Bayesian analysis of oncogenic pathway activation

Aaron Bryant

a thesis submitted for the degree of

Master of Science
at the University of Otago, Dunedin,
New Zealand.

May 2011
Abstract

Through the use of microarray technology researchers are now able to simultaneously measure the expression levels of tens of thousands of genes. Among other things, this allows the construction of profiles of activation of various pathways within tumour samples. Using a metacohort of 2116 breast cancer patients, this thesis explores various Bayesian factor regression methods to estimate the probability of pathway activation using gene expression data. The relationship between these probabilities and various histological methods is also examined.

The methods of estimation can be broken into two main categories, univariate and multivariate. The univariate method has been explored previously, and this thesis aims to replicate the work of Bild et al. (2006), as well as explore other univariate models. In this thesis the univariate models were implemented mainly in R using the rbprobit package, although work was also undertaken in implementing these models in Python using PyMC. The multivariate approach has not been taken before, and is of interest due to the fact that biological pathways do not act in isolation. The multivariate techniques look to find correlation between the various pathways, to give more accurate estimates of the probability of pathway activation.

Once the probabilities of pathway activation were estimated using the various univariate and multivariate methods, the best model from the two categories was selected. The estimates from these two models were then combined with various histological information (i.e. estrogen receptor status, progesterone receptor status, and lymph node status), to see whether or not there is any advantage in using a multivariate approach over a univariate one, and to determine if pathway activation status is associated with the histological information.
Acknowledgements

The initial impetus to embark on this thesis, was provided by my good friend Zoe van Havre who inspired and encouraged me to pursue a Masters degree, so I would like to thank her for the push in the right direction.

I would like thank my family for their constant support and encouragement in everything I pursue in life, I would especially like to thank my mother for acting as my proof-reader in my academic pursuits.

A special note must go to Art Van der Lay his architectural advice provided me with plenty of comic relief while completing this thesis. I would also like to thank the other Maths and Stats postgraduates with whom I spent many an hour having interesting conversations and debates. A particular mention need to go to Ross, Tim, Daniel, Ella, and Harish with who, I shared a glorious summer playing twilight cricket.

This thesis would not have been possible without the funding I received from the Breast cancer research trust through a grant to Associate Professor Cristin Print at the University of Auckland, and also a Beverley Postgraduate Scholarship that was jointly funded by Biochemistry and Mathematics and Statistics departments at the University of Otago.

Thank you to Chris Fonnesbeck, one my original co-supervisors for his help on all things Python. Lastly but by no means least I would like to thank my supervisor Dr. Mik Black for all the help and support he has given me, including but not limited to, helping me find funding, giving me an extra set of eyes to find bugs in code, to just sharing a good laugh in his office.
**Contents**

1 **Introduction**  
   1.1 Microarray technology  
   1.2 Biology of breast cancer  
   1.3 Biomedical applications of microarrays  
   1.4 Microarrays applied to breast cancer  
   1.5 Summary  

2 **Bayesian Factor Regression Modelling**  
   2.1 Microarray analysis  
   2.2 Dimensionality reduction  
   2.3 Probit regression model  
   2.4 Microarray analysis  
   2.5 Generating the metagenes  
   2.6 The three models  
   2.7 Gibbs sampling/Data imputation for the univariate  
   2.8 Simulation  
   2.9 Python Implementation  
   2.10 Introduction to python  
   2.11 Creating metagenes  
   2.12 Python implementation of univariate model  
   2.13 Gradient graphs  
   2.14 Summary  

3 **Multivariate Models**  
   3.1 Generating the multivariate metagenes  
   3.2 Comparison of the two multivariate metagenes  
   3.3 Multivariate probit regression model  
   3.4 Allenby multivariate probit  
   3.5 Multivariate distribution  
   3.6 Alternate multivariate model  
   3.7 Testing the correlation estimator  
   3.8 Multivariate data imputation  
   3.9 Multivariate simulation  
   3.10 Summary  

  

# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Survival rates for each of the Stages of Breast cancer. Based on data collected by the National Cancer Data Base from 2001 and 2002.</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Number of arrays available for each oncogenic pathway</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>Cohorts used in Study</td>
<td>21</td>
</tr>
<tr>
<td>2.3</td>
<td>Number of Arrays available for Each Oncogenic Pathway</td>
<td>34</td>
</tr>
<tr>
<td>4.1</td>
<td>Univariate models</td>
<td>61</td>
</tr>
<tr>
<td>4.2</td>
<td>Spearman’s rank correlation coefficients between probability of activation</td>
<td>68</td>
</tr>
<tr>
<td>4.3</td>
<td>Correlation between multivariate probability of activation</td>
<td>81</td>
</tr>
<tr>
<td>4.4</td>
<td>Correlation between multivariate probability of activation</td>
<td>83</td>
</tr>
<tr>
<td>5.1</td>
<td>Coefficients for the univariate probit model with lymph node status as the dependant variable, and the respective pathways as the predictor variable.</td>
<td>88</td>
</tr>
<tr>
<td>5.2</td>
<td>Coefficients for the univariate probit model with estrogen receptor status as the dependant variable, and the respective pathways as the predictor variable.</td>
<td>90</td>
</tr>
<tr>
<td>5.3</td>
<td>Coefficients for the univariate probit model with progesterone receptor status as the dependant variable, and the respective pathways as the predictor variable.</td>
<td>92</td>
</tr>
<tr>
<td>5.4</td>
<td>Table comparing the mean and 95% credible intervals of the pathway coefficients, between the glm and the rbprobit models for Estrogen receptor status.</td>
<td>95</td>
</tr>
<tr>
<td>5.5</td>
<td>Table comparing the mean and 95% credible intervals of the pathway coefficients, between the glm and the rbprobit models for disease free survival.</td>
<td>97</td>
</tr>
</tbody>
</table>
# List of Figures

1.1 Basic Diagram of the “Central Dogma of Molecular Biology” .................................. 2

2.1 Each individual graph shows a scatterplot of the normal and pathway cell lines for each of the 5 pathways with the primary metagene as the x axis and the secondary metagene as the y axis. The metagenes used were generated using the “univariate method”. The cell line points are colour coded with the normal being green, BCat - Pink, E2F3 - Blue, Myc - Red, Ras - Black, Src - Cyan .................................................. 23

2.2 Plots comparing the estimates probability of activation using the univariate probit model, when the response is totally known (x axis) and when there are missing values (y axis). Comparing how standardising effects these estimates, plot (a) shows the raw data, plot (b) shows the standardised data. ................................................................. 27

2.3 Plots comparing the estimates probability of activation using the univariate probit model, when the response is totally known (x axis) and when there are missing values (y axis). Comparing how a larger quantity of missing values effects the estimates. .................................................. 28

2.3 Scatterplots of normal, pathway, and tumour cell lines. These graphs have been colour coded on a sliding spectrum from the lowest values (blue) to the highest values (red) of probability of activation. ................. 37

3.1 Each individual graph shows a scatterplot of the normal and pathway cell lines for each of the five pathways with the primary metagene as the x axis and the secondary metagene as the y axis. The metagenes used were generated using the “multivariate method one”. The cell line points are colour coded with the normal being green, BCat - Pink, E2F3 - Blue, Myc - Red, Ras - Black, Src - Cyan .................. 42

3.2 Each individual graph shows a scatterplot of the normal and pathway cell lines for each of the five pathways with the primary metagene as the x axis and the secondary metagene as the y axis. The metagenes used were generated using the “multivariate method two”. The cell line points are colour coded with the normal being green, BCat - Pink, E2F3 - Blue, Myc - Red, Ras - Black, Src - Cyan .......................... 43

3.3 Contour plot of a bivariate Normal with \( \mu = 0 \) and \( \rho = 0.5 \) .................. 47

3.4 Contour plot of a bivariate Normal with \( \mu = 0 \) and \( \rho = 0 \) ............... 48
3.5 Graphs plotting the three different pathways primary metagenes (from multivariate method one) against one another. The points are color coded with the normal - Green, E2F3 - Blue, Myc - Red, Ras - Black, tumours - Grey. The main interest in these graphs is the correlation of the tumours between metagenes for the different pathways.

