our research is to enhance medical treatment in patients. Currently, treatment is provided through therapeutic drugs which have physiological effects on the order of 8 to 24 hours which can be considered to be long run behavior in the context of regulatory networks. Thus, long run behavior would be particularly appropriate for predicting drug effects and patient outcomes.

Biologists use knockout models to disable a specific gene or a set of genes in a model animal and then observe the resulting physiology to determine protein functions and interactions. Our stochastic state models also provide us a platform with which we can replicate these experiments and examine the resulting steady states. Comparison of the proteins' known functions and physiological phenotypes,

the resulting increases or decreases in prevalence, and the recorded physiology of the real knockout experiment together provide us with a mechanism for validating our stochastic state space model. For instance, consider the knockout experiment in Fig. 2.3G. Here we have knocked out protein B. This results in a PSTG where all states with protein B active are no longer considered valid and the resulting model has a PSTG consisting only of states with inactive protein B. This results in two attractor states, one for each original basin.

The PSTG resulting from the 28 NF-κB pathways in Table 2.1 was too large to be visually interpreted, although for illustrative purposes a small subset of it is included in Fig. 2.4. We compared the behavior of our model in a steady state fashion with the phenotypes and measured protein quantities as found in the knockout studies. Due to the qualitative nature of the data collected in the knockout experimental studies used, the comparisons between the model and the study will by necessity be qualitative. We believe that this still allows for satisfactory validation given the complexity of the model, the inherently noisy nature of biological systems and experimentation, and the large number of knockout studies examined.

2.7.1 A20$^{-/-}$

Werner et al. [88] aimed to derive ordinary differential equation (ODE) models of NF-κB regulation in response to TNFα and LPS stimulation. One of their model parameters includes the negative feedback of A20, and to justify this, they compared A20$^{+/+}$ against A20$^{-/-}$ Murine 3T3 immortalized fibroblasts and measured IKK and NF-κB activity in response to 45 minutes of TNFα stimulation. It is clear in this comparison that the A20$^{-/-}$ activity of IKK and NF-κB is much higher than that of the A20$^{+/+}$ cells. This is consistent with Table 2.2(a) produced by simulating our model where the levels of IKKα and p65 both increase when the model is constrained

Table 2.2: Knockout studies and simulations. Reproduced with permission from [56].

Knockout Species	Baseline Conditions		Baseline SSA	KO SSA	Reference
	TNF	LPS			
(a) A20$^{-/-}$	1	0	p65=0.389, p52=0.611, IKKα=0.611	p65=0.400, p52=1.00, IKKα=1.00	[88]
(b) IKK$\beta^{-/-}$	1	0	p65=0.389, IκB=0.487, AP-1=1.00	p65=0.00, IκB=1.00, AP-1=1.00	[65]
(c) IKK$\beta^{-/-}$, TNFR$^{-/-}$	1	0	p65=0.389, AP-1=1.00	p65=0.00, AP-1=0.00	[65]
(d) p65$^{-/-}$	1	0	p65=0.389, AP-1=1.00	p65=0.00, AP-1=1.00	[35, 76]
(e) IKK$\alpha^{-/-}$	0	1	p65=0.600, A20=0.600, IκB=0.300	p65=0.667, A20=0.667, IκB=0.333	[66, 61]
(f) IKK$\beta^{-/-}$	0	1	p65=0.60, AP-1=1.00	p65=0.00, AP-1=1.00	[35, 75]
(g) NEMO$^{-/-}$ (macrophage)	0	1	p65=0.60, AP-1=1.00	p65=0.67, AP-1=1.00	[55]
(h) NEMO$^{-/-}$ (general)	0	1	p65=0.67, AP-1=1.00	p65=0.67, AP-1=1.00	[55]

to A20$^{-/-}$. While the increase in p65 is small, the direction of the change is consistent with the findings of Werner et al.

2.7.2 IKK$\beta^{-/-}$ and TNFR$^{-/-}$

Li et al. [65] used IKK$\beta^{-/-}$ knockout mice to investigate the role of IKKβ in the NF-κB signaling pathway. They determined that the lack of IKKβ increased hepatocyte death due to TNFα (TNFα toxicity). This was caused by a reduction in the amount of phosphorylated IκB and a corresponding decrease in the activation of NF-κB which is anti-apoptotic. They also found that the IKK$\beta^{-/-}$ knockout did not affect c-Jun levels, a member of the proapoptotic AP-1 family which helps explain the increased toxicity. They also measured an increase in stability in IκB from IKK$\beta^{-/-}$ lines, which is seen in our model as an increase in the activity of IκB in Table 2.2(b).

They also determined that a IKK$\beta^{-/-}$, TNFR$^{-/-}$ double knockout where both of these genes were knocked out simultaneously allowed the mice to survive to term (rescuing the phenotype). We mirrored this same result in our model as the IKK$\beta^{-/-}$, TNFR$^{-/-}$ double knockout in Table 2.2(c) shows the same reduction of p65 as the single knockout, but the c-Jun (AP-1 family) activation is also reduced which reduces the pro-apoptotic nature of the TNFα stimulus under IKK$\beta^{-/-}$ knockout conditions. This explains the reduction in TNFα toxicity.

2.7.3 p65$^{-/-}$

Prendes et al. [76] used fetal liver hematopoietic precursors from mice embryos deficient in RelA (p65) to study the effect of RelA deficiency in lymphocytes. They found that the loss of RelA increased TNFα toxicity greatly which was ameliorated when cells were induced by virus to produce the antiapoptotic NF-κB target gene cFLIP. This indicates that the increased cell death was due to the inhibition of NF-

κB and this is backed by our knockout model by a reduction in RelA at steady state in Table 2.2(d).

2.7.4 IKK$\alpha^{-/-}$

Li et al. [66] used embryonic liver-derived macrophages (ELDM) from IKK$\alpha^{-/-}$ mice to determine the role of IKKα in the innate immune system's inflammation response. IKK$\alpha^{-/-}$ ELDM cells were found to exhibit higher than normal antigen presenting response and higher NF-κB levels in response to LPS stimulation. In addition, Lawrence et al. [61] used a model with an inactivatable variant of IKKα (denoted by IKK$\alpha^{A/A}$) and observed an increase in NF-κB and A20 upon the application of LPS, both of which match our model in Table 2.2(e).

Li et al. found a decrease in the post-induction response of IκB in their IKK$\alpha^{-/-}$ cells whereas Lawrence et al. measured an increase in the amount of IκB for their IKK$\alpha^{A/A}$ macrophages. Li et al. put forth a possible explanation for this discrepancy: in IKK$\alpha^{-/-}$ knockouts, the absent IKKα proteins are no longer competing with IKKβ for NEMO binding locations allowing more IKKβ to homodimerize under NEMO [44]. This in turn results in more effective IκB kinase activity and thus less IκB than the IKKα-IKKβ-NEMO complexes that exists in IKK$\alpha^{A/A}$ mutants and normal cells.

Our model as presented in this section uses only two states to describe the state of a protein. This approximation suffices when the behavior of an inactivated protein is the same as that when that protein is absent. In this case, it is okay to associate both the absence and inactivation of the protein into state 0. However, in the case of IKKα, the effect of the protein's absence is different from that of its inactivation and thus for complete accuracy we would need an additional state to encode for this level of detail.

Accordingly, our model has included the pathway associated with IKKα's inactivation and therefore matches the observations of Lawrence et al's inactivation model because their experimental method resulted in an inactivation of IKKα rather than its complete absence as in Li et al's.

2.7.5 IKK$\beta^{-/-}$

Park et al. [75] used fetal liver derived macrophages (FLDM) deficient in IKKβ to investigate the mechanism of macrophage survival under stimulus to TLR4 receptors. Some bacterial toxins such as *Salmonella* AvrA inhibit NF-κB while stimulating the TLR4 receptor which was observed to result in the stimulation of macrophage apoptosis. This is mirrored in our model in Table 2.2(f) where we see that p65 activity drops while the AP-1 proapoptotic family remains activated. This observation is along the same lines as that in Table 2.2(b) where, in addition, we also tracked alterations in IκB activity under different exogenous stimuli conditions.

2.7.6 NEMO$^{-/-}$

Kim et al. [55] analyzed NEMO$^{-/-}$ Murine B cells. Because these mice die early in embryogenesis, they used an *in vitro* differentiation process to convert embryonic stem cells to B cells. They found that NEMO is not required for B cell development, but does affect its survival. Specifically, after an application of LPS for three days (+LPS) or mock stimulation for the control (-LPS), the wild-type B cells maintained population levels while the +LPS NEMO-deficient group declined in population. Oddly enough however, the -LPS NEMO-deficient cell group also declined in similar proportions which confounds the simple explanation of NF-κB stimulation from LPS increasing the cell apoptosis rate.

In our model, we see in Table 2.2(g) that our NEMO$^{-/-}$ simulation actually shows an increase in p65 NF-κB activation levels with a constant AP-1 level which would

seem to indicate an increase in cell survival. This conflicts with the finding in Kim et al. but looking more closely at the pathways, we see that the IKK$\alpha = 1 \implies$ p65 $= 0$ pathway in table 2.1 is from [61] where the inhibition of p65 from IKKα is seen in macrophages. It is entirely possible that this pathway is different in developed B cells and indeed, if we remove this pathway from the model we see in Table 2.2(h) that the p65 levels are unchanged in the NEMO-deficient model which means the model does not contain enough information to predict any change in behavior for this knockout configuration.

This reinforces a key assumption made in the model derived in this section. The model is only as accurate as the pathway data used in its generation, and for biological regulatory systems it is often acceptable to use pathways from different cell types and contexts. However, to achieve maximum model fidelity and prediction, it is necessary to obtain pathways from the same cell types that we wish to make predictions about.

2.8 Conclusions

In this section we have presented a method to produce a regulatory network model using only minimal assumptions of predictor proteins and utilizing literature backed pathway information. The resulting networks assume no data other than that given and were validated using a number of biological knockout experiments from the literature that gave matching results. The use of minimal modeling assumptions, along with the use of literature backed information result in a model that is built on a solid foundation of biological experimentation, and will allow for further validation and refinement through comparison with high-throughput data and new pathway data as they become available.

We believe that techniques such as these will play a critical role in future drug

discovery and predicting the effects of potential drugs. The linear and intuitive nature of (marginal) biological pathways does not completely capture the (multivariate) complex network level behaviors of reality. However, the methods presented here will allow for these pathways to be used as a whole to describe the possible network behaviors in a way that could some day guide physician therapy and drug design.

To extend this work, we will next develop new techniques to leverage the data derived from high-throughput experiments to refine these pathway derived models. This will allow for networks that merge the two greatest sources of biological knowledge, the new and the old, into models with better predictive power. Additionally, it will be done in a minimal assumption environment that can be extended and refined as new pathways are developed and validated.

Some pathways, such as the phosphorylation of IκB by IKKα and IKKβ could not be represented in our modeling with one hundred percent accuracy due to quantization errors in our binary discretization. In reality, both IKKβ and IKKα phosphorylate and deactivate IκB but IKKβ deactivates IκB at a much greater rate than IKKα. Unfortunately, the binary quantization does not allow this information to be retained accurately and in this model we decided to ignore the direct effect of IKKα on IκB. Such a decision could be justified based on the wide difference in the magnitudes of the effects of IKKβ and IKKα on IκB. In the future, however, we would like to use finer quantization levels for important and well understood components of the regulatory system to enable a model that more closely reflects reality without greatly increasing the knowledge requirements or uncertainty.

In this section, we have ignored the fact that many species in the model such as AP-1 and IκB are actually dimerizing families of proteins, and the specific proportions of these dimers could be important to determining the resulting cellular response. The naive approach of simply accounting for each possible combination

of dimers would result in a huge increase in uncertainty in the model and future work should focus on more sophisticated ways to handle this real-world biological complexity.

In the next section, we consider an improvement to this heuristic algorithm developed here. Briefly, by starting with an appropriate estimation problem and cost criterion we can produce optimal estimators relative to the modeling assumptions and cost criterion.

Algorithm 1: ProduceKMapForProtein(P,x)

```
/* P is the sorted list of pathway segments as described in [63]
   in ascending number of predictive proteins.  */
/* x is the name of the specific protein that we want to produce
   a Karnaugh map for.  */
/* This algorithm is simple and optimized for clarity of
   exposition rather than runtime.  Therefore, we use two loops
   through the pathways.  The first is to collect the set of
   predictor proteins for this protein, and the second is to fill
   in the entries of the Karnaugh map.  */
foreach p ← P do
    // Get the subjects S of this pathway
    S ← Subjects(p);
    if x ∈ S then
        // Add the proteins from this pathway's predicate
        // to the set of predictors:
        PredictorSet ← PredictorSet ∪ PredicateProteins(p);
initialize the Karnaugh map K[] with x's in each location;
foreach p ← P do
    // Decompose the pathway into its constituent parts
    S ← Subjects(p);
    if x ∈ S then
        n ← |PredicateProteins(p)|;
        C ← PredicateCondition(p);
        v ← NextStateValue(p);
        // Now iterate over the locations in the Karnaugh map
        foreach permutation e of the values of the PredictorSet do
            /* If the permutation e matches the condition of the
               predicate for this pathway, and if this pathway can
               override the information already in the Karnaugh map
               (i.e.  it is more specific than it with n < z), then
               overwrite it with the next state value subscripted
               with the specificity of this pathway.  Otherwise mark
               the location as a conflict with this pathways
               specificity of n.  */
            // for all z < n and y ∈ {0,1}
            if C ⊂ e and (K[e] = c_z or K[e] = x or K[e] = y_z) then
                K[e] ← v_n;
            else
                K[e] ← c_n;
// K[] is the resulting Karnaugh map
```

3. MCMC IMPLEMENTATION OF THE OPTIMAL BAYESIAN CLASSIFIER FOR NON-GAUSSIAN MODELS: MODEL-BASED RNA-SEQ CLASSIFICATION*

3.1 Background

The possibility of genomic phenotype classification arose with the inception of gene-expression microarrays. From the outset, two fundamental problems have frustrated the endeavor: (1) the inaccuracy of microarray measurements, and (2) small samples. Our particular application of interest is classification using RNA-Seq data. Modern RNA-Seq technologies sequence small RNA fragments (mRNA) to measure gene expression, where the number of reads mapped to a gene on the reference genome defines the count data. Given that RNA-Seq data has advantages over microarray data, in particular, more accurate measurement, we still confront the second fundamental problem, which is statistical, not technological: small samples cause re-sampling-based classifier error estimators to be very inaccurate due to excessive variance and lack of regression with the true error [7, 41, 39, 40]. Since the error rate of a classifier quantifies its predictive accuracy, it is the salient epistemological attribute of any classifier. The inability to satisfactorily estimate the error with model-free methods with small samples implies that genomic classifier error estimation is virtually impossible without the use of prior information, so that the whole small-sample classification problem becomes unapproachable in a model-free framework [29].

*Parts of this section are reproduced with permission (CC by 4.0) from Knight, J.M.; Ivanov, I.; Dougherty, E.R. "MCMC implementation of the optimal Bayesian classifier for non-Gaussian models: model-based RNA-Seq classification", BMC Bioinformatics Page 401. Volume: 15, Issue: 1, 2014 doi:10.1186/s12859-014-0401-3 Copyright © 2014 Knight et al.; licensee BioMed Central Ltd.

The situation has been addressed by utilizing prior knowledge via a Bayesian approach that considers a prior distribution on an uncertainty class of feature-label distributions [15, 16]. For expression-based classification, prior distributions have been constructed using expression data not employed in classifier design [14] and known regulatory pathways [32]. Given that a prior model must be assumed to achieve satisfactory error estimation, an obvious course of action is to derive an optimal classifier based on the prior knowledge and the sample data, the result being an optimal Bayesian classifier (OBC) that is guaranteed to have the best average performance of any classifier relative to the posterior distribution derived from the prior distribution and data [19, 20]. While Bayesian classification does not depend on particular distributional forms, closed-form solutions have been derived for the multinomial model and Gaussian models using linear classifiers for the minimum mean squared error (MMSE) error estimate [15, 16], the MSE of the error estimate [17, 18], and an optimal Bayesian classifier (OBC) relative to the prior distribution [19, 20], the latter being expressed in terms of *effective class conditional distributions*, which are expectations relative to the posterior distribution of the class-conditional distributions. The closed-form solutions depend on particular models (multinomial and Gaussian) and the existence of conjugate priors, which can be too constraining for practical applications such as RNA-Seq classification.

Much of the statistical literature concerning classification of RNA-Seq data attempts to address differential expression testing, that is, univariate statistical testing on an individual gene basis. These attempts typically model RNA-Seq data via negative binomial [2, 79] and Poisson distributions [70]. In addition, network inference has been attempted using a hierarchical Poisson log-normal model [33], and clustering of RNA-Seq data points has utilized various approaches [83, 77]. However, in clinical settings one is often interested in sample classification: the problem of

classifying the RNA-Seq data from unlabeled patients using a set of labeled training data. One of the few RNA-Seq-specific attempts towards this goal uses a Poisson modeling assumption with independent features [89]. The Poisson model is completely parameterized by its mean and thus is known to exhibit problems in fitting RNA-Seq data due to the overdispersion typically observed in such datasets.

In this section, we focus on modeling the pipeline that starts with extracting the gene concentrations from the biological samples and their subsequent processing by the sequencing instrument [36]. This is accomplished using a hierarchical, multivariate Poisson model (MP). Specifically, gene concentration levels are modeled by a log-normal distribution and the sequencing instrument sampling of those is modeled via a Poisson process. This allows us to accurately model the RNA-Seq data overdispersion as demonstrated by marginal variance calculations and posterior predictive model diagnostics in Section 3.2.5. In addition, this hierarchical model allows for inferring any covariance structure observed between the features.

Whereas Dalton and Dougherty have presented a computational method for nonlinear classifiers in the Gaussian model [14], this still depends upon conjugate priors. In this work, we remove the constraints imposed by the requirement of a closed-form solution by developing the optimal Bayesian classifier using a Markov-chain-Monte-Carlo (MCMC) methodology. This provides a computational framework for calculating the OBC for any parameterized class conditional-density and any prior distribution. Most notably, this allows us to use distributions designed to closely model particular datasets and a prior distribution of any form to improve classification performance in small-sample settings, in particular, for RNA-Seq-based classification.

3.2 Methods

3.2.1 Notation

Throughout, we use capital letters to indicate random variables, lower case letters to indicate individual realizations of random variables or indices, bold latin characters for observed vectors, and Greek letters for latent features and parameters. We write $p(\mathbf{X})$ as the probability measure over the random variable $\mathbf{X}$. $p(\mathbf{X})$ may be a probability mass function, probability density function, or arbitrary probability measure. $p(\mathbf{x}|y)$ denotes the conditional probability $p(\mathbf{X} = \mathbf{x}|Y = y)$. Similarly, following Bayesian convention, we write parameterized distributions by conditioning on the parameter, for instance, $p(\mathbf{X}|Y, \theta)$, and posterior expectations by conditioning on the sample, such as $E[\mathbf{X}|Y, S_n]$, where S_n and all other values are defined in Section 3.2.2. If it is unclear which density an expectation is taken with respect to, then we denote it in subscript notation, such as $E_{\theta|S_n}[\cdot]$, where the expectation is taken with respect to the density $p(\theta|S_n)$.

3.2.2 Review of optimal Bayesian classification

Binary classification considers a set of n labeled training data points, $S_n = \{(\mathbf{x}, y)\}_1^n$, where $y \in \{0, 1\}$ is the class label and $\mathbf{x} \in \mathcal{X}$ is the feature vector over a feature space $\mathcal{X}$. An example of binary classification in a clinical setting might include class 0 and 1 being two types of cancers, or normal and cancerous tissues. Available features would then be the gene or genes that will eventually be used in the designed classifier to assign this label. The feature space $\mathcal{X}$ would be the set of possible gene expression measurements for all genes in the feature vector. The labeled training data S_n would be the set of gene expression measurements from samples which had undergone further testing (possibly observation with the passage of time, cell culturing, or more invasive followup procedures) to identify the type or

malignancy of the tissue. Using S_n, we design a classifier ψ that hopefully performs well on the unknown joint feature-label distribution $p(\mathbf{X}, Y)$. In the same clinical example, the classifier ψ could then identify the type of cancer using gene expression measurements alone.