3.6 A plot of the distributions of the predicted correlations from 1000 samples. There were nine different correlations ranging from -1 to 1. A blue line indicates the posterior mean of these predictions, and a red line shows what the actual observed correlation was.

3.7 A plot of the distribution of the predicted correlations between the metagenes generated by “multivariate method 1”. The first column is between E2F3 and Myc, the second column is E2F3 and Ras, and the third between Myc and Ras. The top row shows the distributions of the predicted correlations before the metagenes were standardised, the bottom row shows the predicted correlations after the metagenes have been standardised to have mean of 0 and standard deviation of 1.

3.8 Plots comparing the estimates probability of activation using the multivariate probit model, when the response is totally known (x axis) and when there are missing values (y axis). (a) and (b) contain non-standardized and standardized data respectively.

3.9 Plots comparing the estimates probability of activation using the multivariate probit model, when the response is totally known (x axis) and when there are missing values (y axis). (a) has 190 missing responses, (b) has 150 missing responses, and (c) has 50 missing responses.

4.1 Trace plot of the estimated values for the $\beta_1$ terms in each of the respective pathways (E2F3, Myc, Ras) for the simple univariate model. There was a draw of 100,000 and every 20th term was kept.

4.2 Trace plot of the estimated values for the $\beta_0$ and $\beta_1$ terms in each of the respective pathways (E2F3, Myc, Ras) for the intercept univariate model. There was a draw of 100,000 and every 20th term was kept.

4.3 Trace plot of the estimated values for the $\beta_0$ and $\beta_1$ terms in each of the respective pathways (E2F3, Myc, Ras) for the two metagene univariate model. There was a draw of 100,000 and every 20th term was kept.

4.4 Geweke diagnostic plots for the simple model $\beta_1$ (var1) terms for each of the three pathways (E2F3, Myc, Ras).

4.5 Geweke diagnostic plots for the intercept model $\beta_0$ (var1) and $\beta_1$ (var2) terms for each of the two pathways where the Geweke plots could be made (E2F3, Myc) The Ras pathway returned a non-convergence error and no plot could be generated.

4.6 Geweke diagnostic plots for the two metagene model $\beta_0$ (var1) and $\beta_1$ (var2) terms for each of the three pathways (E2F3, Myc, Ras).
4.7 Comparing the predicted probabilities of activation for univariate models. The columns are divided into the three different pairs of models. The first column compares the simple model (x axis) with the intercept model (y axis), the second column compares the simple model (x axis) with the two metagene model (y axis), the third column compares the intercept model (x axis) with the two metagene model (y axis). The rows indicate which pathway is being examined, the top row shows E2F3, the middle row shows Myc, and the bottom row shows Ras.

4.8 Credible intervals for the probabilities of activation in the tumours for the three univariate models are shown as individual boxes within each graph (simple, intercept, two metagene), for each of the three pathways, indicated by the separate graphs (E2F3, Myc, Ras).

4.9 Trace plot of the estimated values for each of the nine $\beta$ terms for the multivariate model. There was a draw of 25,000 and every 5$^{th}$ term was kept. $\beta_{11}$, $\beta_{12}$, and $\beta_{13}$ relate to the E2F3 pathway, $\beta_{21}$, $\beta_{22}$, and $\beta_{23}$ relate to the Myc pathway, $\beta_{31}$, $\beta_{32}$, and $\beta_{33}$ relate to the Ras pathway.

4.10 Trace plot of the estimated values for each of the nine $\beta$ terms for the alternate multivariate model. There was a draw of 25,000 and every 5$^{th}$ term was kept. $\beta_{11}$, $\beta_{12}$, and $\beta_{13}$ relate to the E2F3 pathway, $\beta_{21}$, $\beta_{22}$, and $\beta_{23}$ relate to the Myc pathway, $\beta_{31}$, $\beta_{32}$, and $\beta_{33}$ relate to the Ras pathway.