By parameterizing this unknown joint distribution in a model-based Bayesian framework one can derive an optimal Bayesian classifier (OBC) that minimizes the expected error over the space of all classifiers under assumed forms of the class-conditional densities. Specifically, under Gaussian and multinomial class-conditional densities and their corresponding conjugate prior distributions, closed-form solutions for the OBC [19, 20] and the first two moments of the error estimate conditioned on the sample [17, 18] have been obtained.

The parameterization of the feature-label distribution consists of the marginal class probability c and the class-conditional densities $p(\mathbf{x}|y, \theta_y)$, where a particular value $\theta_y \in \Theta_y$ specifies a single class-conditional density contained in the class of densities defined over the space Θ_y, which will be a Cartesian product as described in Section 3.2.4. Therefore, for a two-class problem, we specify a parameterized joint feature-label distribution as $\theta = (c, \theta_0, \theta_1) \in \Theta = [0, 1] \times \Theta_0 \times \Theta_1$. In the Bayesian classification framework, these values are then treated as random variables, so that we may consider quantities such as the expectation of c, or another random variable conditioned on the value of the parameter vector θ.

Fig. 3.1 describes the inter-relationships between the quantities of interest in the general theoretic framework of Bayesian classification. The tree shows a subset of the derivations possible from the posterior feature-label parameter distribution to the OBC classifier and error estimates. Specifically, directed edges indicate that the child can be derived from the parent by performing the operation indicated by the edge label. Closed-form solutions of the quantities highlighted in grey have

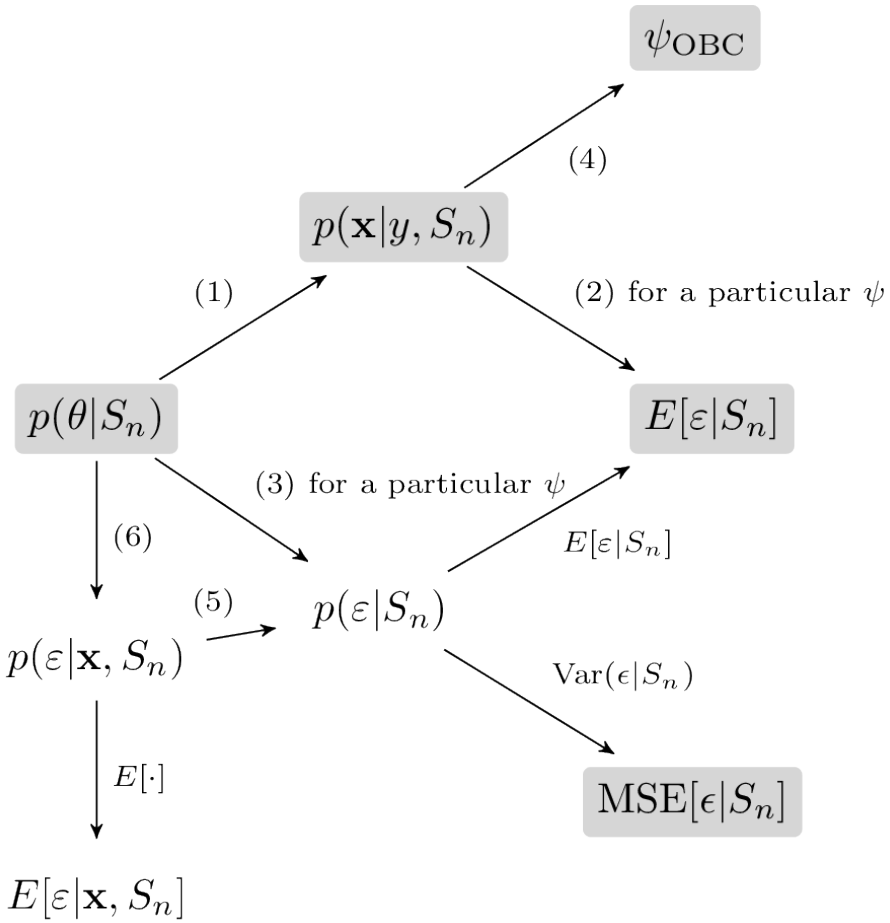

Figure 3.1: A Bayesian classification derivation tree summarizing the relationships between several important quantities in the general theoretical framework of Bayesian classification. A directed edge between a parent and its child indicates that the child can be derived from the parent by the equations indicated in the edge label. The root of the tree $p(\theta|S_n)$ is the posterior distribution of the feature label parameters and by taking expectations with respect to this distribution, we can derive the effective class conditional densities $p(\mathbf{x}|y, S_n)$ and the distribution of the classifier error $p(\varepsilon|S_n)$. Then these quantities give rise to the OBC, and MMSE and MSE estimates for the error as described in the text. Quantities highlighted in grey are given in closed form for Gaussian and multinomial distributions in [17]. Reproduced with permission from [57].

been calculated for the Gaussian and multinomial feature-label distributions [15, 16].

As in those derivations, the tree assumes independence between the marginal class

probability c and the class-conditional parameters θ_y. In addition, the posterior of c is assumed known throughout the tree. Fig. 3.1 demonstrates a primary benefit of the Bayesian approach to classification. Once we obtain the posterior distribution of the class-conditional parameters, it is straightforward to calculate many relevant quantities through appropriately crafted conditional expectations. In this section we demonstrate how to approximate any quantity in the tree for arbitrary class conditional densities and arbitrary prior distributions.

We now examine the tree in more detail. Starting at the far left of the tree, $p(\theta|S_n)$ is the posterior distribution of the parameterized feature-label distribution – posterior to the labeled samples in S_n. Typically, error estimates and the optimal classifier are our primary interest, so that this posterior distribution is traditionally used as a means to compute other quantities and is not of interest by itself.

The *effective class-conditional density* is the marginal predictive posterior of the feature vector $\mathbf{X}$ conditioned S_n and the class variable Y,

$$p(\mathbf{x}|y, S_n) = \int_{\Theta_y} p(\mathbf{x}|y, \theta_y) p(\theta_y|S_n) d\theta_y. \tag{3.1}$$

It gives the distribution of the feature vector using a weighted average over all the parameterized class-conditional densities in Θ_y given a class y. The weights in this expectation are the posterior, $p(\theta_y|S_n)$, evaluated at each θ_y.

The true error of classifier ψ is $\varepsilon = p(\psi(\mathbf{X}) \neq Y)$. Given the sample data S_n, ε is a random unknown quantity in the Bayesian framework. The MMSE estimate given

in [17] can be written as

$$\begin{aligned} E[\varepsilon|S_n] &= p(\psi(\mathbf{X}) \neq Y|S_n) \\ &= E_{\theta|S_n}[p(\psi(\mathbf{X}) \neq Y|\theta, S_n)] \\ &= \hat{c}\varepsilon_0(\theta_0, \psi) + (1-\hat{c})\varepsilon_1(\theta_1, \psi) \\ &= \int_{\mathcal{X}} (\hat{c}p(\mathbf{x}|0, S_n)\mathbf{I}_{\mathbf{x}\in R_1} \\ &\quad + (1-\hat{c})p(\mathbf{x}|1, S_n)\mathbf{I}_{\mathbf{x}\in R_0})d\mathbf{x}, \end{aligned} \tag{3.2}$$

where $\mathbf{I}_A$ is the indicator function for event A, $\hat{c} = E[c|S_n]$ is the posterior expectation of c, R_y is the region of the feature space the classifier predicts to be class y, $\mathcal{X}$ is the feature space, and $\varepsilon_y(\theta_y, \psi)$ is the error of classifier ψ contributed by class y on the fixed distribution θ_y.

We can also obtain the full posterior distribution of the error,

$$\begin{aligned} p(\varepsilon|S_n) &= \int_{\Theta} p(\varepsilon|\theta)p(\theta|S_n)d\theta \\ &= E_{\theta|S_n}[p(\varepsilon|\theta)], \end{aligned} \tag{3.3}$$

where $p(\varepsilon|\theta)$ is the true error for a fixed feature-label distribution and fixed classifier. We denote this deterministic function by $\varepsilon(\theta, \psi)$. As shown in Fig. 3.1, the MMSE estimate and the sample conditioned MSE for this error can also be calculated using the first two moments of the error distribution.

With the MMSE estimator defined, the optimal Bayesian classifier (OBC) is the classifier minimizing the expected error by pointwise minimization of the integral

(3.2) [20]:

$$\psi_{\mathrm{OBC}}(\mathbf{x}) = \begin{cases} 0 & \text{if } \hat{c}p(\mathbf{x}|0, S_n) \geq (1-\hat{c})p(\mathbf{x}|1, S_n), \\ 1 & \text{otherwise.} \end{cases}. \tag{3.4}$$

3.2.3 Conditional error estimator

If the true feature-label distribution were known, then we could compute the true error of a classifier exactly as an expectation over the conditional error [30]:

$$\varepsilon = p(\psi(\mathbf{X}) \neq Y) = \int_{\mathcal{X}} p(\psi(\mathbf{x}) \neq Y|\mathbf{x})p(\mathbf{x})d\mathbf{x}.$$

Treating ε as a random variable, one can similarly derive its posterior distribution by conditioning on the feature vector:

$$\begin{aligned} p(\varepsilon|S_n) &= \int_{\mathcal{X}} p(\varepsilon, \mathbf{x}|S_n)d\mathbf{x} \\ &= \int_{\mathcal{X}}\int_{\Theta} p(\varepsilon, \theta, \mathbf{x}|S_n)d\theta d\mathbf{x} \\ &= \int_{\Theta} p(\theta|S_n)\int_{\mathcal{X}} p(\varepsilon|\mathbf{x}, \theta)p(\mathbf{x}|S_n)d\mathbf{x}d\theta, \end{aligned} \tag{3.5}$$

which is different than the derivation of the same quantity in (3.3).

This introduces the idea of the *conditional error estimator,* which we define as the MMSE estimate of the classification error conditioned on the feature vector $\mathbf{x}$,

$$\begin{aligned} \hat{\varepsilon}(\psi, \mathbf{x}) &= E_{\theta|S_n}[\varepsilon|\mathbf{x}, S_n] \\ &= p(\psi(\mathbf{x}) \neq Y|\mathbf{x}, S_n) \\ &= \frac{p(\mathbf{x}|Y \neq \psi(\mathbf{x}), S_n)p(Y \neq \psi(\mathbf{x})|S_n)}{p(\mathbf{x}|S_n)} \\ &= Z^{-1}p(\mathbf{x}|Y \neq \psi(\mathbf{x}), S_n)p(Y \neq \psi(\mathbf{x})|S_n), \end{aligned} \tag{3.6}$$

as expanded through application of Bayes' theorem, where Z is a normalizing constant given by

$$Z = p(\mathbf{x}|S_n) = \sum_{y \in \{0,1\}} p(\mathbf{x}|y, S_n)p(y|S_n).$$

In addition to being useful in the above alternative derivation of the classifier's error posterior, the conditional error estimate has other practical applications. When classifying an unlabeled data point, we would like to estimate the error of the classifier output for that particular data point, as opposed to the overall error estimate for the classifier.

For the OBC, from (3.4) the conditional error estimator can be written as

$$\hat{\varepsilon}(\psi_{\mathrm{OBC}}, \mathbf{x}) = Z^{-1} \min_{y \in \{0,1\}} \{p(\mathbf{x}|y, S_n)p(y|S_n)\}. \tag{3.7}$$

In sum, using the effective class-conditional densities and the posterior marginal probabilities one can calculate conditional error estimates for points in the feature space in addition to the earlier quantities described.

3.2.4 The multivariate Poisson model

With the widespread use of next-generation sequencing techniques, classification approaches must be developed to account for the discrete nature of the mapped sequence data and to accommodate the various types of prior information available regarding these experiments.

Gene concentration levels can be modeled using a log-normal distribution [5, 4]. As discussed in the introduction, we assume that the sequencing instrument samples this mRNA concentration through a Poisson process and obtains $X_{i,j}$ reads for sample

point i and gene j. We model this as

$$p(X_{i,j}|\lambda_{i,j}) \sim \text{Poisson}(d_i \exp(\lambda_{i,j})), \tag{3.8}$$

where $\lambda_{i,j}$ is the location parameter of the log-normal distribution for sample i and gene j, and d_i is a variable accounting for the sequencing depth as determined by the sequencing process [36]. For each i, we model the location parameter vector λ_i with a multivariate Gaussian distribution, $\lambda_i \sim \text{Normal}(\mu, \Sigma)$. We then consider the mean μ and covariance Σ of the gene concentrations as independent quantities for each class y.

The entire MP model is represented in Fig. 3.2 as a plate diagram. The distribution of a single class y is parameterized by $\theta_y = (\mu, \Sigma, \mathbf{d}, \lambda)$, where $\mathbf{d} = (d_1, \ldots, d_n)$ and $\lambda = (\lambda_{i,j}), i = 1, 2, ...n, j = 1, 2, ..., D$, for n sample points and D total genes. Therefore, $\theta_y \in \Theta_y = \mathbb{R}^D \times \mathbb{R}^{D\times D} \times \mathbb{R}^n \times \mathbb{R}^{D\times n}$. The feature-label distribution parameterization for the two-class problem is then given by $\theta = (c, \theta_0, \theta_1)$, where $c = p(Y = 0)$, the prior probability for class 0.

To ensure a proper posterior with unit integral, we place weakly informative priors over the latent variables in the MP model. In choosing these values, we have aimed to avoid the complications that can occur with overly diffuse priors, such as Lindley's paradox [68, 82]. We choose:

$$\begin{aligned}
\mu_y &\sim \text{Normal}(\eta_y, \nu^2 I_D) \\
\Sigma_y &\sim \text{Inverse-Wishart}(\kappa_y, S_y) \\
c &\sim \text{Beta}(1, 1),
\end{aligned}$$

where each element of μ_y is distributed according to a univariate Gaussian. Unless

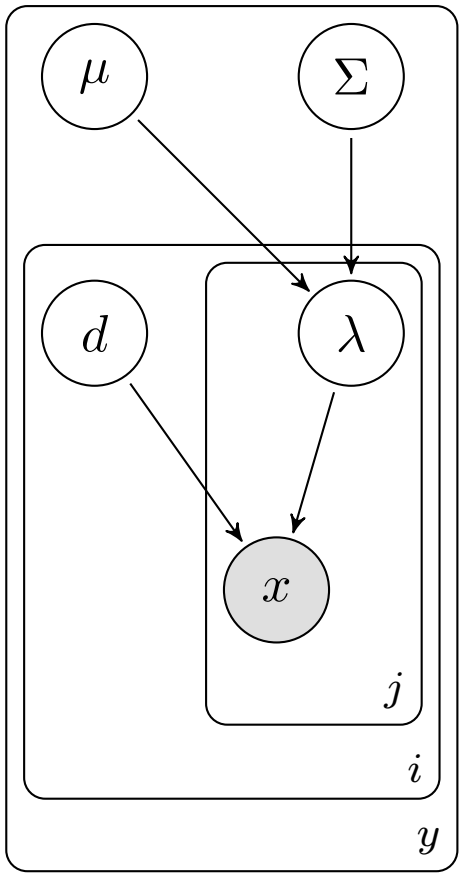

Figure 3.2: Mutivariate Poisson model plate diagram. A plate diagram for the multivariate Poisson model. The outermost plate represents the classes that we are interested in classifying against, where i is the index of the sample in class y, and j are the genes being modeled. Reproduced with permission from [57].

otherwise stated, η is the D dimensional zero vector, $\nu^2 = 25$, $\kappa = 10$, and $S = (\kappa - 1 - D)I_D$. For computational and identifiability reasons, $\mathbf{d}$ is fixed to be a vector of normalization constants in order to match the different sequencing depths across all the samples. In practice, $\mathbf{d}$ can be approximated by an upper quartile normalization, which has been shown to be effective [25].

In any Bayesian approach the choice of prior affects the results, especially when only a few data points are given. In the case of MMSE classifier error estimation

in the Bayesian framework, robustness to incorrect modeling assumptions has been extensively studied in [16] and in those studies performance held up well for various kinds of incorrect modeling assumptions. Robustness of optimal Bayesian classifiers to false modeling assumptions was extensively studied in [20]. Again, good robustness was exhibited. Of course, one can get bad small-sample results by intentionally selecting an inaccurate prior. In general, if one is confident in his knowledge, then a tight prior is called for because tighter priors require less data for good performance; on the other hand, when one is not confident, then prudence calls for a less informative prior. As proven in [20], OBC classification is consistent under very general conditions; however, a prior whose mass is concentrated far away from the true parameters will perform worse than one that is non-informative. These issues have been extensively discussed in the Bayesian literature [50, 51, 6, 32]. In the end, performance is the measure of worth and our results with synthetic and real data indicate solid performance for the modeling approach used herein.

3.2.5 Overdispersion

The MP model uses the Poisson distribution in a hierarchical scheme. It is important to note that, while the read counts are modeled as *conditionally* Poisson in equation 3.8, the observed read counts are not *marginally* Poisson distributed. To demonstrate this, consider a one-dimensional simplification of the MP model in which X is the number of reads observed,λ is the log of the RNA concentration, and

$$\lambda \sim \text{Normal}(\mu, \sigma^2)$$

$$X \sim \text{Poisson}(\exp(\lambda)).$$

Then for the marginal variance of X,

$$\begin{aligned}\mathrm{Var}(X) &= \mathrm{E}[\mathrm{Var}(X|\lambda)] + \mathrm{Var}(\mathrm{E}[X|\lambda]) \\ &= e^{(\mu+\sigma^2/2)} + (e^{\sigma^2}-1)e^{(2\mu+\sigma^2)} \\ &\geq e^{\mu} = \mathrm{Var}(\mathrm{Poisson}(e^{\mu}))\end{aligned}$$

where μ and σ^2 are the mean and variance of the log of the concentration. Therefore, when $\sigma^2 > 0$, the marginal variance of X is always greater than that of a Poisson random variable with the same effective rate.

In addition, by carrying out a posterior predictive model check [34, p. 143] by computing marginal posterior p-values against real RNA-Seq data, we can quantitatively assess the ability of the MP model to fit the dispersion of the TCGA data. For a test statistic T, we compute the p-value by comparing the test statistic on the true data $T(S_n)$ and the value of the statistic averaged across the posterior predictive distribution $T(x^{rep})$, where $x^{rep} \sim p(x|S_n)$:

$$\begin{aligned}p_T &= \Pr(T(x^{rep}) \geq T(S_n)|S_n) \\ &= \Pr(T(x^{rep}) > T(S_n)|S_n) \\ &\quad + (0.5)\Pr(T(x^{rep}) = T(S_n)|S_n) \\ &\approx \frac{1}{M}\sum_{i=0}^{M} \mathbf{I}\{T(x^{rep(s)}) > T(S_n)\} \\ &\quad + 0.5\mathbf{I}\{T(x^{rep(s)}) = T(S_n)\},\end{aligned}$$

where $x^{rep(s)}$ are Monte Carlo samples taken from the posterior predictive distribution $p(x|S_n)$ using the M Monte Carlo samples from the posterior distribution of θ as described in Section 4.2.2. The term $(0.5)\Pr(T(x^{rep}) = T(S_n)|S_n)$ is necessary due

to the discrete nature of RNA-Seq data. P-values away from 0 and 1 indicate that the model posterior produces test statistics both above and below that measured on the real data.

We also consider where the real test statistic falls in relation to credible intervals of the test statistic to consider the magnitude of any differences. We apply the inter-quartile distance test statistic to provide a measure of the MP model's ability to fit the dispersion of RNA-Seq data. We also consider several other test quantities in Appendix C.

3.2.6 Prior calibration using discarded features

Since designed classifiers typically use very few of the totality of observed genes, only a small fraction of the data is used for classifier design. Similarly to [14], we can use the discarded features to calibrate the inverse-Wishart prior for our MP OBC. Our goal is to obtain hyperparameters $S, \mathbf{m}, \kappa$, and ν^2 for each class from our training data S_n. In general, we do not expect the discarded features to give us information about any particular genes and the specific covariances between genes, so we make the simplifying assumptions that we learn information from the discarded genes in an aggregate sense. Thus, we consider the following structure on the hyperparameters: $\mathbf{m} = m[1, 1, \ldots, 1]^T$ and

$$S = \sigma^2 \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \cdots & 1 \end{bmatrix},$$

where $m \in \mathbb{R}$, $\sigma^2 > 0$, and $-1 \leq \rho \leq 1$. For each class, we need to determine values for five scalar quantities: m, ν^2, σ^2, ρ, and κ.

Due to the hierarchical design of the MP model, we cannot apply the method

of moments in a direct fashion, as did [14]. Instead, we utilize a sampling based approach to the method of moments. This MCMC sampling approach has been examined in [9] as an extension to the generalized method of moments [42]. The sampling approach uses the discarded features in an additional MCMC run evaluated prior to the primary classification MCMC procedure as discussed in Section 4.2.2 – and then proceeds to the method of moments. In this calibration MCMC, we initialize all prior distributions with flat priors and use the discarded features to obtain samples from the posterior distribution of μ and Σ. Typically, the number of discarded features F is much larger than the dimensionality D of the classification problem. Therefore, due to computation time, we uniformly sample F_s pairs of features from F and average the resulting runs rather than using all or large groups of discarded features in a single MCMC run. We use the following procedure (for the complete algorithm, see Appendix A):

1. For each randomly chosen discarded feature pair (s in total):

   (a) Obtain MCMC samples using the feature pair as data and flat priors.

   (b) Record posterior averages of μ and Σ.

2. Average over these posterior averages as given by eqs.(3.15)-(3.19).

3. Using the resulting five hyperparameter estimates, run the final MCMC for classification.

Following [14], we use the moments of the posterior samples to determine the hyperparameters through the following relations: The mean of an inverse-Wishart distribution is

$$\mathrm{E}[\Sigma] = \frac{S}{\kappa - D - 1}, \tag{3.9}$$

which together with our simplified covariance structure implies

$$\sigma^2 = (\kappa - D - 1)\mathrm{E}[\Sigma_{11}], \tag{3.10}$$

$$\rho = \frac{\mathrm{E}[\Sigma_{12}]}{\mathrm{E}[\Sigma_{11}]}. \tag{3.11}$$

The variance of the first diagonal of an inverse-Wishart matrix can be used to solve for κ via

$$\kappa = \frac{2(\mathrm{E}[\Sigma_{11}])^2}{\mathrm{Var}(\Sigma_{11})} + D + 3. \tag{3.12}$$

As we have samples of μ directly from our posterior, we obtain

$$m = \mathrm{E}[\mu_1], \tag{3.13}$$

$$\nu = \mathrm{Var}[\mu_1]. \tag{3.14}$$

In order to use equations (3.9)-(3.14), we obtain estimates of the moments from MCMC performed over the F_s discarded feature pairs. For the i-th feature pair we obtain the posterior means $\widehat{\mu}_1^{(i)}, \widehat{\Sigma}_{11}^{(i)}$, and $\widehat{\Sigma}_{12}^{(i)}$ and then average:

$$\widehat{\mathrm{E}}[\mu_1] = \frac{1}{F_s}\sum_{i=1}^{F_s} \hat{\mu}_1^{(i)} \tag{3.15}$$

$$\widehat{\mathrm{Var}}[\mu_1] = \frac{1}{F_s - 1}\sum_{i=1}^{F_s} (\widehat{\mathrm{E}}[\mu_1] - \hat{\mu}_1^{(i)})^2 \tag{3.16}$$

$$\widehat{\mathrm{E}}[\Sigma_{11}] = \frac{1}{F_s}\sum_{i=1}^{F_s} \frac{\hat{\Sigma}_{11}^{(i)} + \hat{\Sigma}_{22}^{(i)}}{2} \tag{3.17}$$

$$\widehat{\mathrm{E}}[\Sigma_{12}] = \frac{1}{F_s}\sum_{i=1}^{F_s} \hat{\Sigma}_{12}^{(i)} \tag{3.18}$$

$$\widehat{\mathrm{Var}}[\Sigma_{11}] = \frac{1}{F_s - 1}\sum_{i=1}^{F_s} (\widehat{\mathrm{E}}[\Sigma_{11}] - \hat{\Sigma}_{11}^{(i)})^2. \tag{3.19}$$

We substitute the estimates from Eqs. (3.15)-(3.19) back into Eqs. (3.9)-(3.14) to obtain the final hyperparameter estimates.

One must keep in mind that the calibration procedure explicitly assumes the MP model. Hence, one can only expect an improvement in the classification performance if the data follow the MP model.

3.2.7 Computation

To obtain the MP OBC, we approximate the effective class conditional densities in order to minimize the expected error in a pointwise fashion:

$$\begin{aligned} p(\mathbf{x}|y, S_n) &= \int_{\Theta_y} p(\mathbf{x}|y, \theta_y) p(\theta_y|S_n) d\theta_y \\ &\approx \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{x}|y, \theta_y^{(i)}), \end{aligned} \tag{3.20}$$

where $\theta_y^{(i)}$ are M samples of θ_y from the model posterior distributions.

For clarity of presentation, we do not consider the class variable y, and we assume a single class. We do this because the computation can be performed per-class due to the assumed independence between the classes and the marginal probability, $p(c, \theta_0, \theta_1) = p(c)p(\theta_0)p(\theta_1)$.

To obtain posterior samples of θ using the Metropolis Hastings MCMC algorithm we define a proposal distribution $p(\theta'|\theta)$ to obtain a new value for the class parameters θ' from the old values θ. We then calculate the acceptance ratio

$$R = \min\left\{1, \frac{p(\theta'|S_n)p(\theta'|\theta)}{p(\theta|S_n)p(\theta|\theta')}\right\} = \min\left\{1, \frac{p(S_n|\theta')p(\theta')}{p(S_n|\theta)p(\theta)}\right\},$$

under the assumption of a symmetric proposal distribution ($p(\theta'|\theta) = p(\theta|\theta')$). The process of proposing and accepting samples from this distribution with the proba-

bility R induces a Markov chain. Positivity of the proposal distribution ($p(\theta'|\theta) > 0$ for any θ) is a sufficient condition for ergodicity of this Markov chain. Furthermore, this Markov chain admits a steady-state distribution equal to our desired posterior distribution $p(\theta|S_n)$ [38].

From the definition of the likelihood,

$$\begin{aligned} p(S_n|\theta) &= \prod_{i=1}^{n} p(\mathbf{x}_i|\theta) = \prod_{i=1}^{n} p(\mathbf{x}_i|\lambda_i) \\ &= \prod_{i=1}^{n}\prod_{d=1}^{D} p(x_{i,d}|\lambda_{i,d}), \end{aligned}$$

where $p(\mathbf{x}_i|\theta) = p(\mathbf{x}_i|\lambda_i)$ owing to conditional independence. From the definition of the prior,

$$\begin{aligned} p(\theta) &= p(\mu, \Sigma, \lambda) \\ &= p(\lambda|\mu, \Sigma)p(\mu|\Sigma)p(\Sigma) \\ &= \prod_{i=1}^{n} p(\lambda_i|\mu, \Sigma)p(\mu|\Sigma)p(\Sigma). \end{aligned}$$

The posterior predictive distribution in (3.20) is approximated by

$$\begin{aligned} p(\mathbf{x}|S_n) &\approx \frac{1}{M}\sum_{i=1}^{M} p(\mathbf{x}|\theta^{(i)}) \\ &= \frac{1}{M}\sum_{i=1}^{M}\int_{\Lambda} p(\lambda|\theta^{(i)})p(\mathbf{x}|\lambda)d\lambda \\ &= \frac{1}{M}\sum_{i=1}^{M}\int_{\Lambda} p(\lambda|\theta^{(i)})\prod_{k=1}^{D} p(x_k|\lambda_k)d\lambda \\ &\approx \frac{1}{MT}\sum_{i=1}^{M}\sum_{g=1}^{T}\prod_{k=1}^{D} p(x_k|\lambda_k^{(g)}), \end{aligned}$$

where, $p(\mathbf{x}_k|\lambda_k) \sim \text{Poisson}(d_k \exp(\lambda_k))$, $\lambda \sim \text{Normal}(\mu, \Sigma)$, $\Lambda = \mathbb{R}^{n \times D}$, and the $\lambda^{(g)}$ are T vector-valued samples drawn from the appropriate class's posterior distribution used to approximate the inner intractable integral. In addition, we use this approximation of the effective class-conditional density to calculate the conditional error estimates of (3.7) in a pointwise fashion.

Finally, because we have assumed a conjugate prior distribution for the marginal class probability c, the posterior expectation takes the closed form

$$\mathrm{E}_{\theta|S_n}[c] = \frac{n_0 + \alpha_0}{n_0 + n_1 + \alpha_0 + \alpha_1},$$

where the n_y are the number of training samples obtained from class y and the α_y are hyperparameters set to 1 for an uninformative prior. Conjugacy was used for this one parameter because the increased flexibility of the full sampling approach was deemed not necessary due to the constrained, univariate nature of the parameter. If more complex relationships between c and other parameters were desired, then a sampling approach using non-conjugate priors would be straightforward to implement.

3.2.8 Synthetic data

To evaluate OBC performance in the setting of the MP model, we generate synthetic data using the method proposed in [47] to simulate gene expression/mRNA concentrations (see Appendix B). These gene expression values are then statistically sampled to emulate modern sequencing machines as described in [36]. Parameter values are drawn from the following distributions to examine a wide variety of clas-

sification problems:

$$\begin{aligned}
\mu_y &\sim \text{Normal}(0, 0.2), \\
\sigma_y &\sim \text{Inverse-Gamma}(1, 3), \\
\rho &= \text{Uniform}(0.0, 1.0), \\
d_{\text{low}} &= 9, \\
d_{\text{high}} &= 11, \\
\text{blocksize} &= 5.
\end{aligned}$$

With these parameters, ten global, twenty heterogeneous, and ten non-marker features are generated. Then four features are randomly chosen to represent a mixture of features of various classification quality. Following [36], the features in the data are zero mean and unit standard deviation normalized except for the MP OBC. The exception occurs because the MP model expects features to be positive integers and normalization is not necessary. The discarded features are used for calibration of the MP OBC priors, and 3000 samples are generated from each class to estimate the true classification rate for each classifier.

We use four features in this synthetic data classification study owing to limited computational resources as discussed in Section 3.3.3.

The synthetic data generation method proposed in [47] imposes the strong assumption of a homogeneous covariance (HC) structure between the two classes of data. This assumption does not hold for biological situations where interactions between features are not necessarily preserved between classes, and this occurs frequently in biology when considering the possible effects of canalizing genes, nonlinear gene regulation, and mutations in the case of cancer [71, 28]. Specifically, if the canal-

izing gene is not observed, and differs in activity between the two classes, then the measured correlation between two downstream genes could potentially be negligible in one class while strong in the other class. Similarly, for highly nonlinear gene regulation, if a gene in one class is in the saturation region of its response curve from a master gene, then the correlation will be low, while a lower expression level in the other class would allow for a large measured correlation with the same canalizing gene. And finally, if one class represents normal gene expression and the other tumor-related expression, then a correlation might exist from a functioning pathway in the normal tissue, but a mutation could result in a lack of correlation effects in the tumor.

Hence, we modify the synthetic data generation procedure in an attempt to produce synthetic datasets more representative of such nonlinear phenomena in biology. In this modified procedure, we allow independent covariance (IC) matrices for the features of the two classes. To generate these covariance matrices, Σ_y, we utilize independent draws from inverse-Wishart distributions with parameters $\kappa_y = 22, D = 20$, and scale matrix $S = \sigma_y^2(\kappa - 1 - D)I_D$. To examine the effects of feature correlation in IC datasets, we can also generate low-correlation covariance matrices by zeroing the off-diagonal terms. Once the covariance matrix for class y is obtained, location parameters for gene-expression values for each sample point are drawn from the respective multivariate normal distribution $\lambda_y \sim N(\mu_y, \Sigma_y)$. Each sample point is then assumed to be normalized through an upper quartile or other suitable method, but in practice any sample-based normalization is imperfect. We reflect this variation by drawing the sequencing depth d_i from a Uniform$(d_{\text{low}}, d_{\text{high}})$ distribution, giving the rate of the Poisson process as $d_i \exp \lambda_i$. The number of reads for a single gene from a single sample is then drawn from this Poisson distribution. See Appendix A for more detail.

The OBC is optimal on average across the space of distributions determined by its prior distributions. To avoid biasing the performance comparison, we draw the classification problem datasets using different distributions than those of the OBC priors. See Appendix A for more detail.

3.2.9 Real data

We consider a real RNA-Seq dataset composed of level 3, RNASeqV2 data from the Cancer Genome Atlas (TCGA) project. It contains 484 and 470 specimens from lung adenocarcinoma and lung squamous cell carcinoma tumor biopsies, respectively. The samples are mapped read counts against 20531 known human RNA transcripts as generated by the University of North Carolina at Chapel Hill, one of the Genome Sequencing Centers for the TCGA. The data for each cancer type is the result of processing approximately 20 billion reads and the read count files for each are one gigabyte apiece. The problem is to classify the tumor types. Because the class-0 (lung adenocarcinoma) and class-1 (lung squamous cell carcinoma) sample sizes, 484 and 470, are not chosen randomly, we are confronted with the problem of separate sampling. This means that there is no way to obtain a posterior distribution for c and therefore c must be known in advance. Based upon records from 2006-2010, we have a very accurate estimate, $48,600/141,300 \approx 0.34$ [78]. Whereas we can use the value of c directly, along with all of the data, in designing the OBC, for classification rules that do not use c explicitly, the separately sampled data must be maximally subsampled to the proper sampling ratio c before applying the classification rule [31]. This means that for N_{trn} desired samples, the sample subsets will contain $\text{round}((1-c)N_{trn})$ and $\text{round}(cN_{trn})$ for class 0 and 1, respectively. Moreover, holdout error estimation, which we use here, must be properly adapted for separate sampling for all design methods, including the OBC. The holdout estimate is given by

$$\hat{\varepsilon}_c = c\hat{\varepsilon}_0 + (1-c)\hat{\varepsilon}_1,$$

where $\hat{\varepsilon}_0$ and $\hat{\varepsilon}_1$ are the ordinary holdout estimators (performed on all remaining data samples not used for training) for the class-0 and class-1 errors, respectively [31]. We note that many studies have made the mistake of using classification rules designed for random sampling when sampling is separate. This can have devastating effects on classifier performance [31].

While averaging over sample subsets for holdout error estimation, we also average over uniformly, randomly selected gene subsets of size 4. This sampling occurs from low (1-10 average reads per gene) expression genes. We sample from these lower expression genes because we are ultimately interested in classification problems where the delineation between phenotypes is determined by genes with low expression. We used 10,000 for averaging in order to obtain a large enough sample over this feature and sample subset space to achieve repeatable results (data not shown). Computational runtime for each sample and gene subset was similar to the synthetic data.

3.3 Results and discussion

Appendix A contains a simple two-class, two-feature demonstration of the overall procedure to allow for easy visualization and interpretation. Here we discuss the results for the synthetic and real data.

3.3.1 Synthetic data

To evaluate classification performance, classifiers were trained using 3NN, LDA, and c-support vector machines with a radial basis function kernel [30]. Starting with the homogeneous-covariance model, Fig. 3.3a shows that the performance of

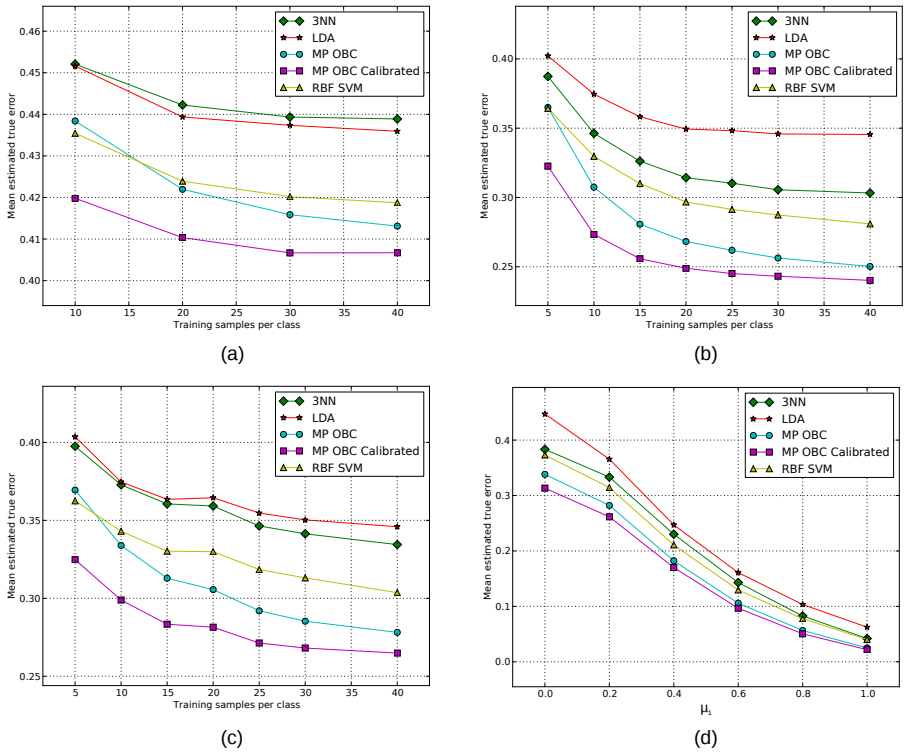

Figure 3.3: Synthetic data classification results with (a) homogeneous-covariance, (b) high correlation independent-covariance, (c) low correlation independent-covariance, and (d) high correlation independent-covariance data at several problem difficulties. Reproduced with permission from [57].

the multivariate Poisson OBC is better than nonlinear SVM when more than 10 samples are available and is significantly better than any other classifier when using calibrated features. Equivalently, by using discarded features, we can obtain the same classification performance while requiring fewer training samples.

In the case of independent-covariance data with highly correlated features, Fig. 3.3b shows superior classification performance of the MP OBC at nearly all sample sizes considered. In addition, for calibrated prior distributions, the performance of the MP OBC improves. This improvement is greater when the sample sizes are small, which demonstrates the importance of additional knowledge (through discarded features)

when data are expensive to obtain or not readily available.

The superior performance of the OBC relative to LDA, 3NN, and SVM in Fig. 3.3b is on account of classification optimization relative to the model, which characterizes prior information. To further investigate OBC improvement, we again considered heterogeneous covariance matrices but with independent features to determine if there is any difference in the relative performance between the classifiers. In fact, the results provided in Fig. 3.3c show identical relative performance to the error curves in Fig. 3.3b, thereby indicating that both the standard classifiers and the OBC, relative performance (at least in the case considered) is not affected by whether or not the features are correlated. Indeed, comparing Fig. 3.3a with Figs. 3.3b and 3.3c, we see that the relative performance of SVM, MP OBC, and calibrated MP OBC is the same in both the homogeneous and heterogeneous models. The switch in relative performance between LDA and 3NN between Fig. 3.3a and Figs. 3.3b and 3.3c is not surprising because LDA is optimal for a fixed (known) homogeneous Gaussian model but not for a heterogeneous Gaussian model.