4.11 Trace plot of the estimated values for each of the nine $\sigma$ terms for the multivariate model. There was a draw of 25,000 and every 5$^{th}$ term was kept. $\sigma_{11}$, $\sigma_{12}$, and $\sigma_{13}$ relate to the E2F3 pathway, $\sigma_{21}$, $\sigma_{22}$, and $\sigma_{23}$ relate to the Myc pathway, $\sigma_{31}$, $\sigma_{32}$, and $\sigma_{33}$ relate to the Ras pathway.

4.12 Trace plot of the estimated values for the three of the $\sigma$ terms in the alternate multivariate model. There was a draw of 25,000 and every 5$^{th}$ term was kept. Only $\sigma_{11}$ showing the correlation between E2F3 and Myc, $\sigma_{22}$ showing the correlation between E2F3 and Ras, and $\sigma_{33}$ showing the correlation between Myc and Ras will be shown, as under the alternate model the other $\sigma$'s are set to be 1.

4.13 Geweke diagnostic plots for the nine multivariate model $\beta$ terms.

4.14 Geweke diagnostic plots for the nine alternate multivariate model $\beta$ terms.

4.15 Geweke diagnostic plots for the nine multivariate model $\sigma$ terms.

4.16 Geweke diagnostic plots for the three alternate multivariate model $\sigma$ terms which show correlation between the different pathways. (i.e. $\sigma_{11}$ for E2F3 and Myc, $\sigma_{22}$ for E2F3 and Ras, and $\sigma_{33}$ for Myc and Ras).

4.17 Comparing the predicted probabilities of activation for multivariate models and the simple univariate model. The columns are divided into the three different pairs of models. The first column compares the simple univariate model (x axis) with the multivariate model (y axis), the second column compares the simple univariate model (x axis) with the alternate multivariate model (y axis), the third column compares the multivariate model (x axis) with the alternate multivariate model (y axis). The rows indicate which pathway is being examined, the top row shows E2F3, the middle row shows Myc, and the bottom row shows Ras.
4.18 Credible intervals for the probabilities of activation in the tumours for the two multivariate models and the simple univariate model are shown as individual boxes within each graph (simple univariate, multivariate, alternate multivariate), for each of the three pathways, indicated by the separate graphs (E2F3, Myc, Ras) ........................................ 82

5.1 Lymph node status and pathway activation probabilities generated by the multivariate model ........................................ 87
5.2 Lymph node status and pathway activation probabilities generated by the univariate model ........................................ 87
5.3 Estrogen receptor status, pathway activation probabilities generated by the multivariate model ................................. 89
5.4 Estrogen receptor status, pathway activation probabilities generated by the univariate model ................................. 89
5.5 Progesterone receptor status, pathway activation probabilities generated by the multivariate model ................................. 91
5.6 Progesterone receptor status, pathway activation probabilities generated by the univariate model ................................. 91
5.7 (a) probability of activation of the E2F3 pathway classified by Estrogen Receptor (ER) status, for all tumours. (b) probability of activation of the E2F3 pathway for all tumours in cohort GSE7390, including the 95% credible intervals, split by classification of ER status ....................... 93
5.8 (a) probability of activation of the Myc pathway classified by Estrogen Receptor (ER) status, for all tumours. (b) probability of activation of the Myc pathway for all tumours in cohort GSE7390, including the 95% credible intervals, split by classification of ER status ....................... 93
5.9 (a) probability of activation of the Ras pathway classified by Estrogen Receptor (ER) status, for all tumours. (b) probability of activation of the Ras pathway for all tumours in cohort GSE7390, including the 95% credible intervals, split by classification of ER status ....................... 94
5.10 (a) probability of activation of the E2F3 pathway classified by disease free survival (DFS) status, for all tumours. (b) probability of activation of the E2F3 pathway for all tumours in cohort GSE7390, split by classification of DFS status ........................................ 96
5.11 (a) probability of activation of the Myc pathway classified by disease free survival (DFS) status, for all tumours. (b) probability of activation of the Myc pathway for all tumours in cohort GSE7390, split by classification of DFS status ........................................ 96
5.12 (a) probability of activation of the Ras pathway classified by disease free survival (DFS) status, for all tumours. (b) probability of activation of the Ras pathway for all tumours in cohort GSE7390, split by classification of DFS status ........................................ 97
Chapter 1