The larger overall classification errors in Fig. 3.3a as compared to Figs. 3.3b and 3.3c are due to the different covariance matrices generated by the HC and IC models. Each model required different generating distributions for $\{\sigma_y, \rho\}$ and $\{S, \kappa\}$ for the HC and IC cases, respectively, and the particular choices made in Section 3.2.8 resulted in larger dispersions and higher errors in the HC models than the IC models. To demonstrate this, we tested LDA with 1000 training and testing samples across 1000 random generating distributions and found the average HC classification error to be 0.41 and the IC error to be 0.32. This is despite LDA being optimal for homogeneous, fixed, known Gaussian cases and sub-optimal for heterogeneous, fixed, known Gaussian cases, where the former is similar to the HC case.

These differences in overall error rates are also consistent with the intuition that

larger differences between class covariance matrices induce higher classification error rates. This is seen in: the low classification error rates in Fig. 3.3c where all elements of the covariance matrices are not shared between the two classes, the high error rates in Fig. 3.3a where the covariance matrices are identical between the two classes, and the error rates of Fig. 3.3b falling in between the previous two figures where only the diagonal elements of the covariance matrices differ between the two classes and all off-diagonal elements are shared (and zero).

Still using independent-covariance data, we fixed the mean of class 0 at $\mu_0 = 0.0$ in Fig. 3.3d, and varied μ_1 from 0.0 to 1.0 to make the classification problem harder and easier, respectively. Across this range of classification problems, the MP OBC had better classification performance than the other classification methods. In addition, calibrated priors improved performance further, especially for harder classification problems.

3.3.2 Real data

In Table 3.1, we chose ten genes at random from adenocarcinoma tumor TCGA samples and performed model diagnostics [34, p. 143] by calculating posterior predictive p-values for interquartile distance (IQR) as a measure of dispersion. In the Appendix C, we provide additional test statistics and graphical predictive posterior model diagnostics. These results indicate that RNA-Seq overdispersion is modeled sufficiently with the MP model.

In Fig. 3.4, we see mean holdout errors averaged over 10,000 training sets and testing sets of TCGA data as described in Section 3.2.9. Here the MP OBC performs better than all other classifiers across most training sample sizes considered, but calibration does not improve performance for this particular dataset. Recall that improvement owing to calibration depends on the extent to which the data satisfy

Table 3.1: Posterior predictive model diagnostics are given for 10 randomly selected genes from adenocarcinoma TCGA samples. Inter-quartile distance (IQR) is used as a robust measure of dispersion. In the table, $IQR(S_n)$ is the training data's IQR, followed by the 95-th credible interval, and the posterior predictive P-value. In cases where the P-value is close to 0 or 1, the true test statistic's distance from the 95-th credible interval can be used to determine the magnitude of the mis-fit. Reproduced with permission from [57].

Gene ID	IQR(S_n)	95% int. for IQR(x^{rep})	p-value
UPK1A|11045	2.12	$[1.0, 3.0]$	0.09
OR4P4|81300	0.00	$[0.0, 0.0]$	0.50
PCDHA12|56137	139.22	$[107.8, 187.0]$	0.54
MDS2|259283	1.85	$[2.0, 5.0]$	1.00
AXIN2|8313	347.69	$[331.5, 439.3]$	0.85
DYNLT1|6993	848.41	$[830.0, 1043.3]$	0.90
RARA|5914	786.43	$[706.8, 881.3]$	0.62
TMEM194A|23306	396.06	$[367.0, 471.3]$	0.76
AGPS|8540	496.45	$[505.8, 636.5]$	0.97
NLRP2|55655	854.47	$[381.3, 677.5]$	0.00

the MP model. If the aim of this section were to build an operational classifier based on the TCGA data, then we would have to go back and extensively study the data set to examine deviations from the model – for instance, outliers; however, here our aim is to show the functionality of the OBC with non-Gaussian data based on MCMC and apply it to the MP model. The fact that the MP OBC performs well on the real data satisfies this aim. Calibration is a tricky business and it would be a major separate study to characterize the manner in which model variation affects calibration, even if we were to perform an intensive study of this particular data set. Performance on the synthetic data demonstrates the effectiveness of the calibration when the model is satisfied.

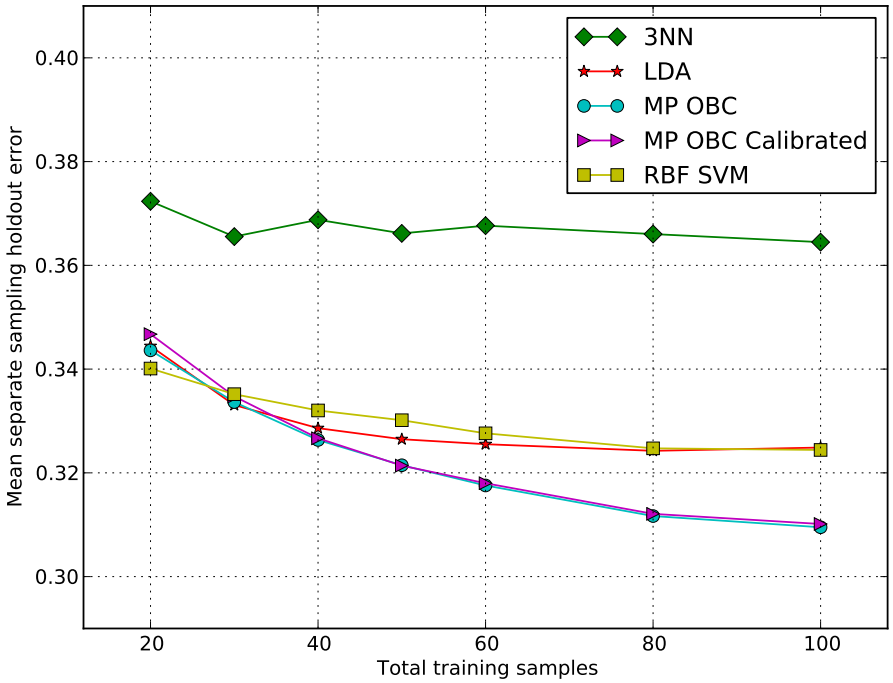

Figure 3.4: TCGA RNA-Seq Classification. Average holdout errors were computed over 10,000 training sets and feature subsets using two types of lung cancer RNA-Seq data from TCGA. MP OBC with and without calibrated priors demonstrates superior performs across a range of training sample sizes. In addition, providing the MP OBC with calibrated priors does not appear to improve performance in this particular dataset. Reproduced with permission from [57].

3.3.3 Computational limitations

The results in Fig. 3.3 and Fig. 3.4 required tens of thousands of MCMC runs. Owing to limited available computational resources, we could only allocate around 30 seconds on a single CPU core for each MCMC run. This necessitated using only four genes for these classification results as each iteration of the MCMC procedure has time complexity of $O(D^3)$, where D is the number of features. In practice, one would have a small number of data sets and could use parallel computing to devote more time and computing effort for the classification. For example, in timescales on the order of hours on a typical workstation, we have successfully performed classification

using 50 genes.

The other classification methods compared in this study have smaller computational requirements and can correspondingly handle larger numbers of features given the same available resources. However, for the small sample sizes often available in biology, 50 genes is typically beyond the "peaking" point where most classifiers decrease in classification performance as more features are added (for a fixed number of training samples) [48]. Incidentally, the OBC does not suffer this "peaking phenomenon" as shown in [19].

In addition, the computational time requirements of classification is typically not a bottleneck in translational medicine given the timescales used in collecting biological data. In these settings, the accuracy of classification is much more valuable than rapid runtimes, and this is the primary advantage of the computational OBC framework proposed in this section.

3.4 Conclusions

We have demonstrated that Bayesian classification can be applied to specific problem domains such as RNA-Seq through statistical modeling and MCMC computation. The resulting classifier provides superior classification performance compared to state-of-the-art classifiers such as SVM with a radial basis kernel. Although we have not discussed error estimation – our interest in the present section being classification, *ipso facto*, the MCMC approach to optimal Bayesian classification can be applied, via [15, 16] and [17, 18], to obtain optimal MMSE error estimators for any classification rule and sample-conditioned evaluation of the MSE for error estimation.

In the next section, we extend this work to the error estimation in the optimal Bayesian error estimate and mean square error of the Bayesian error estimate. We then use these two estimators for feature selection as a method to find gene sets

which well separate biological phenotypes.

4. DETECTING MULTIVARIATE GENE INTERACTIONS IN RNA-SEQ DATA USING OPTIMAL BAYESIAN CLASSIFICATION*

In the previous section, we developed the multivariate Poisson statistical model of sequencing data and built an Optimal Bayesian Classifier relative to that model for sequencing datasets. In this section, we consider optimal error estimation using the same hierarchical Poisson model as a biological discovery tool through feature selection.

4.1 Introduction

RNA-Seq is a high-throughput technique for measuring the gene expression profile of a target tissue or even single cells. Due to its increased accuracy and flexibility over microarray technologies, it is widely applied in biological fields to uncover the transcriptional mechanisms at play in a given physiology or phenotype.

Typically, this analysis involves mapping the RNA-Seq reads to a reference genome, quantifying transcript expression, and then performing testing for differential gene expression to determine which genes are expressed at significantly different levels in the phenotypes being compared. Tools such as Cufflinks [86], edgeR [79], and DE-Seq2 [69] provide these univariate statistical tests using well characterized univariate statistical models of gene expression.

However, one is often interested in phenotypes which can only be differentiated by the state of several genes simultaneously. These multivariate relationships cannot be detected using univariate testing procedures only. Instead, it is necessary to consider the joint expression patterns between multiple genes simultaneously and the ability

*Parts of this section were reproduced with permission from Knight, J.M.; Ivanov, I.; Dougherty, E.R. "Detecting Multivariate Gene Interactions in RNA-Seq Data Using Optimal Bayesian Classification", *under review* [58] © 2015.

to use this joint expression to differentiate the phenotypes of interest.

While this problem can be approached using the study of multivariate statistical testing, we instead opt to utilize the theory of statistical classification for two primary reasons. First, translational medicine aims to apply scientific knowledge to improve medical practice, and classification's prediction of phenotypes from gene expression data is well aligned with this goal. Secondly, the model-based approach used in optimal Bayesian classification allows for the use of prior biological knowledge to improve results in the small number of samples typically available in biological studies.

Here, we employ the optimal Bayesian classifier and optimal Bayesian error estimator to quantify the relationship between the joint gene expression information and phenotypes of interest. We begin in Section 4.2.1 by reviewing optimal error estimation with optimal Bayesian classification. Section 4.2.2 explains our approach to computation using Monte Carlo techniques including Markov Chain Monte Carlo. Then Section 4.3 describes the dietary intervention study dataset and discusses the overall study design. Section 4.4.2 discusses the results of the computational study, and Section 4.4.3 considers the biological implications of the top performing gene sets.

4.2 Methods

4.2.1 Optimal Bayesian classification

Binary classification considers a set of n labeled training data points, $S_n = \{(\boldsymbol{x}, y)\}_1^n$, where $y \in \{0, 1\}$ is the class label and $\boldsymbol{x} \in \mathcal{X}$ is the feature vector over a feature space $\mathcal{X}$. Using S_n, we design a classifier ψ based on data from the unknown joint feature-label distribution $p(\boldsymbol{X}, Y)$. By parameterizing this unknown joint distribution in a model-based Bayesian framework one can derive an optimal Bayesian

classifier (OBC) that minimizes the expected error over the space of all classifiers under assumed forms of the class-conditional densities [19, 20]. While extending this framework to the multiple class classification problem is straightforward, this section utilizes the two-class problem formulation to aid biological interpretation.

We parameterize the feature-label distribution into the marginal class probability $c = p(y = 0)$ and the class-conditional densities $p(\mathbf{x}|y, \boldsymbol{\theta}_y)$, where a particular value $\boldsymbol{\theta}_y \in \Theta_y$ specifies a single class-conditional density and for a two-class problem $\boldsymbol{\theta} = (c, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) \in \Theta = [0, 1] \times \Theta_0 \times \Theta_1$. In the Bayesian classification framework, the components of $\boldsymbol{\theta}$ are treated as random variables, so that we may consider quantities such as the expectation of c, or the conditional expectation of some other quantity conditioned on the value of the parameter vector $\boldsymbol{\theta}$.

4.2.1.1 The optimal Bayesian error estimate

The true error of a classifier ψ can be written as $\varepsilon = p(\psi(\mathbf{X}) \neq Y)$. Given the sample data S_n, one can utilize a Bayesian framework and compute the posterior distribution $p(\varepsilon|S_n)$. Additionally, one can consider the conditional expectation of the posterior $\mathbb{E}[\varepsilon|S_n]$, which is taken with respect to the model posterior distribution $p(\boldsymbol{\theta}|S_n)$. Keeping in mind that, in the Bayesian framework, the true error is a function of both $\boldsymbol{\theta}$ and S_n, this conditional expectation provides an optimal estimate of the true error of the designed classifier relative to mean-square error (MSE) with respect to the joint distribution of $\boldsymbol{\theta}$ and S_n [15, 16]. This minimum mean-square error (MMSE) estimate is known as the *optimal Bayesian error estimate* (BEE) and is defined by

$$\hat{\varepsilon} = \mathbb{E}[\varepsilon|S_n] = \mathbb{E}_{\boldsymbol{\theta}|S_n}[p(\psi(\mathbf{X}) \neq Y|\boldsymbol{\theta}, S_n)].$$

Optimality follows directly from classical MSE theory. Moreover, according to that theory the BEE is an unbiased estimate of the true error relative to the sampling distribution.

With the BEE defined, the optimal Bayesian classifier (OBC) for binary classification is given by [19]

$$\psi_{\mathrm{OBC}}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \hat{c}p(\boldsymbol{x}|y=0,S_n) \geq (1-\hat{c})p(\boldsymbol{x}|y=1,S_n), \\ 1 & \text{otherwise.} \end{cases}$$

where $\hat{c} = \mathbb{E}[c|S_n]$ is the expected posterior marginal class probability. The OBC is the classifier minimizing the expected error through pointwise minimization.

4.2.1.2 Uncertainty quantification for the optimal Bayesian error estimate

In addition to the BEE estimate, one is often interested in the uncertainty associated with the estimate. This quantification of uncertainty captures the inaccuracy of our modeling assumptions, the noise in the data, and the amount of data that we possess. It is given by the posterior variance of the error ε:

$$\begin{aligned} \mathrm{Var}(\varepsilon \mid S_n) &= \mathbb{E}_{\boldsymbol{\theta}|S_n}[(\varepsilon(\boldsymbol{\theta}) - \hat{\varepsilon})^2] \\ &= \int_{\boldsymbol{\Theta}} \varepsilon(\boldsymbol{\theta})^2 p(\boldsymbol{\theta}|S_n)\mathrm{d}\boldsymbol{\theta} - \hat{\varepsilon}^2. \end{aligned}$$

This conditional variance is equal to the conditional MSE of the BEE (BEEMSE) as an estimator of the true error given the sample [17]:

$$\mathrm{MSE}(\hat{\varepsilon}|S_n) = \mathbb{E}_{\boldsymbol{\theta}|S_n}[(\hat{\varepsilon} - \varepsilon)^2|S_n)].$$

Section 4.2.2 considers the efficient computation of this quantity.

4.2.2 Computation

Using the multivariate Poisson model of Section 3, the goal is to obtain the OBC, BEE, and BEEMSE given a labeled RNA-Seq dataset. The posterior distribution $p(\boldsymbol{\theta}|S_n)$ of $\boldsymbol{\theta}$ is sufficient for this; however, the hierarchical multivariate Poisson (MP) model is not conjugate. Thus, no known analytical closed form solution exists and we must instead sample from the posterior using MCMC utilizing the prior distribution and likelihood function [57],

$$p(S_n|\boldsymbol{\theta}) = \prod_y \prod_{i=1}^{n_y} \prod_{d=1}^{D} p(x_{y,i,d}|\lambda_{y,i,d}),$$

where $S_{n_y}^y$ are n_y number of training samples from class y only, $\boldsymbol{x}_{y,i}$ are feature values of the training samples from class y, $\boldsymbol{\lambda}_y$ are the λ values from class y, and $p(\boldsymbol{x}_{y,i}|\boldsymbol{\theta}_y) = p(\boldsymbol{x}_{y,i}|\boldsymbol{\lambda}_{y,i})$ owing to conditional independence. The prior can also be decomposed using the assumed independence between the classes and conditional independence between training samples given the model parameters as [57]

$$p(\boldsymbol{\theta}) = \prod_y \prod_{i=1}^{n} p(\boldsymbol{\lambda}_{y,i}|\boldsymbol{\mu}_y, \Sigma_y) p(\boldsymbol{\mu}_y|\Sigma_y) p(\Sigma_y).$$

Using this form of the prior distribution and likelihood, we obtain samples of $\boldsymbol{\theta}$ from the posterior distribution using Adaptive Metropolis-within-Gibbs Markov Chain Monte Carlo. As in [57], we approximate the effective class conditional density:

$$p(\boldsymbol{x}|y, S_n) \approx \frac{1}{T_{\boldsymbol{\theta}}} \sum_{i=0}^{T_{\boldsymbol{\theta}}} p(\boldsymbol{x}|\boldsymbol{\theta}^{(i)}, y), \tag{4.1}$$

where $\boldsymbol{\theta}^{(i)}$ are the $T_{\boldsymbol{\theta}}$ samples drawn using MCMC. The OBC can then be calculated

point-wise. The BEE of the OBC can also be determined using the effective class conditional density:

$$\begin{aligned}\hat{\varepsilon} &= \int_{\mathcal{X}} \sum_{y} \boldsymbol{I}_{\psi(\boldsymbol{x})\neq y} p(y, \boldsymbol{x}|S_n)\mathrm{d}\boldsymbol{x} \\ &= \int_{\mathcal{X}} \min_{y} p(\boldsymbol{x}|, y, S_n)p(y|S_n)\mathrm{d}\boldsymbol{x}\end{aligned}$$

However, this integral is difficult to compute numerically due to the time taken to evaluate the integrand (4.1) and the discrete, yet large, nature of the integration space, which poses problems for traditional quadrature routines.

Instead, we reconsidered the integrand to obtain

$$\begin{aligned}\hat{\varepsilon} &= \int_{\mathcal{X}} \min_{y} p(y|\boldsymbol{x}, S_n)p(\boldsymbol{x}|S_n)\mathrm{d}\boldsymbol{x} \\ &\approx \frac{1}{T_{\boldsymbol{x}}} \sum_{i=1}^{T_{\boldsymbol{x}}} \min_{y} p(y|\boldsymbol{x}^{(i)}, S_n) \\ &\approx \frac{1}{T_{\boldsymbol{x}}} \sum_{i=1}^{T_{\boldsymbol{x}}} \min_{y} \left[\frac{p(\boldsymbol{x}^{(i)}|y, S_n)p(y|S_n)}{\sum_y p(\boldsymbol{x}^{(i)}|y, S_n)p(y|S_n)} \right],\end{aligned}$$

where the $\boldsymbol{x}^{(i)}$ are the $T_{\boldsymbol{x}}$ samples drawn from the effective conditional densities from both classes. This integration is straightforward to compute as drawing from the effective conditional density is equivalent to the efficient process of drawing samples from the posterior samples of $\boldsymbol{\theta}$.