Introduction

In the past decade genomic technologies have become a vital component of biomedical research. One of the main reasons for this uptake is a drive towards “personalised medicine”, in which patient treatment is tailored to the molecular information specific to each individual. In this thesis we explore the use of Bayesian statistical analysis of gene expression microarray data, in the context of breast cancer. The data that will be used is from a large “meta-cohort” comprised of 2116 breast cancer patients from 14 distinct studies assembled by Dr Mik Black and his collaborators, Dr Lance Miller (Wake Forest University Medical School) and Dr Heather Cunliffe (Translational Genomics Research Institute).

1.1 Microarray technology

“Since each type of cell produces only its characteristic proteins- and not proteins characteristic of other cell types - it becomes apparent that differentiation of the cells of a multicellular organism depends on the inactivation of certain groups of genes and the activation of others.” - (Curtis and Barnes, 1989).

A gene is a heredity unit in all living organisms. It is a region of DNA (deoxyribonucleic acid) that codes for proteins and RNA (ribonucleic acid) chains. DNA and RNA are both nucleic acids that are made up of long chains of nucleotides. A nucleotide has three subunits, a phosphate group, a five-carbon sugar, and a nitrogenous base. The sugar base is either ribose (RNA) or deoxyribose (DNA), deoxyribose has one less oxygen atom then ribose. There are five different nitrogenous bases found in nucleotides, three of these are found in both DNA and RNA, adenine, guanine and cytosine, while
thymine is only found in DNA, and uracil is only found in RNA. Although their chemical components are very similar, DNA and RNA play very different biological roles. DNA is the carrier of the genetic message, while the function of RNA is to transcribe the genetic message from the DNA, and translate it into proteins (Curtis and Barnes, 1989).

Gene expression is the process by which information from a gene is used to generate proteins, by way of the intermediate product RNA. It is important to note some genes can produce “functional” RNAs, which are also included under the definition of gene product. By measuring the amount of these gene products, the activity of genes can be inferred. Abnormal activity can be associated with disease-causing alleles\(^1\), or with inappropriate activation of one or more genes.

The process of gene expression has two steps: the DNA sequence of a gene is transcribed into mRNA, which in turn is translated into protein. This process of transcription and translation is known as the “Central Dogma of Molecular Biology” (Crick, 1970).

\[\text{Figure 1.1: Basic Diagram of the “Central Dogma of Molecular Biology}}\]

Genes are transcribed into mRNA inside the nucleus of the cell. This process is initiated by the attachment of an RNA polymerase to a particular nucleotide sequence known as the “promotor” on one of the strands of DNA; the template or non-coding strand. As the RNA polymerase transverses the DNA strand, it uses complimentary base pairing to generate the messenger RNA (mRNA). The mRNA is an exact copy of the other strand of the DNA, known as the coding strand, except the Thymines are

\[^1\text{one of the multiple forms of DNA sequence associated with a particular gene}\]
replaced with Uracils. The gene coding strand contains two distinct sets of regions: Exons (coding regions) and Introns (non-coding regions). The introns are ignored during the transcription process, and are removed before the mRNA moves to the cytoplasm from the nucleus for the translation step.

The translation process is where information from the mRNA is used to make amino acids. Proteins contain twenty different amino acids, but DNA and RNA only contain four different bases, so these bases are arranged in to triplets known as codons. Each codon represents an amino acid, although it is worth noting that each amino acid can be represented by several different codons. The ribosome (a protein complex within the cell) builds an amino acid chain based on the ordering of the mRNA codons. Translation is terminated when a stop codon is encountered, causing the amino acid chain to be released. This amino acid chain is subject to further processing including folding, and the makeup and shape of the chain determines the final structure and function of the protein that is produced.

Genes are said to be expressing when they produce protein, so it is a valid assumption that by measuring the global amount of certain proteins present in cells, it would be possible to get an idea of the gene expression. Unfortunately it is very difficult to accurately measure the amounts of protein in cells in a single experiment, largely as a result of their complex three-dimensional structure, and lack of stability relative to nucleic acids. Therefore instead, moving one step back in the gene expression process, amounts of mRNA are measured. mRNA is unique to each gene, so by measuring the levels of mRNA in a cell, an idea of the gene expression can be obtained.

In 1975 Edwin Southern developed a method of measuring the quantity of specific DNA sequences, this method became known as “Southern blotting” (Southern, 1975). The “Southern blotting” method starts by cutting high-molecular weight DNA strands into smaller fragments. These smaller fragments are then passed through a gel matrix by way of electrophoresis, which is the use of electromotive force that moves molecules through the gel. Different sized molecules move through the gel at different rates, with the larger molecules moving more slowly. If the DNA strand is too big it can be broken down using acid. A membrane is applied to the top of the gel, even pressure is applied, and the DNA is transferred from the gel to the membrane. The DNA is bound to the membrane by ion exchange interactions, as the DNA is negatively charged and the membrane positively charged. The membrane is then baked or exposed to ultraviolet radiation to permanently attach the DNA to it. The hybridization probe is then
exposed onto the membrane, this involves the specific DNA fragment, which is marked with fluorescent dye or by incorporating radioactivity so it can be identified, being applied to the membrane where it attaches itself to the target DNA segments. After the hybridization, the membrane is washed to remove the excess probe and then the membrane can be examined to learn about the expression of the target DNA.

In 1977 “northern blotting” was invented, this was an adaptation of the “Southern blotting” and was used for measuring amounts of mRNA, thus providing a means to measure gene expression. Using traditional blotting methods to assay gene expression, researchers were only able to look at a few genes at a time (Ramaswamy and Golub, 2002), making the process slow.

The next major advance after blotting techniques was the use of arrays, a technology, which is similar to the analysis done in northern blotting but on a much more grand scale. The first type of arrays used from the mid 80’s were known as macroarrays, which are made by spotting DNA probes on a membrane-type material. With a spot size of 300 microns, this limited the number of probes that could be used to about 2000 per array (Parihar, 2007). The next progression was the development of microarrays, with early (non-commercial) microarrays created using higher density spotting techniques. These microarrays were created using a robotic pin-based system accurately dispensing DNA solution onto a spot of about 150 microns on a glass slide. Originally this DNA came from EST (expressed sequence tag) libraries, but is now more frequently obtained as synthetic oligonucleotides (short manufactured DNA sequences). The robot prints this information onto glass slides that are coated with a special material to allow the DNA to attach to them, each of the spots is known as a probe. The samples also have to be prepared, most often this is complementary DNA (cDNA) generated from mRNA from the cells we wish to hybridize to the slide. Commonly a two channel hybridization experiment is performed where mRNA samples from two experimental conditions are taken, and each sample is purified. The two samples are then reverse transcribed into cDNA and during this process a fluorescent dye is attached to the samples (in effect labelling the cDNA’s). Most often these are Cy5 (red) and Cy3 (green). The two samples are then mixed and pipetted onto the microarray. Ideally the cDNA transcripts will only bind to their complementary probes. The slide is washed and put in a centrifuge to remove any of cDNA that has attached to the slide. Then it is scanned, using a laser to excite the labels to fluorescence, producing an image of where the genes are expressing, green indicating one condition, red in the other, and colours in between
these indicating different ratios of each experimental condition.