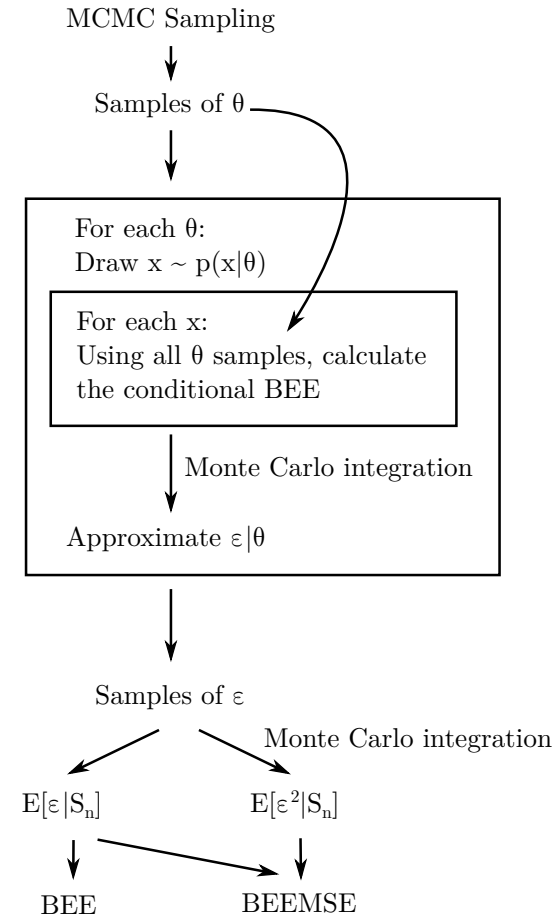

Figure 4.1: BEEMSE calculations utilize MCMC sampling from the posterior of $\boldsymbol{\theta}$. Then for each sample of $\boldsymbol{\theta}$, $\varepsilon|\boldsymbol{\theta}$ is approximated using a draw of $\boldsymbol{x}$ from $p(x|y,\boldsymbol{\theta})$. Then the conditional BEE error is computed for each of these in order to form a Monte Carlo approximation to $\varepsilon|\boldsymbol{\theta}$. Then these approximations are again used in a Monte Carlo integration step to approximate $\hat{\varepsilon}$ and $\mathbb{E}[\varepsilon^2]$.

Computing the BEEMSE requires the first moment ($\hat{\varepsilon}$) and the second moment,

$$\begin{aligned}\mathbb{E}[\varepsilon^2|S_n] &= \int_{\boldsymbol{\Theta}} \varepsilon(\boldsymbol{\theta})^2 p(\boldsymbol{\theta}|S_n)\mathrm{d}\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\Theta}} \left[\int_{\mathcal{X}} \sum_{y} \boldsymbol{I}_{\psi(\boldsymbol{x})\neq y} p(y|\boldsymbol{x},\boldsymbol{\theta}) p(\boldsymbol{x}|\boldsymbol{\theta})\mathrm{d}\boldsymbol{x}\right]^2 \\ &\quad \times p(\boldsymbol{\theta}|S_n)\mathrm{d}\boldsymbol{\theta}.\end{aligned}$$

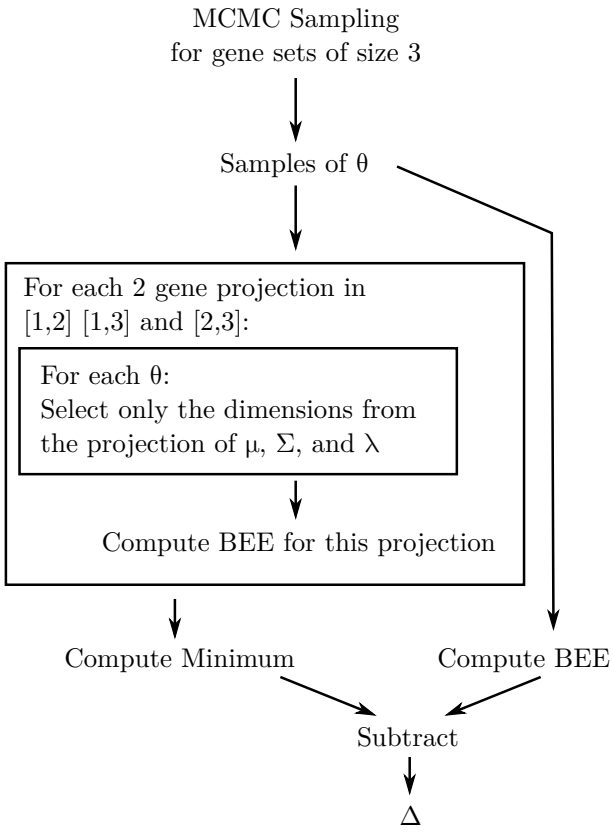

Figure 4.2: Computation of Δ is sensitive to Monte Carlo approximation error so naive calculations of each error quantity are inadequate. Instead we used the above scheme where the main insight is that the BEE computation for each gene subset must be made using the same MCMC samples of $\boldsymbol{\theta}$ but projected down to the appropriate dimension. This results in the Δ quantities shown in Fig. 4.8.

By using a Monte Carlo approximation for the above integrals and simplifying for binary classification,

$$\mathbb{E}[\varepsilon^2|S_n] \approx \frac{1}{T_{\boldsymbol{\theta}}} \sum_{i=0}^{T_{\boldsymbol{\theta}}} \left[\frac{1}{T_{\boldsymbol{x}}} \sum_{j=0}^{T_{\boldsymbol{x}}} p(\breve{y}|\boldsymbol{x}^{(j)}, \boldsymbol{\theta}^{(i)}) \right]^2 ,$$

where $\breve{y} = \arg\min_y p(y|\boldsymbol{x}, S_n)$. This process is shown in Fig. 4.1.

We now define the quantity Δ to be the reduction of classification error by adding an additional feature to the classification problem. Consider a classification problem

using three genes with $\hat{\varepsilon}_3 = \alpha$. Given the classification errors $\hat{\varepsilon}_2(i)$ for the i classification problems using two gene subsets of the three original features, we define

$$\Delta = \min_i \hat{\varepsilon}_2(i) - \hat{\varepsilon}_3 \tag{4.2}$$

Naive computations of Δ can easily be dominated by the error from Monte Carlo approximations. A robust way of computing Δ can be computed using the process in Fig. 4.2.

4.2.3 Normalization

Normalization of RNA-Seq datasets is important for reproducible differential expression testing and many methods have been proposed and tested [25]. We computed the upper quartile normalization factor as a surrogate for sequencing depth and used that quantity for each sample's d parameter. We call this normalization approach a "model" based normalization.

To perform draws from the posterior predictive distribution $p(x|y, S_n)$, the mean of the normalization factor of training samples from class y is used as the value for the average sequencing depth factor d.

We also compared the classification errors using this method of normalization against the raw data and a normalized count approach where the upper quartile normalized counts were scaled up by the average normalization factor and rounded back to integers. We denote the two approaches as the "raw" and "count" techniques respectively.

4.2.4 MCMC convergence diagnostics

It is a well understood limitation of MCMC that it is not possible to determine if the Markov Chain has reached convergence in a finite number of iterations [67]. We

employ a convergence diagnostic to check against simple forms of non-convergence. Here we use the Gelman-Rubin statistic [8] and calculate the potential scale reduction factor (PSRF) for each element of our MP model. We run five parallel MCMC chains, each with 10000 iterations, 2000 burn-in samples and a 1/50 sub-sampling ratio. We assume convergence for each element of the MP model when $|1 - \text{PSRF}| < 0.05$ [13].

4.3 Dietary intervention study

A preclinical dietary study was carried out to determine the interplay between poly-unsaturated fatty acids (PUFA) and the short chain fatty acid, butyrate, which is generated by fiber fermentation in the intestine. Rats were treated with dietary fish oil plus the fermentable fiber, pectin or with a control (non chemoprotective) diet containing corn oil plus cellulose. Six rats per treatment group were then injected with the colon-specific carcinogen azoxymethane (AOM) to investigate protective dietary effects during cancer progression. Comparisons of particular interest include the fish oil/pectin and corn oil/cellulose AOM groups (fpa-cca), fish oil/pectin and corn oil/pectin AOM groups (fpa-cpa), and the corn oil/pectin and corn oil/cellulose (cpa-cca) AOM groups.

An average of 38M 50bp single-end Illumina reads were obtained per treatment group with averages of 33M, 42M, and 41M from the fpa, cca, and cpa groups, respectively. Spliced alignment was performed against the rat genome (rn5) using the STAR aligner [26], and the resultant alignments were further processed with HTSeq-count [3] to perform reference annotation-based transcript assembly using standard parameters. Differential expression analyses were performed with edgeR [79] and DESeq [69].

Classification was performed as shown in Fig. 4.3. Using prior biological knowledge, 858 genes were selected for investigation with this dataset. To further aid

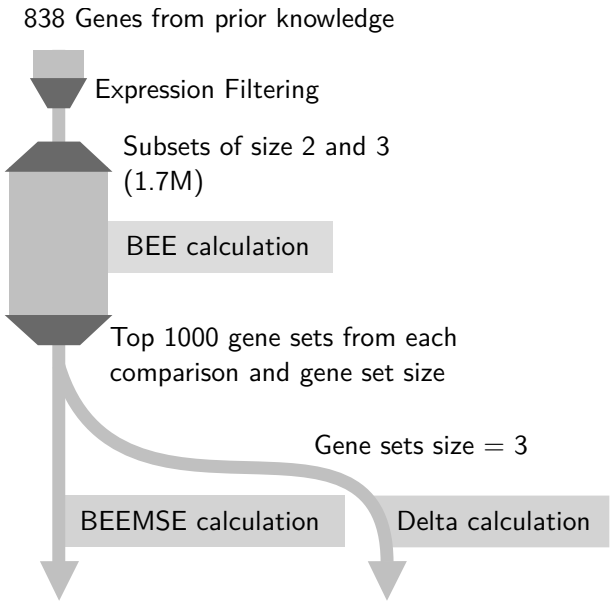

Figure 4.3: Classification of 858 genes from prior knowledge was performed with an expression filtration step, then BEE calculations were performed on all 1.7M gene sets across the three comparisons and two dimensionalities (sets of two and three genes). Then the lowest 1000 classification error sets were selected from each comparison and run in additional BEEMSE and Δ calculations.

biological interpretation, we filtered out genes with an FPKM value and average read count of less than one in both groups of the comparison. Moreover, we only considered genes where the absolute value of the log fold change between the groups was greater than 0.3.

This filtering reduced the list of genes to be evaluated from the previously selected 858 to 185, 58, and 159 in the fpa-cpa, cpa-cca, and fpa-cca comparisons, respectively. Computing all two and three gene subsets of these three comparisons resulted in 1.7M BEE calculations to be performed over 200 cores over a period of several days.

Subsequently, the top 1000 gene sets from each comparison and dimensionality (two and three) were additionally used to perform BEEMSE and Δ computations.

The computations described in Section 4.2.2 require knowledge or an estimate

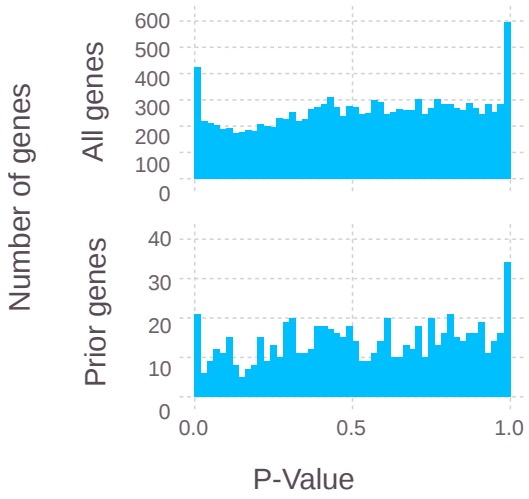

Figure 4.4: For the fpa-cca comparison the above histogram for all 12,000 genes and for the 858 genes in the prior knowledge gene list show that the majority of genes in the prior knowledge list set are not differentially expressed and have a distribution of P-values to the entire dataset.

for the value of the parameter c. Because of the specific experimental design and the purpose of the OBC classification, i.e. to examine sets of genes that can discriminate well between the experimentally generated phenotypes, we assume $c = 0.5$. This choice reflects the experimental design that makes no *a priori* preference towards the classes being compared. Thus, we considered the error contribution from each class as equally important.

4.4 Results

4.4.1 Differential expression analysis

To establish a point of comparison Fig. 4.4 shows the distribution of un-adjusted P-values for the entire 12,000 genes in the fpa-cca comparison and of the 858 genes used in classification. The distribution of P-values for the 858 genes shows that most

Table 4.1: The top ten differentially expressed genes of the 858 gene list by adjusted P-value as reported by DESeq in the fpa-cca comparison.

Gene	Adjusted P-Value
Hoxa2	0.0001
Fabp1	0.0061
Nucks1	0.0085
Plaa	0.0227
P4hb	0.0227
Il6st	0.0250
Pax6	0.0517
Aldh2	0.0774
Ccndbp1	0.0784
Rxra	0.0867

genes were not differentially expressed and the distribution of P-values was similar to that of the entire dataset.

Table 4.1 shows the top 10 genes from the 858 gene list as reported by adjusted P-value from DESeq. Using a traditional 0.05 threshold, only six genes would be considered statistically differentially expressed.

4.4.2 Overall error distributions

The overall distributions of classification errors from a random sampling of the 300M possible gene sets from the 858 prior-knowledge-selected genes are given in Fig. 4.5 split across the number of genes used and the phenotype comparison.

Classification errors in the cpa-cca comparison are significantly higher than the other two groups. This matches previous qPCR data (not published) that also indicated greater transcriptional differences in animals fed dietary fat as opposed to fiber. It can also be noted that the three-gene sets show slightly lower classification errors on average than the two-gene sets.

Fig. 4.6 shows the relationship between the BEE and the BEEMSE across this

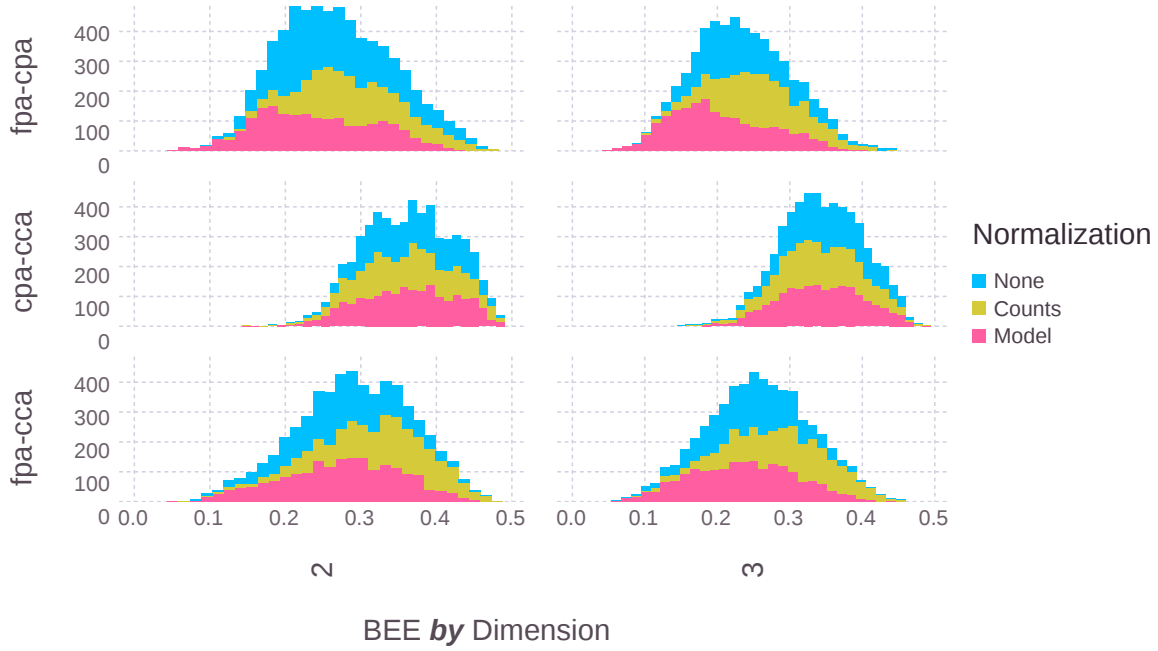

Figure 4.5: The overall classification error distributions are shown by the number of features (panel x-axis), dietary comparison (panel y-axis), and normalization type (stacked plot colors). Average classification error is slightly lower (0.28 vs 0.30) as expected for classification with 3 genes when compared against two genes. Additionally, the three dietary comparisons (oil and fiber types (FP/CC), oil only (FP/CP), and fiber only (CP/CC)), showed differences in average classification performance.

dataset. The figure shows that the BEE and BEEMSE are tightly correlated at low BEE values, and this correlation diminishes at higher values of the BEE.

More gene sets in the lower left of Fig. 4.6 indicates that the fpa-cca and fpa-cpa comparisons are well separated by a larger combination of genes than the cpa-cca comparison. The fewer number of gene sets in the lower left for the cpa-cca comparison indicates that the transcriptional differences between the phenotypes are small, we are considering the wrong set of genes, or the data for these phenotypes might contain a higher level of noise.

Table 4.2: The top four lowest classification error gene sets for each of the three comparisons and for two and three genes. In addition, the Δ value for the three gene comparisons gives the reduction of classification error when adding the third gene to the best performing gene subset of size two. Errors, MSEs, and Δs below 0.01 are not displayed here due to the larger relative effects of Monte Carlo error at these value ranges.

Gene	Gene	Gene	P-Value	P-Value	P-Value	Comparison	$\hat{\varepsilon}$	$\text{Var}(\varepsilon \mid S_n)$	Δ
Arg2	Adamts1		0.11	0.82		FPA-CCA	< 0.01	< 0.01	
Adamts1	Lgals3bp		0.82	0.69		FPA-CCA	< 0.01	< 0.01	
Fabp1	Arg2		0.00	0.11		FPA-CCA	< 0.01	< 0.01	
Fgfr1	Adamts1		1.00	0.82		FPA-CCA	< 0.01	< 0.01	
Arg2	Adamts1		0.11	0.82		FPA-CPA	< 0.01	< 0.01	
Adamts1	Lgals3bp		0.82	0.69		FPA-CPA	< 0.01	< 0.01	
Fabp1	Arg2		0.00	0.11		FPA-CPA	0.0125	< 0.01	
Fabp1	Scd1		0.00	0.97		FPA-CPA	0.0127	< 0.01	
Ccne1	Crabp2		0.99	0.93		CPA-CCA	0.1219	< 0.01	
Ccne1	Bmp3		0.99	0.29		CPA-CCA	0.1311	< 0.01	
Fabp1	Crabp2		0.00	0.93		CPA-CCA	0.1348	< 0.01	
Ccne1	Tnfrsf12a		0.99	0.66		CPA-CCA	0.1366	< 0.01	
Fabp1	Pde4b	P2rx2	0.00	0.45	0.95	FPA-CCA	< 0.01	< 0.01	< 0.01
Fabp1	Pde4b	Scd1	0.00	0.45	0.97	FPA-CCA	< 0.01	< 0.01	< 0.01
Fabp1	Pde4a	Pde4b	0.00	0.71	0.45	FPA-CCA	< 0.01	< 0.01	< 0.01
Fabp1	Pde4b	Arg2	0.00	0.45	0.11	FPA-CCA	< 0.01	< 0.01	< 0.01
Fabp1	Pde4b	P2rx2	0.00	0.45	0.95	FPA-CPA	< 0.01	< 0.01	< 0.01
Fabp1	Pde4a	Pde4b	0.00	0.71	0.45	FPA-CPA	< 0.01	< 0.01	< 0.01
Fabp1	Pde4b	Scd1	0.00	0.45	0.97	FPA-CPA	< 0.01	< 0.01	< 0.01
Fabp1	Pde4b	Fgfr1	0.00	0.45	1.00	FPA-CPA	< 0.01	< 0.01	< 0.01
Ccne1	Fabp1	Crabp2	0.99	0.00	0.93	CPA-CCA	0.0916	< 0.01	0.0312
Ccne1	Bmp3	Crabp2	0.99	0.29	0.93	CPA-CCA	0.0945	< 0.01	0.0348
Ccne1	Fabp6	Dpep1	0.99	0.38	0.98	CPA-CCA	0.0969	< 0.01	0.0399
Ccne1	Dpep1	Abcb1a	0.99	0.98	0.34	CPA-CCA	0.0969	< 0.01	0.0449

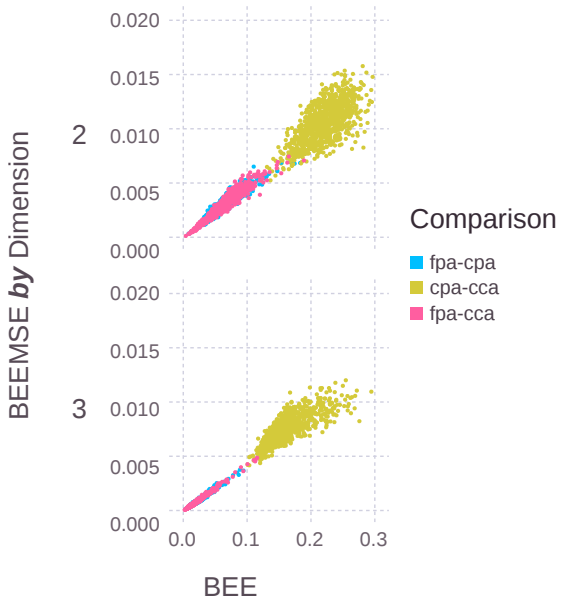

Figure 4.6: For the best 1000 gene sets for each dietary comparison, the BEEMSE tends to increase as a function of BEE.