One of the most commonly used microarray platforms is the Affymetrix commercial system, which builds the oligonucleotides directly onto the surface of the array. This technique is known as photolithography (Chee et al., 1996), and is very expensive so lends itself to mass-production. The limitation of using commercially produced microarrays is the loss of the customization that the lower cost spotted arrays allow, meaning sometimes the array may include genes that the experimenter is not interested in. The benefit is that these arrays are very precise and allow high-density spots, with more than one million probe sequences per array.

Unlike spotted arrays, only a single sample can be used on these arrays, therefore experiments on these chips only involve a single-channel hybridization. One of the benefits of a single channel hybridization is that an aberrant sample cannot affect the raw data produced by the other sample, whereas in the two channel system one low-quality sample can affect the overall precision even if the other sample is of high quality. The obvious drawback of using a single channel system over a two channel one is that twice as many arrays are required (Mahendra, 2010).

The extent to which microarray technology has advanced the ability to genetically analyse biological systems has been compared to how microprocessors have accelerated computation (Zhou and Thompson, 2004; Parihar, 2007), although a common complaint about microarray technologies is that the studies associated with them tend to be more descriptive then analytical (Golub et al., 1999; Zhou and Thompson, 2004).

The crux of microarray technology is its ability to exploit the way a given mRNA molecule binds specifically to the DNA template from which it originates. The expression levels of specific genes is calculated by estimating the amount of mRNA attached to each “spot” on the array. This information can then be used to produce a gene expression profile for each sample.

1.2 Biology of breast cancer

Breast tissue is a complex network of lobules and mammary ducts, resembling a bunch of grapes (http://ww5.komen.org/default.aspx, 2009a). The lobules are where the breast milk is created, and the mammary ducts transport the milk to the nipple. Around 85% of cases of breast cancer start in the mammary ducts, with the remaining
15% starting in the lobules.

In the healthy body, there are natural systems that control creation, growth and death of cells. When the body is injured, with a cut for example, cell production can be increased to repair the body. Cancer occurs when the regulators of cell growth stop working, and cell growth start to exceed cell death (Hanahan et al., 2000). This abnormal growth can lead to tumour formation, which in turn promotes angiogenesis (production of new blood vessels) to bring in nutrients and oxygen. Metastasis occurs when cancer cells leave the tumour site and travel throughout the body in blood vessels, or in the lymphatic system, to establish new tumours at distant sites.

Cancers are associated with damage (mutations) to the genes that regulate cell growth and division (Sherr, 2000), or inherited alleles (Walsh and King, 2007) or epigenetic change (Jones and Baylin, 2002). This damage can be hereditary, or caused by exposure to radiation or other mutation inducing chemicals like those found in cigarettes. The damage may also occur from spontaneous errors during normal cell growth and death. If these mutations occur in specific genes that control cell proliferation and these changes are copied into the next generation of cells created, then as these cells are altered further this leads to uncontrolled proliferation of the mutated cells.

There are two main types of breast cancer “Invasive Breast Cancer” and “Ductal Carcinoma in situ”. Ductal Carcinoma in Situ (DCIS) occurs when abnormal cells grow inside milk ducts but have not spread into surrounding tissue, making it non-invasive. It is possible for DCIS to develop into an invasive breast cancer. Invasive breast cancer occurs when the abnormal cells that have grown in the lobes or ducts become mobile and begin to spread into surrounding tissue. Breast cancer is divided into five major stages (Rosen, 2001), with stages II and III being divided into 2 and 3 subclassifications respectively. The initial stage is called stage 0 is a non-invasive cancer such as DCIS, with no evidence of it moving away from the site of origin. The scale then goes through invasive cancers, differentiated by different sized tumours and the amount of involvement with the lymph nodes. The final stage is stage IV which is when the cancer has spread to other organs in the body. Stages 0 - IIIa are considered early stages, and stages IIIb-IV are considered advanced. Table 1.1 gives the expected survival of people who were diagnosed with breast cancer in 2001 and 2002 taken from
the National Cancer Data Base (http://www.cancer.org/, 2010)

Table 1.1: Survival rates for each of the Stages of Breast cancer. Based on data collected by the National Cancer Data Base from 2001 and 2002.