As more genes are used for classification, the points shift to the left and down as the dataset becomes more separable (if any such separation exists).

The top four gene pairs from each classification grouping are shown in Table 4.2. Genes such as Fabp1 are present both in this list for classification and in Table 4.1 as a differentially expressed gene for the fpa-cca comparison. Most other genes, however, have non-significant adjusted P-values, such as Arg2 (P=0.11, adjusted P=1.0) and Adamts1 (P=0.82, adjusted P=1.0), yet together can have classification errors of less than 1%. This is illustrated along with the OBC decision boundary in Fig. 4.7.

For classifications using gene sets of size three, we compute Δ to show the amount of classification improvement by adding an additional gene. Fig. 4.8 shows the distribution of Δ for the three comparison groups. Because the cpa-cca comparison

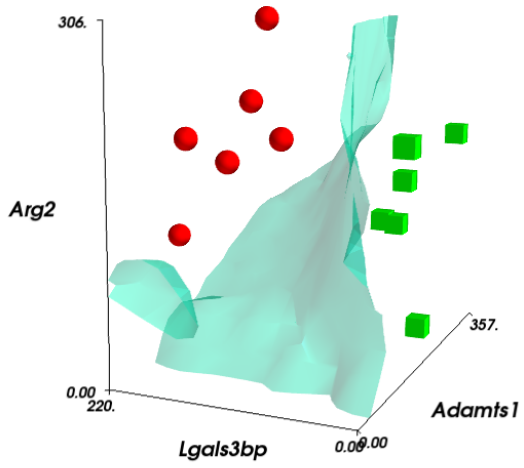

Figure 4.7: Normalized count expressions are shown for the three genes Arg2, Lgals3bp, and Adamts1. The cubes and spheres indicate the fpa and cca samples, respectively. Using the marching cubes contouring algorithm, an approximate rendering of the nonlinear OBC decision boundary is also displayed.

has the highest classification errors it shows the largest improvements (> 0.02) by adding an additional feature to the classification problem.

One concern with using approximation algorithms is whether the computation is sufficiently repeatable. To address this, we ran the top 200 gene subsets from each comparison twice and computed the correlation of BEE estimates from the two runs. A Pearson correlation coefficient of 0.999 across the six comparisons indicated that the computation is repeatable.

Fig. 4.9 shows the comparison between no normalization, count-based normalization, and model-based normalization. The normalized counts show an increase in classification error, potentially due to the 2% rounding error induced from the final integer conversion.

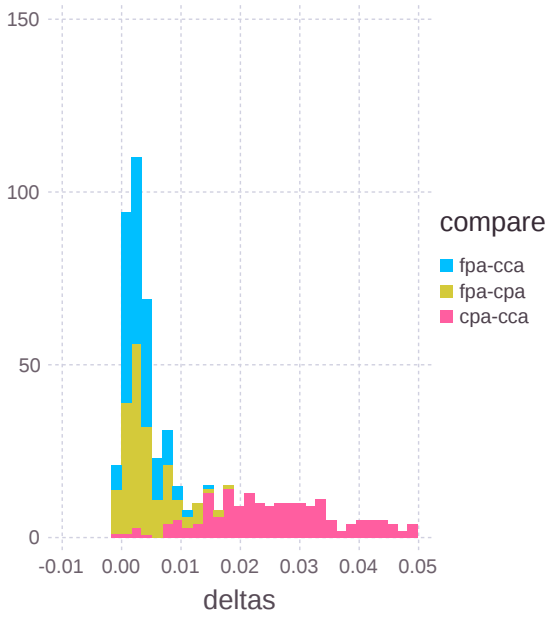

Figure 4.8: The distribution of Δ for the top 1000 gene sets of the three comparison groups. CPA-CCA has the largest Δ values which corresponds to that comparison having the largest classification errors. Negative values of Δ are due to approximation error.

4.4.3 Biological findings

The interactions of dietary fiber-derived compounds in the colonic lumen can have a substantial impact on the metabolism and kinetics of the colon epithelial cell population and suppress inflammation and neoplasia [12, 59]. It has been proposed by us and others that n-3 PUFA found in fish oil and butyrate (a fiber fermentation product) interact in the colon to profoundly suppress colon cancer [23, 10].

We found that in the chemoprotective treatment containing fish oil and fermentable fiber (FPA-CCA), two genes, Fabp1 and Eno3, were prominently detected together with a wide variety of other genes in the gene sets of length three. The calculated 23 fold over-expression of Fabp1 correlates with previous findings classifying

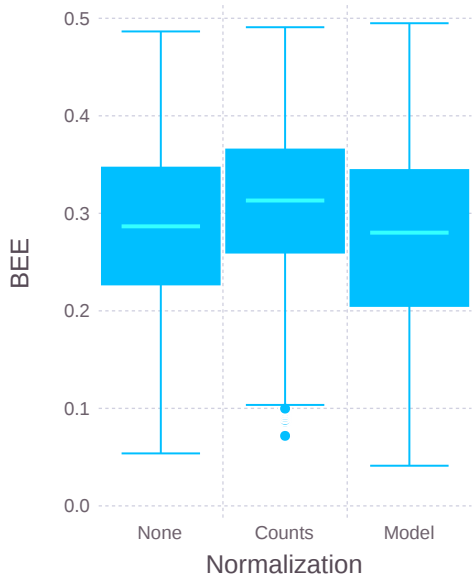

Figure 4.9: Overall classification error varied depending on whether normalization was used (None) and whether it was implemented as a pre-processing step applied to the data (Counts) or input into the model through the sequencing depth variable d (Model).

Fabp1 as a tumor suppressor in colon cancer [62, 80]. Since Fabp1 is a gene involved in the uptake and metabolic processing of PUFA, the higher levels of fish oil derived n-3 PUFA are the most likely cause for its increased expression and chemoprotective activity [73]. The 1.3 fold upregulation of Eno3 is likely due to the fermentable fiber in the diet. Fermentable fiber leads to the generation of the HDAC inhibitor butyrate, which has been previously associated with concurrent increases in enolase 3 (Eno3) levels and differentiation, a hallmark of chemoprotection [85, 87]. The expression levels of Fabp1 and Eno3 are shown along with the OBC classification boundary in Fig. 4.10.

Other genes present with Fabp1 and Eno3 included Ccnd2 (BEE=0.018), Arg2 (BEE=0.004) and Cdk1 (BEE=0.014). Ccnd2, a gene responsible for enhancing can-

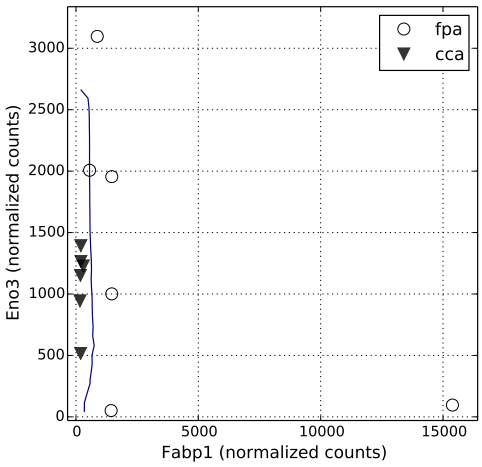

Figure 4.10: Normalized counts of the Fabp1 and Eno3 genes were plotted against each other in relation to the OBC decision boundary (black line) for the fpa-cca comparison.

cer cell proliferation, was downregulated by 1.4 fold in the FPA chemoprotective diet. Similarly, anti-oncogenic genes Arg2 and Cdk1 were also present in low-classification gene sets of size three and these were elevated by 3.34 fold and 1.28 fold, respectively, in the FPA diet. Both of these genes can be upregulated by HDAC inhibitors, a putative mechanism for the chemoprotective nature of these diets [64, 74].

Although these three genes were not considered differentially expressed through individual testing (P-values of 0.11, 0.36, and 0.51 for Arg2, Ccnd2, and Cdk1, respectively), they were indicated as highly predictive in delineating the phenotype when using the Bayesian error estimator. These data suggest that via multivariate interactions with other genes, the BEE highlights novel genes for the purposes of hypothesis generation or further biomarker development/validation. This supports the biological relevance of the BEE as a useful tool in RNA-seq analysis.

4.5 Conclusion

This work demonstrates an application of the OBC and BEE in identifying multivariate gene interactions in RNA-Seq data for the purpose of differentiating biological phenotypes.

Using 858 genes selected by prior knowledge in a preclinical RNA-Seq dataset, we identified genes that in combination yield low classification errors, whereas each gene individually had a large P-value for differential expression. Thus, the application of our previously developed Bayesian classification framework [57] enables investigators to generate new hypotheses regarding multivariate effects between these gene sets and the observed phenotypes.

In addition, new tools such as BEE-BEEMSE scatter plots offer additional diagnostic visualizations to assess the quality of RNA-Seq data, similarly to a volcano plots, as done in DE testing. Future work needs to be performed on synthetic data sets to better uncover the utility of such a graphical representation of the classification performance. Computing and reporting the classification improvements as represented by the Δ quantity is also of particular interest to biologists as large values could potentially indicate more complex interactions between genes than merely P-values or even BEE alone can.

5. CONCLUSIONS

In this dissertation we laid out two main approaches to utilize prior biological knowledge for the purposes of predictive inference. This progression represents notable advances in computational biology from the data driven methodologies that prevail in the field to those that utilize both data and other sources of knowledge about the biology and measurement methodologies.

The algorithm presented to transform incomplete and potentially conflicting biological pathway knowledge into a stochastic, predictive framework provides a foundation upon which researchers can build or enrich network models. This compliments the data-driven network inference approaches which are common in the network inference literature. In addition, the qualitative validation used provides another technique of testing predictive models utilizing existing literature sources.

Furthermore, we introduced a statistical model for sequencing data, that together with extension and computational approaches to the Bayesian framework of optimal minimum mean square error estimation, allowed us to produce the Optimal Bayesian Classifier for sequencing datasets. This classifier provides a step forward in the field of biological data classification, and is available in a well documented, open source code repository (`https://github.com/binarybana/OBC.jl`). Moreover, it provides a flexible framework for additional improvements to the model, a platform to test additional methods of prior construction from biological data, and an example of the enabling role of computation in directly unlocking optimal estimators with respect to informative statistical models.

We are currently extending this approach to network models and working to addressing several important questions.

- Given an underlying statistical model, which forms of prior information over the model parameters are most informative towards reducing the uncertainty of the different estimators of interest?

- Given a model class, is there an optimal complexity for a given data set or characteristic function?

- For other cost functions (other than mean square error), is the posterior distribution of the model parameters sufficient to derive the optimal estimator?

REFERENCES

[1] R. Albert. Network inference, analysis, and modeling in systems biology. *The Plant Cell Online*, 19(11):3327–3338, 2007.

[2] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):1–12, 2010.

[3] S. Anders, P. T. Pyl, and W. Huber. HTSeq–a python framework to work with high-throughput sequencing data. *Bioinformatics*, 31(2):166–169, 2014.

[4] S. Attoor, E. R. Dougherty, Y. Chen, M. L. Bittner, and J. M. Trent. Which is better for cDNA-microarray-based classification: ratios or direct intensities. *Bioinformatics*, 20(16):2513–2520, 2004.

[5] M. Bengtsson, A. Ståhlberg, P. Rorsman, and M. Kubista. Gene expression profiling in single cells from the pancreatic islets of Langerhans reveals lognormal distribution of mRNA levels. *Genome Research*, 15(10):1388–1392, 2005.

[6] J. O. Berger and J. M. Bernardo. On the development of reference priors. *Bayesian Statistics*, 4(4):35–60, 1992.

[7] U. M. Braga-Neto and E. R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.

[8] S. P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, 1998.

[9] M. Carrasco and J.-P. Florens. Simulation-based method of moments and efficiency. *Journal of Business & Economic Statistics*, 20(4):482–492, 2002.

[10] R. S. Chapkin, J. Seo, D. N. McMurray, and J. R. Lupton. Mechanisms by which docosahexaenoic acid and related fatty acids reduce colon cancer risk and inflammatory disorders of the intestine. *Chemistry and Physics of Lipids*, 153(1):14–23, 2008.

[11] R. Cheong, A. Hoffmann, and A. Levchenko. Understanding NF-κB signaling via mathematical modeling. *Molecular Systems Biology*, 4(1), 2008.

[12] Y. Cho, H. Kim, N. D. Turner, J. C. Mann, J. Wei, S. S. Taddeo, L. A. Davidson, N. Wang, M. Vannucci, R. J. Carroll, et al. A chemoprotective fish oil-and pectin-containing diet temporally alters gene expression profiles in exfoliated rat colonocytes throughout oncogenesis. *The Journal of Nutrition*, 141(6):1029–1035, 2011.

[13] M. K. Cowles and B. P. Carlin. Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

[14] L. A. Dalton and E. R. Dougherty. Application of the Bayesian MMSE estimator for classification error to gene expression microarray data. *Bioinformatics*, 27(13):1822–1831, 2011.

[15] L. A. Dalton and E. R. Dougherty. Bayesian minimum mean-square error estimation for classification error – part I: definition and the Bayesian MMSE error estimator for discrete classification. *IEEE Transactions on Signal Processing*, 59(1):115–129, 2011.

[16] L. A. Dalton and E. R. Dougherty. Bayesian minimum mean-square error estimation for classification error – part II: the Bayesian MMSE error estimator for linear classification of Gaussian distributions. *IEEE Transactions on Signal Processing*, 59(1):130–144, 2011.

[17] L. A. Dalton and E. R. Dougherty. Exact sample conditioned MSE performance of the Bayesian MMSE estimator for classification error – part I: representation. *IEEE Transactions on Signal Processing*, 60(5):2575–2587, 2012.

[18] L. A. Dalton and E. R. Dougherty. Exact sample conditioned MSE performance of the Bayesian MMSE estimator for classification error – part II: consistency and performance analysis. *IEEE Transactions on Signal Processing*, 60(5):2588–2603, 2012.

[19] L. A. Dalton and E. R. Dougherty. Optimal classifiers with minimum expected error within a Bayesian framework – part I: discrete and gaussian models. *IEEE Transactions on Pattern Recognition*, 46(5):1301–1314, 2013.

[20] L. A. Dalton and E. R. Dougherty. Optimal classifiers with minimum expected error within a Bayesian framework – part II: properties and performance analysis. *IEEE Transactions on Pattern Recognition*, 46(5):1288–1300, 2013.

[21] A. Datta and E. Dougherty. *Introduction to genomic signal processing with control*. CRC Press, Boca Raton, FL, 2007.

[22] M. Davidich and S. Bornholdt. Boolean network model predicts cell cycle sequence of fission yeast. *PLoS One*, 3(2):e1672, 2008.

[23] L. A. Davidson, D. V. Nguyen, R. M. Hokanson, E. S. Callaway, R. B. Isett, N. D. Turner, E. R. Dougherty, N. Wang, J. R. Lupton, R. J. Carroll, et al. Chemopreventive n-3 polyunsaturated fatty acids reprogram genetic signatures during colon cancer initiation and progression in the rat. *Cancer Research*, 64(18):6797–6804, 2004.

[24] H. De Jong. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[25] M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, et al. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics*, 14(6):671–683, 2013.

[26] A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.

[27] E. Dougherty and A. Datta. Genomic signal processing: diagnosis and therapy. *Signal Processing Magazine, IEEE*, 22(1):107–112, 2005.

[28] E. R. Dougherty, M. Brun, J. M. Trent, and M. L. Bittner. Conditioning-based modeling of contextual genomic regulation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(2):310–320, 2009.

[29] E. R. Dougherty, A. Zollanvari, and U. M. Braga-Neto. The illusion of distribution-free small-sample classification in genomics. *Current Genomics*, 12(5):333–341, 2011.

[30] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, Hoboken, NJ, 2012.

[31] M. S. Esfahani and E. R. Dougherty. Effect of separate sampling on classification accuracy. *Bioinformatics*, 30(2):242–250, 2014.

[32] M. S. Esfahani and E. R. Dougherty. Incorporation of biological pathway knowledge in the construction of priors for optimal Bayesian classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(1):202–218, 2014.

[33] M. Gallopin, A. Rau, and F. Jaffrézic. A Hierarchical Poisson Log-Normal Model for Network Inference from RNA Sequencing Data. *PloS One*, 8(10):e77503, 2013.

[34] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. CRC Press, Boca Raton, FL, 2013.

[35] S. Gerondakis, R. Grumont, R. Gugasyan, L. Wong, I. Isomura, W. Ho, and A. Banerjee. Unravelling the complexities of the NF-κB signalling pathway using mouse knockout and transgenic models. *Oncogene*, 25(51):6781–6799, 2006.

[36] N. Ghaffari, M. R. Yousefi, C. D. Johnson, I. Ivanov, and E. R. Dougherty. Modeling the next generation sequencing sample processing pipeline for the purposes of classification. *BMC Bioinformatics*, 14(1):307–321, 2013.

[37] S. Ghosh, M. J. May, and E. B. Kopp. NF-κB and REL proteins: evolutionarily conserved mediators of immune responses. *Annual Review of Immunology*, 16(1):225–260, 1998.

[38] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*, volume 2. CRC Press, Boca Raton, FL, 1996.

[39] B. Hanczar and E. R. Dougherty. On the comparison of classifiers for microarray data. *Current Bioinformatics*, 5(1):29–39, 2010.

[40] B. Hanczar and E. R. Dougherty. The reliability of estimated confidence intervals for classification error rates when only a single sample is available. *Pattern Recognition*, 46(3):1067–1077, 2013.

[41] B. Hanczar, J. Hua, and E. R. Dougherty. Decorrelation of the true and estimated classifier errors in high-dimensional settings. *EURASIP Journal on Bioinformatics and Systems Biology*, 2(7):2–13, 2007.

[42] L. P. Hansen and K. J. Singleton. Generalized instrumental variables estimation of nonlinear rational expectations models. *Econometrica*, 50(5):1269–1286, 1982.

[43] M. S. Hayden and S. Ghosh. Signaling to NF-κB. *Genes & Development*, 18(18):2195–2224, 2004.

[44] M. S. Hayden and S. Ghosh. Shared principles in NF-κB signaling. *Cell*, 132(3):344–362, 2008.

[45] B. Hayete, T. Gardner, and J. Collins. Size matters: network inference tackles the genome scale. *Molecular Systems Biology*, 3(1):1–3, 2007.

[46] M. Hecker, S. Lambeck, S. Toepfer, E. Van Someren, and R. Guthke. Gene regulatory network inference: Data integration in dynamic models–A review. *Biosystems*, 96(1):86–103, 2009.

[47] J. Hua, W. D. Tembe, and E. R. Dougherty. Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, 42(3):409–424, 2009.

[48] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, 21(8):1509–1515, 2005.

[49] S. Huang. Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery. *Journal of Molecular Medicine*, 77(6):469–480, 1999.

[50] E. T. Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):227–241, 1968.

[51] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

[52] M. Karnaugh. The map method for synthesis of combinational logic circuits Electrical Engineering. *Technical Operations*, 21(1):593–599, 1953.

[53] S. Kauffman. *The origins of order: Self organization and selection in evolution*. Oxford University Press, USA, 1993.

[54] T. Kawai, O. Adachi, T. Ogawa, K. Takeda, and S. Akira. Unresponsiveness of MyD88-deficient mice to endotoxin. *Immunity*, 11(1):115–122, 1999.

[55] S. Kim, R. La Motte-Mohs, D. Rudolph, J. Zúñiga-Pflücker, and T. Mak. The role of nuclear factor-κB essential modulator (NEMO) in B cell development and survival. *Proceedings of the National Academy of Sciences of the United States of America*, 100(3):1203, 2003.

[56] J. M. Knight, A. Datta, and E. R. Dougherty. Generating stochastic gene regulatory networks consistent with pathway information and steady-state behavior. *IEEE Transactions on Biomedical Engineering*, 59(6):1701–1710, 2012.

[57] J. M. Knight, I. Ivanov, and E. R. Dougherty. MCMC implementation of the optimal Bayesian classifier for non-Gaussian models: model-based RNA-Seq classification. *BMC Bioinformatics*, 15(1):401, 2014.

[58] J. M. Knight, I. Ivanov, and E. R. Dougherty. Detecting multivariate gene interactions in RNA-seq data using optimal Bayesian classification. *Under review*, 2015.

[59] S. Kolar, R. Barhoumi, C. K. Jones, J. Wesley, J. R. Lupton, Y.-Y. Fan, and R. S. Chapkin. Interactive effects of fatty acid and butyrate-induced mitochon-

drial Ca2+ loading and apoptosis in colonocytes. *Cancer*, 117(23):5294–5303, 2011.

[60] B. Kuo. *Digital control systems*. SRL Publishing, Champaign, IL, 1977.

[61] T. Lawrence, M. Bebien, G. Liu, V. Nizet, and M. Karin. IKKα limits macrophage NF-κB activation and contributes to the resolution of inflammation. *Nature*, 434(7037):1138–1143, 2005.

[62] L. Lawrie, S. Dundas, S. Curran, and G. Murray. Liver fatty acid binding protein expression in colorectal neoplasia. *British journal of cancer*, 90(10):1955–1960, 2004.

[63] R. Layek, A. Datta, and E. Dougherty. From biological pathways to regulatory networks. *Molecular BioSystems*, 7(3):843–851, 2011.

[64] V. Leone, D. D'Angelo, I. Rubio, P. M. de Freitas, A. Federico, M. Colamaio, P. Pallante, G. Medeiros-Neto, and A. Fusco. MiR-1 is a tumor suppressor in thyroid carcinogenesis targeting CCND2, CXCR4, and SDF-1α. *The Journal of Clinical Endocrinology & Metabolism*, 96(9):E1388–E1398, 2011.

[65] Q. Li, D. Antwerp, F. Mercurio, K. Lee, and I. Verma. Severe liver degeneration in mice lacking the IB kinase 2 gene. *Science*, 284(5412):321, 1999.

[66] Q. Li, Q. Lu, V. Bottero, G. Estepa, L. Morrison, F. Mercurio, and I. Verma. Enhanced NF-κB activation and cellular function in macrophages lacking IκB kinase 1 (IKK1). *Proceedings of the National Academy of Sciences of the United States of America*, 102(35):12425, 2005.

[67] F. Liang, C. Liu, and R. Carroll. *Advanced Markov chain Monte Carlo methods: learning from past samples*. John Wiley & Sons, Hoboken, NJ, 2011.

[68] D. V. Lindley. A statistical paradox. *Biometrika*, 44(1/2):187–192, 1957.

[69] M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. *Genome biology*, 15(12):550, 2014.

[70] J. C. Marioni, C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad. RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research*, 18(9):1509–1517, 2008.

[71] D. C. Martins, U. M. Braga-Neto, R. F. Hashimoto, M. L. Bittner, and E. R. Dougherty. Intrinsically multivariate predictive genes. *IEEE Journal of Selected Topics in Signal Processing*, 2(3):424–439, 2008.

[72] P. Meyer, K. Kontos, F. Lafitte, and G. Bontempi. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP Journal on Bioinformatics and Systems Biology*, 2(7):1–8, 2007.

[73] A. W. Norris and A. A. Spector. Very long chain n-3 and n-6 polyunsaturated fatty acids bind strongly to liver fatty acid-binding protein. *Journal of Lipid Research*, 43(4):646–653, 2002.

[74] D. Pandey, G. Sikka, Y. Bergman, J. H. Kim, S. Ryoo, L. Romer, and D. Berkowitz. Transcriptional regulation of endothelial arginase by histone deacetylase. *Arteriosclerosis, Thrombosis, and Vascular Biology*, 15(3):102–114, 2014.

[75] J. Park, F. Greten, A. Wong, R. Westrick, J. Arthur, K. Otsu, A. Hoffmann, M. Montminy, and M. Karin. Signaling pathways and genes that inhibit pathogen-induced macrophage apoptosis–CREB and NF-κB as key regulators. *Immunity*, 23(3):319–329, 2005.

[76] M. Prendes, Y. Zheng, and A. Beg. Regulation of developing B cell survival by RelA-containing NF-κ B complexes. *The Journal of Immunology*, 171(8):3963,

2003.

[77] A. Rau, G. Celeux, M.-L. Martin-Magniette, and C. Maugis-Rabusseau. Clustering high-throughput sequencing data with Poisson mixture models. *Inria*, 77(86):36–48, 2011.

[78] L. Ries, D. Melbert, M. Krapcho, D. Stinchcomb, N. Howlader, M. Horner, A. Mariotto, B. Miller, E. Feuer, S. Altekruse, et al. SEER cancer statistics review, 1975-2005. *Bethesda, MD: National Cancer Institute*, pages 1975–2005, 2008.

[79] M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

[80] Y. Satoh, K. Mori, K. Kitano, J. Kitayama, H. Yokota, H. Sasaki, H. Uozaki, M. Fukayama, Y. Seto, H. Nagawa, et al. Analysis for the combination expression of CK20, FABP1 and MUC2 is sensitive for the prediction of peritoneal recurrence in gastric cancer. *Japanese Journal of Clinical Oncology*, 5(9):165–179, 2011.

[81] M. Schmidt-Supprian, W. Bloch, G. Courtois, K. Addicks, A. Israel, K. Rajewsky, and M. Pasparakis. NEMO/IKKγ-deficient mice model incontinentia pigmenti. *Molecular Cell*, 5(6):981–992, 2000.

[82] G. Shafer. Lindley's paradox. *Journal of the American Statistical Association*, 77(378):325–334, 1982.

[83] Y. Si, P. Liu, P. Li, and T. P. Brutnell. Model-based clustering for RNA-seq data. *Bioinformatics*, 30(2):197–205, 2014.

[84] S. Simmons, C. Fan, and R. Ramabhadran. Cellular stress response pathway system as a sentinel ensemble in toxicological screening. *Toxicological Sciences*, 111(2):202–212, 2009.

[85] R. Stierum, M. Gaspari, Y. Dommels, T. Ouatas, H. Pluk, S. Jespersen, J. Vogels, K. Verhoeckx, J. Groten, and B. Ommen. Proteome analysis reveals novel proteins associated with proliferation and differentiation of the colorectal cancer cell line Caco-2. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, 1650(1):73–91, 2003.

[86] C. Trapnell, A. Roberts, L. Goff, G. Pertea, D. Kim, D. R. Kelley, H. Pimentel, S. L. Salzberg, J. L. Rinn, and L. Pachter. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols*, 7(3):562–578, 2012.

[87] A. Valentini, M. Biancolella, F. Amati, P. Gravina, R. Miano, G. Chillemi, A. Farcomeni, S. Bueno, G. Vespasiani, A. Desideri, et al. Valproic acid induces neuroendocrine differentiation and UGT2B7 up-regulation in human prostate carcinoma cell line. *Drug Metabolism and Disposition*, 35(6):968–972, 2007.

[88] S. Werner, D. Barken, and A. Hoffmann. Stimulus specificity of gene expression programs determined by temporal control of IKK activity. *Science*, 309(5742):1857, 2005.

[89] D. M. Witten et al. Classification and clustering of sequencing data using a Poisson model. *The Annals of Applied Statistics*, 5(4):2493–2518, 2011.

[90] S. Zhang, A. Kovalenko, G. Cantarella, and D. Wallach. Recruitment of the IKK Signalosome to the p55 TNF Receptor:: RIP and A20 Bind to NEMO upon Receptor Stimulation. *Immunity*, 12(3):301–311, 2000.

APPENDIX A

ADDITIONAL ALGORITHMS FOR OPTIMAL BAYESIAN CLASSIFICATION

Algorithm 2: Calibrate Priors - This is the procedure to use discarded features to create calibrated prior distributions for use in a classification problem. Reproduced with permission from [57].

[1] μ-list $\leftarrow$ [] $\Sigma_{i,i}$-list $\leftarrow$ [] $\Sigma_{i,j}$-list $\leftarrow$ [] **for** $i = 1 : s$ **do**

s pairs sampled Randomly select a pair of features f_1, f_2 $dsub \leftarrow$ data for this pair Initialize uniform priors MCMCSamples $\leftarrow N$ MCMC Samples using $dsub$ $\mathrm{E}[\mu_1], \mathrm{E}[\mu_2] \leftarrow$ Sample μ means from MCMCSamples Append $\mathrm{E}[\mu_1], \mathrm{E}[\mu_2]$ to μ-list Append $\mathrm{E}[\Sigma_{1,1}], \mathrm{E}[\Sigma_{2,2}]$ to $\Sigma_{i,i}$-list Append $\mathrm{E}[\Sigma_{0,1}]$ to $\Sigma_{i,j}$-list sigdiagmean $\leftarrow$ mean($\Sigma_{i,i}$-list) sigoffmean $\leftarrow$ mean($\Sigma_{i,j}$-list) sigdiagvar $\leftarrow \frac{1}{s-1} \sum_{i=1}^{s} (\text{sigdiagmean} - \Sigma_{i,i}\text{-list}[i])^2$ $\widehat{m} \leftarrow$ mean(μ-list) $\widehat{\nu} \leftarrow$ var(μ-list) $\widehat{\sigma^2} \leftarrow 2\times$ sigdiagmean $\times (\frac{\text{sigdiagmean}^2}{\text{sigdiagvar}} + 1)$ $\widehat{\rho} \leftarrow \frac{\text{sigoffmean}}{\text{sigdiagmean}}$ $\widehat{\kappa} \leftarrow \frac{2\times\text{sigdiagmean}^2}{\text{siagdiagvar}} + D + 3$

Algorithm 3: Generate IC Synthetic Data - To examine the effects of independent covariance matrices, we used the following IC method to first draw random covariance matrices for each class, and then to sample data. Reproduced with permission from [57].

[1] $N, \mathrm{d}_{low}, \mathrm{d}_{high}$ N: Number of samples desired $D \leftarrow 20$ $\kappa \leftarrow D + 2$ **for** *each class* **do**

$\mu \leftarrow \text{Normal}(0, 0.2) \times \text{ones}(D)$ $\sigma \leftarrow \text{Normal}(0, 0.2)$
$\Sigma \leftarrow \text{Inverse-Wishart}(I_D(\kappa - D - 1) * \sigma, D + 2)$ **if** *Low correlation features* **then**

off-diagonal$(\Sigma) \leftarrow 0$ data $\leftarrow$ empty $N \times D$ matrix lams $\leftarrow$ Draw N vectors from Normal(μ, Σ) **for** $i = 1 : N$ **do**

$j = 1 : D$ data$[i, j] \leftarrow \text{Poisson-draw}(\text{Uniform}(\mathrm{d}_{low}, \mathrm{d}_{high}) \times \exp(\text{lams}[i, j]))$

Algorithm 4: Synthetic Validation Procedure - The steps used to generate the sets of points for each N_{trn} (the number of training samples in each class along the x-axis in Figure 3.3. Reproduced with permission from [57].

[1] **for** $i = 1 : N$ **do**

N: Number of averages desired $\mu_0 \leftarrow \text{Normal-draw}(0.0, 0.2)$
$\mu_1 \leftarrow \text{Normal-draw}(0.0, 0.2)$ $\sigma_0 \leftarrow \text{InverseGamma-draw}(3.0, 1.0)$
$\sigma_1 \leftarrow \text{InverseGamma-draw}(3.0, 1.0)$ train-data-0 $\leftarrow \text{genData}(\mu_0, \sigma_1, N_{trn}, \rho)$
train-data-1 $\leftarrow \text{genData}(\mu_1, \sigma_1, N_{trn}, \rho)$ test-data-0 $\leftarrow \text{genData}(\mu_0, \sigma_1, N_{test}, \rho)$
test-data-1 $\leftarrow \text{genData}(\mu_0, \sigma_1, N_{test}, \rho)$ used-features $\leftarrow$ Randomly select 4 features Using Training data: hyperparameters $\leftarrow$ MCMC using Algorithm 1 and used-features[c] Train (Run) Calibrated MCMC with hyperparameters Train (Run) MCMC with weakly informative priors Train SVM Train LDA Train 3NN Train Normal OBC Using testing data: Evaluate Calibrated MCMC Evaluate MCMC with weakly informative priors Evaluate SVM Evaluate LDA Evaluate 3NN Evaluate Normal OBC

Algorithm 5: Real Data Validation Procedure - The procedure used to generate each set of points along the x-axis of Figure 3.4 given a desired number of training samples $N_{trntotal}$ over N averages with an *a priori* known value of c. Reproduced with permission from [57].

[1] **for** $i = 1 : N$ **do**

N: Number of averages desired train-data$_0$ $\leftarrow$ draw round$(c * N_{trntotal})$ samples from data$_0$ train-data$_1$ $\leftarrow$ draw round$((1 - c) * N_{trntotal})$ samples from data$_1$ test-data$_0$ $\leftarrow$ data$_0$ - train-data$_0$ test-data$_1$ $\leftarrow$ data$_1$ - train-data$_1$ used-features $\leftarrow$ Randomly select 4 features Using Training data: hyperparameters $\leftarrow$ Algorithm 1 MCMC using used-features^c Train (Run) Calibrated MCMC using hyperparameters Train (Run) MCMC with weakly informative priors Train SVM Train LDA Train 3NN Train Normal OBC Using testing data: Evaluate Calibrated MCMC Evaluate MCMC with weakly informative priors Evaluate SVM Evaluate LDA Evaluate 3NN Evaluate Normal OBC

APPENDIX B

ADDITIONAL FIGURES FOR OPTIMAL BAYESIAN CLASSIFICATION

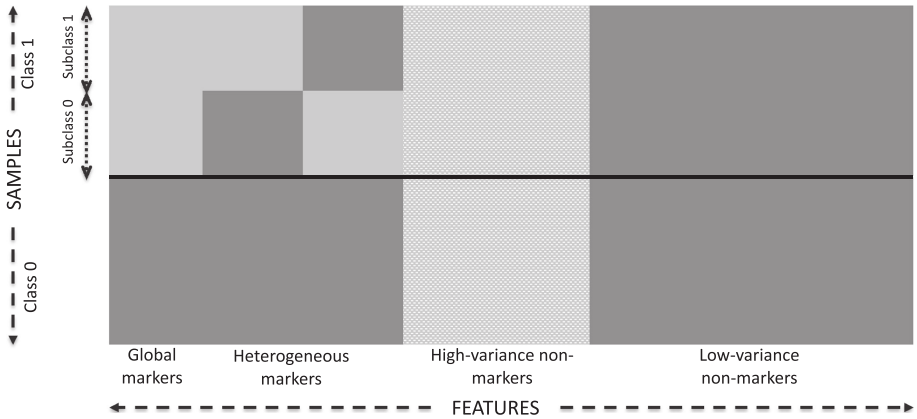

Figure B.1: The multivariate normal distribution used to generate samples for the IC synthetic data case. The block structure indicates the several different types of features that are generated. Used with permission from Ghaffari *et al.*, 2013 [36].

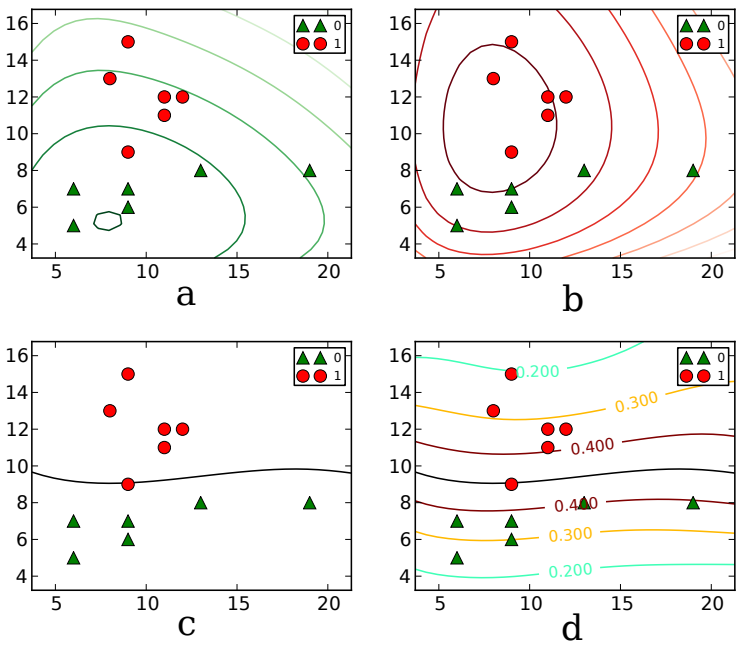

Figure B.2: A simple two class, two gene, synthetic example demonstrates the use of the MP OBC. Six training samples from each class (circles and triangles) are shown in all four panels and used to train the MP model. After MCMC computation, the resulting effective class conditional density contour is shown for the triangles in panel a and the circles in panel b. Panel c then shows the resulting MP OBC decision boundary resulting from these effective class conditional densities and panel d shows the contours of the optimal Bayes conditional error estimate plotted next to the classifier decision boundary. Reproduced with permission from [57].

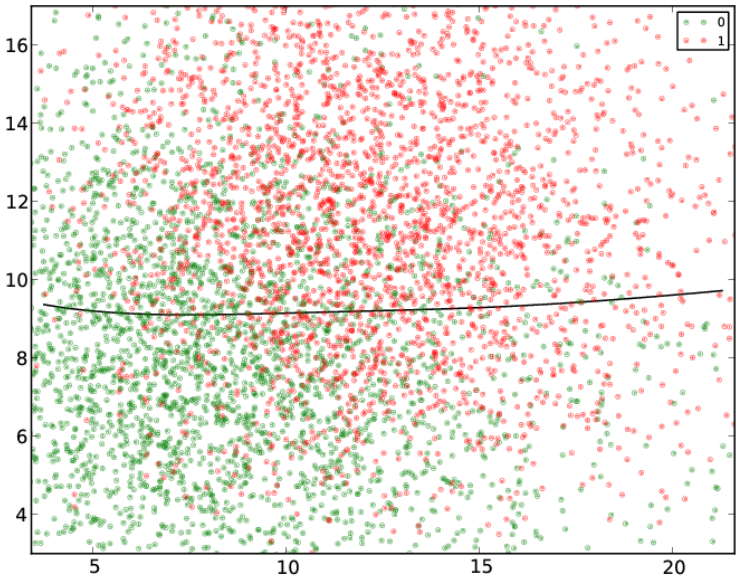

Figure B.3: Using the same classifier, we can now evaluate the performance of the classifier using 3000 testing samples from each class. When evaluated and averaged, this particular example results in a classification error of 0.29. Reproduced with permission from [57].

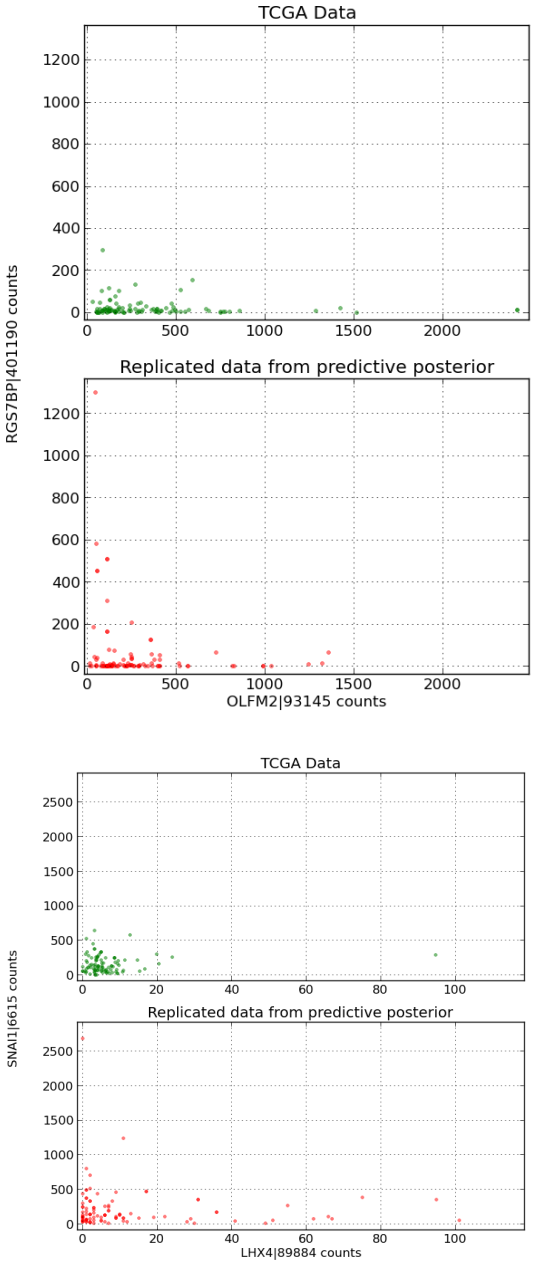

Figure B.4: Two examples of 100 samples from adenocarcinoma TCGA tumor samples and the posterior predictive x^{rep} simulation from the MP model. Reproduced with permission from [57].

APPENDIX C

ADDITIONAL BAYESIAN POSTERIOR P-VALUES

Table C.1: Posterior predictive model diagnostic – 5th quantile. Reproduced with permission from [57].

Gene ID	Mean expression (counts)	$T(S_n)$	95th int. $T(x^{rep})$	P-value
MRGPRX1|259249	0.0	0.00	[0.00, 0.00]	0.50
C17orf105|284067	0.0	0.00	[0.00, 0.00]	0.50
HBBP1|3044	0.1	0.00	[0.00, 0.00]	0.50
SNORA18|677805	0.2	0.00	[0.00, 0.00]	0.50
SCN10A|6336	0.3	0.00	[0.00, 0.00]	0.50
CDCP2|200008	0.7	0.00	[0.00, 0.00]	0.50
FGF17|8822	3.3	0.00	[0.00, 0.00]	0.50
NTN5|126147	5.7	1.32	[0.00, 0.00]	0.00
NCRNA00185|55410	11.5	0.00	[0.00, 0.00]	0.50
CCR8|1237	16.6	1.46	[0.00, 1.90]	0.06
CCDC33|80125	23.1	0.00	[0.00, 0.00]	0.50
PPAPDC3|84814	23.5	6.00	[2.95, 8.95]	0.32
PCDHGB2|56103	68.1	5.19	[1.95, 10.80]	0.47
FAM81A|145773	81.1	13.38	[5.00, 17.90]	0.23
ZNF383|163087	81.2	54.44	[24.00, 48.45]	0.01
ANKRD1|27063	87.1	1.23	[0.00, 3.95]	0.52
UGT2B4|7363	93.4	0.00	[0.00, 0.00]	0.50
IL12RB1|3594	98.9	15.33	[13.85, 36.40]	0.88
ZNF628|89887	160.0	75.54	[41.95, 83.80]	0.16
FBF1|85302	184.2	52.03	[29.60, 75.50]	0.41
ZNF615|284370	209.2	70.60	[36.95, 81.40]	0.16
RHBDD1|84236	299.9	127.69	[82.90, 170.90]	0.46
NICN1|84276	330.1	173.29	[87.95, 187.40]	0.12
COQ6|51004	369.7	193.25	[106.95, 194.20]	0.05
CHAF1A|10036	387.3	150.70	[81.55, 190.65]	0.30
DTD1|92675	534.5	273.70	[141.50, 282.20]	0.07
EARS2|124454	663.7	380.07	[193.10, 356.85]	0.03
KIAA1737|85457	668.3	365.58	[214.25, 393.95]	0.12
LRRC8D|55144	690.2	445.36	[214.45, 405.95]	0.02
SKIL|6498	691.0	336.66	[199.70, 368.85]	0.15
WDR36|134430	761.6	531.67	[258.20, 479.55]	0.02
ZNF259|8882	831.0	455.39	[221.15, 425.20]	0.03
CHSY1|22856	1029.7	474.64	[269.05, 548.10]	0.15
DHX8|1659	1192.9	695.25	[354.10, 728.70]	0.08
AGTRAP|57085	1254.0	539.55	[283.75, 622.85]	0.20
VPS26B|112936	1337.3	643.79	[350.35, 716.25]	0.17
MCM4|4173	1543.3	343.29	[202.80, 528.50]	0.51
SLC2A3|6515	1559.7	338.07	[178.00, 474.20]	0.38
VPS39|23339	1594.0	908.35	[460.90, 964.45]	0.07
FOXA1|3169	1800.3	396.41	[217.10, 584.20]	0.40

Table C.2: Posterior predictive model diagnostic – Median. Reproduced with permission from [57].

Gene ID	Mean expression (counts)	$T(S_n)$	95th int. $T(x^{rep})$	P-value
MRGPRX1|259249	0.0	0.00	[0.00, 0.00]	0.50
C17orf105|284067	0.0	0.00	[0.00, 0.00]	0.50
HBBP1|3044	0.1	0.00	[0.00, 0.00]	0.50
SNORA18|677805	0.2	0.00	[0.00, 0.00]	0.50
SCN10A|6336	0.3	0.00	[0.00, 0.00]	0.50
CDCP2|200008	0.7	0.00	[0.00, 0.00]	0.50
FGF17|8822	3.3	1.01	[0.00, 1.00]	0.03
NTN5|126147	5.7	4.35	[2.00, 7.50]	0.41
NCRNA00185|55410	11.5	0.00	[0.00, 0.00]	0.50
CCR8|1237	16.6	12.18	[4.00, 17.50]	0.22
CCDC33|80125	23.1	6.01	[1.00, 10.00]	0.17
PPAPDC3|84814	23.5	19.68	[13.50, 24.00]	0.36
PCDHGB2|56103	68.1	33.59	[18.50, 46.00]	0.34
FAM81A|145773	81.1	45.82	[31.00, 71.50]	0.59
ZNF383|163087	81.2	85.50	[68.00, 104.50]	0.45
ANKRD1|27063	87.1	20.69	[9.50, 34.00]	0.36
UGT2B4|7363	93.4	1.77	[0.00, 4.00]	0.30
IL12RB1|3594	98.9	80.38	[56.50, 110.00]	0.45
ZNF628|89887	160.0	158.95	[121.00, 194.50]	0.39
FBF1|85302	184.2	162.49	[111.00, 211.00]	0.28
ZNF615|284370	209.2	184.09	[126.00, 218.00]	0.25
RHBDD1|84236	299.9	275.19	[214.00, 337.50]	0.48
NICN1|84276	330.1	320.44	[264.00, 436.00]	0.60
COQ6|51004	369.7	335.85	[265.00, 405.00]	0.46
CHAF1A|10036	387.3	302.09	[252.50, 452.50]	0.75
DTD1|92675	534.5	523.20	[385.00, 626.00]	0.30
EARS2|124454	663.7	603.62	[485.00, 732.50]	0.41
KIAA1737|85457	668.3	676.88	[529.00, 806.50]	0.39
LRRC8D|55144	690.2	645.04	[550.50, 835.00]	0.67
SKIL|6498	691.0	594.36	[479.50, 761.00]	0.56
WDR36|134430	761.6	752.27	[631.50, 909.00]	0.55
ZNF259|8882	831.0	658.89	[551.00, 903.00]	0.71
CHSY1|22856	1029.7	872.60	[721.00, 1146.50]	0.59
DHX8|1659	1192.9	1150.34	[957.00, 1552.00]	0.67
AGTRAP|57085	1254.0	1069.75	[839.00, 1375.50]	0.55
VPS26B|112936	1337.3	1189.98	[912.00, 1415.00]	0.40
MCM4|4173	1543.3	1094.04	[795.00, 1417.00]	0.41
SLC2A3|6515	1559.7	1053.06	[719.50, 1420.50]	0.45
VPS39|23339	1594.0	1651.32	[1274.00, 2078.50]	0.43
FOXA1|3169	1800.3	1168.83	[887.50, 1642.00]	0.62

Table C.3: Posterior predictive model diagnostic – 95th quantile. Reproduced with permission from [57].

Gene ID	Mean expression (counts)	$T(S_n)$	95th int. $T(x^{rep})$	P-value
MRGPRX1|259249	0.0	0.00	[0.00, 0.00]	0.50
C17orf105|284067	0.0	1.01	[0.00, 0.05]	0.00
HBBP1|3044	0.1	0.50	[0.00, 1.00]	0.07
SNORA18|677805	0.2	0.25	[0.00, 0.00]	0.00
SCN10A|6336	0.3	1.49	[0.05, 7.45]	0.38
CDCP2|200008	0.7	2.12	[0.05, 25.10]	0.56
FGF17|8822	3.3	16.60	[6.35, 279.40]	0.78
NTN5|126147	5.7	14.10	[16.40, 109.85]	0.97
NCRNA00185|55410	11.5	38.88	[1.05, 146.00]	0.20
CCR8|1237	16.6	38.64	[42.30, 308.40]	0.97
CCDC33|80125	23.1	107.94	[59.25, 2444.60]	0.85
PPAPDC3|84814	23.5	48.09	[35.05, 87.75]	0.66
PCDHGB2|56103	68.1	179.96	[89.40, 276.50]	0.36
FAM81A|145773	81.1	200.17	[127.60, 378.45]	0.54
ZNF383|163087	81.2	144.50	[144.55, 263.35]	0.95
ANKRD1|27063	87.1	225.79	[82.90, 400.90]	0.31
UGT2B4|7363	93.4	54.66	[19.50, 1361.35]	0.80
IL12RB1|3594	98.9	230.41	[155.80, 432.00]	0.65
ZNF628|89887	160.0	299.84	[269.00, 554.20]	0.86
FBF1|85302	184.2	374.78	[297.80, 748.75]	0.72
ZNF615|284370	209.2	368.14	[316.55, 769.65]	0.87
RHBDD1|84236	299.9	427.40	[455.05, 820.10]	0.97
NICN1|84276	330.1	737.03	[611.90, 1253.15]	0.74
COQ6|51004	369.7	568.48	[542.15, 1066.55]	0.90
CHAF1A|10036	387.3	834.67	[617.75, 1323.90]	0.54
DTD1|92675	534.5	989.05	[832.60, 1590.50]	0.75
EARS2|124454	663.7	1005.24	[970.90, 1902.00]	0.91
KIAA1737|85457	668.3	887.21	[1075.65, 1961.85]	1.00
LRRC8D|55144	690.2	1086.87	[1088.05, 2025.85]	0.95
SKIL|6498	691.0	1140.47	[1019.95, 2039.50]	0.80
WDR36|134430	761.6	1220.71	[1231.45, 2132.60]	0.95
ZNF259|8882	831.0	1169.19	[1186.25, 2164.80]	0.95
CHSY1|22856	1029.7	1705.17	[1557.95, 2971.50]	0.86
DHX8|1659	1192.9	1953.12	[2051.05, 4246.15]	0.97
AGTRAP|57085	1254.0	2240.63	[1841.20, 4096.95]	0.76
VPS26B|112936	1337.3	1847.39	[1949.45, 3436.10]	0.97
MCM4|4173	1543.3	3410.08	[2205.70, 5370.80]	0.41
SLC2A3|6515	1559.7	3378.11	[2183.05, 5712.85]	0.51
VPS39|23339	1594.0	2514.10	[2719.90, 5625.80]	0.98
FOXA1|3169	1800.3	3302.97	[2553.55, 6376.40]	0.73

Table C.4: Posterior predictive model diagnostic – IQR. Reproduced with permission from [57].

Gene ID	Mean expression (counts)	$T(S_n)$	95th int. $T(x^{rep})$	P-value
MRGPRX1|259249	0.0	0.00	[0.00, 0.00]	0.50
C17orf105|284067	0.0	0.00	[0.00, 0.00]	0.50
HBBP1|3044	0.1	0.00	[0.00, 0.00]	0.50
SNORA18|677805	0.2	0.00	[0.00, 0.00]	0.50
SCN10A|6336	0.3	0.50	[0.00, 0.00]	0.03
CDCP2|200008	0.7	0.98	[0.00, 1.00]	0.07
FGF17|8822	3.3	2.59	[1.00, 9.00]	0.52
NTN5|126147	5.7	6.93	[4.50, 21.75]	0.75
NCRNA00185|55410	11.5	5.45	[0.00, 1.50]	0.00
CCR8|1237	16.6	17.69	[11.00, 58.00]	0.66
CCDC33|80125	23.1	17.19	[6.25, 93.25]	0.60
PPAPDC3|84814	23.5	13.91	[11.00, 25.50]	0.79
PCDHGB2|56103	68.1	43.70	[28.50, 82.25]	0.60
FAM81A|145773	81.1	63.53	[38.75, 109.75]	0.51
ZNF383|163087	81.2	34.43	[40.25, 86.75]	1.00
ANKRD1|27063	87.1	74.52	[21.25, 83.75]	0.11
UGT2B4|7363	93.4	6.98	[2.00, 43.50]	0.64
IL12RB1|3594	98.9	84.17	[47.50, 130.00]	0.42
ZNF628|89887	160.0	100.43	[78.50, 171.50]	0.73
FBF1|85302	184.2	154.54	[95.00, 226.75]	0.36
ZNF615|284370	209.2	117.16	[96.00, 228.50]	0.82
RHBDD1|84236	299.9	126.20	[128.75, 275.50]	0.96
NICN1|84276	330.1	246.20	[179.75, 411.50]	0.61
COQ6|51004	369.7	153.72	[153.50, 361.00]	0.95
CHAF1A|10036	387.3	234.48	[172.75, 422.25]	0.72
DTD1|92675	534.5	297.30	[236.75, 533.75]	0.77
EARS2|124454	663.7	237.65	[262.25, 578.50]	0.98
KIAA1737|85457	668.3	220.32	[281.50, 651.75]	0.99
LRRC8D|55144	690.2	359.07	[285.00, 655.50]	0.83
SKIL|6498	691.0	247.07	[290.75, 615.75]	1.00
WDR36|134430	761.6	216.40	[331.25, 737.50]	1.00
ZNF259|8882	831.0	322.44	[314.50, 715.00]	0.94
CHSY1|22856	1029.7	441.38	[442.25, 1005.00]	0.95
DHX8|1659	1192.9	582.12	[584.75, 1240.00]	0.95
AGTRAP|57085	1254.0	622.58	[532.25, 1264.50]	0.86
VPS26B|112936	1337.3	529.40	[536.00, 1111.25]	0.96
MCM4|4173	1543.3	835.39	[631.25, 1588.50]	0.74
SLC2A3|6515	1559.7	845.02	[611.00, 1788.75]	0.79
VPS39|23339	1594.0	586.40	[764.75, 1831.00]	1.00
FOXA1|3169	1800.3	1275.06	[765.00, 2040.75]	0.50

Table C.5: Posterior predictive model diagnostic – Variance. Reproduced with permission from [57].

Gene ID	Mean expression (counts)	$T(S_n)$	95th int. $T(x^{rep})$	P-value
MRGPRX1|259249	0.0	0.00	[0.00, 0.00]	0.04
C17orf105|284067	0.0	0.11	[0.00, 0.12]	0.07
HBBP1|3044	0.1	0.08	[0.00, 0.25]	0.13
SNORA18|677805	0.2	0.01	[0.00, 0.05]	0.25
SCN10A|6336	0.3	0.36	[0.05, 1242.64]	0.69
CDCP2|200008	0.7	0.74	[0.16, 5117.85]	0.80
FGF17|8822	3.3	26.34	[32.90, 1109445.84]	0.97
NTN5|126147	5.7	42.93	[51.13, 8446.14]	0.96
NCRNA00185|55410	11.5	303.63	[1.35, 1340191.71]	0.53
CCR8|1237	16.6	157.20	[285.39, 81521.57]	0.98
CCDC33|80125	23.1	14201.11	[2292.16, 31223489.92]	0.79
PPAPDC3|84814	23.5	181.37	[112.92, 1048.31]	0.80
PCDHGB2|56103	68.1	4048.26	[1228.20, 20426.96]	0.52
FAM81A|145773	81.1	5661.96	[1970.14, 26450.20]	0.53
ZNF383|163087	81.2	1206.74	[1301.43, 6287.70]	0.96
ANKRD1|27063	87.1	9884.12	[1356.27, 51853.87]	0.36
UGT2B4|7363	93.4	5664.11	[291.72, 8368761.19]	0.66
IL12RB1|3594	98.9	4745.05	[2290.87, 27420.28]	0.79
ZNF628|89887	160.0	6534.92	[5024.43, 31689.22]	0.87
FBF1|85302	184.2	12711.43	[7286.06, 71120.08]	0.77
ZNF615|284370	209.2	11122.81	[9659.19, 69434.34]	0.88
RHBDD1|84236	299.9	8546.28	[13399.90, 63578.81]	0.99
NICN1|84276	330.1	32463.87	[26378.50, 154304.78]	0.87
COQ6|51004	369.7	15005.15	[19147.05, 94982.52]	0.98
CHAF1A|10036	387.3	47233.01	[28450.33, 196129.54]	0.74
DTD1|92675	534.5	59301.87	[45681.73, 251258.48]	0.90
EARS2|124454	663.7	52839.46	[58542.64, 301221.33]	0.98
KIAA1737|85457	668.3	48505.67	[75259.02, 340071.74]	0.99
LRRC8D|55144	690.2	48918.16	[78779.94, 378872.68]	1.00
SKIL|6498	691.0	56797.55	[69545.32, 366715.84]	0.97
WDR36|134430	761.6	49947.90	[88525.22, 422834.77]	1.00
ZNF259|8882	831.0	88575.80	[82583.27, 472063.68]	0.94
CHSY1|22856	1029.7	201986.35	[168564.65, 873474.11]	0.90
DHX8|1659	1192.9	348984.33	[285590.39, 1578955.67]	0.87
AGTRAP|57085	1254.0	424287.54	[248618.71, 1869240.10]	0.71
VPS26B|112936	1337.3	158070.60	[240558.63, 1169889.12]	1.00
MCM4|4173	1543.3	901681.93	[386142.75, 3614045.91]	0.63
SLC2A3|6515	1559.7	1037093.04	[421036.96, 4698669.95]	0.61
VPS39|23339	1594.0	328769.35	[481155.30, 2674633.59]	1.00
FOXA1|3169	1800.3	1688527.66	[680898.40, 7470155.18]	0.54