<table>
<thead>
<tr>
<th>Stages</th>
<th>Survival Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>93</td>
</tr>
<tr>
<td>I</td>
<td>88</td>
</tr>
<tr>
<td>IIa</td>
<td>81</td>
</tr>
<tr>
<td>IIb</td>
<td>74</td>
</tr>
<tr>
<td>IIIa</td>
<td>67</td>
</tr>
<tr>
<td>IIIb</td>
<td>41</td>
</tr>
<tr>
<td>IIIc</td>
<td>49</td>
</tr>
<tr>
<td>IV</td>
<td>15</td>
</tr>
</tbody>
</table>

In New Zealand there are 2500 new cases of breast cancer per year in women, resulting in around 600 deaths annually (http://www.nzbcf.org.nz, 2009). Worldwide the figure is put between 1.15 million and 1.5 million new cases of breast cancer that are diagnosed each year in woman, leading to 465,000 deaths. The rate of death is decreasing due to early detection and better treatments (Jemal et al., 2010).

Breast cancer is a collection of multiple diseases, which are differentiated by the presence of different molecular drivers (Di Cosimo and Baselga, 2010). Knowledge of which of these drivers are present in a particular case of breast cancer can be used to make clinical decisions regarding treatment and management. The three most relevant are human epidermal growth factor 2 (Her2), estrogen receptor (ER) and progesterone receptor (PR).

Estrogen receptors (ER) are protein molecules that contain sites to which only estrogen or closely related molecules can bind. Estrogens are molecules that stimulate development and maintenance of female characteristics. In the breast, estrogen is responsible for generating the cells that form the lining of the inside of the milk glands during the menstrual cycle. It is this cellular proliferation that makes estrogen dangerous in breast cancer, as it is accepted that estrogen does not cause the cancer causing mutations, but the increased production of cells increases the likelihood that a mutation will occur (National Cancer Institute, 2006). Breast cancer can be either ER positive or ER negative.
Adjuvant endocrine therapies (also known as hormone therapies) are used in the treatment of ER positive breast cancers, which work by blocking the production of estrogen, thus inhibiting tumour growth. These therapies range from the extreme, where the ovaries are removed as this is the body’s main source of estrogen (this method is used sparingly though especially in younger patients), to the less extreme, such as drug therapy. There are two main types of drugs used. In pre-menopausal woman selective estrogen receptor modulators (SERMs), one of the most common being tamoxifen, are the main choice: these work by blocking estrogen in the breast tissue. The main side effect of SERMs is the onset of menopausal like symptoms due to estrogen being blocked in the rest of the body. In post-menopausal woman the body no longer produces as much estrogen, rather the adrenal glands produce androgen, and the enzyme aromatase found in the fat cells can convert this androgen into estrogen. In these woman aromatase inhibitors (AIs) can be used to block the production of aromatase, reducing the amount of the converted estrogen in the body. AI are associated with increased risk of osteoporosis, raised cholesterol levels, and enhanced menopausal symptoms. Since endocrine therapies are only effective in ER positive tumours, it is important to be able to recognise which cancers are ER positive to avoid unnecessary adjuvant endocrine therapies on woman with ER negative tumours as they will be subjected to the side effects without there being any improvement in their condition.

An ER negative cancer does not contain estrogen receptors, and therefore cannot be treated with tamoxifen because there are no estrogen receptors to inhibit (National Cancer Institute, 2006). Work has been done with microarray technology to classify the ER and Lymph node status of breast cancer (West et al., 2001).

Another receptor that is highly correlated with ER is the progesterone receptor (PR). Progesterone is an ovarian steroid hormone that requires estrogen to be produced. Studies have shown that PR status provides supplementary information to ER status in terms of survival (http://www.labvision.com, 2009). Traditionally it was thought that PR added no new information (Fisher, B. and Redmond, C. and Fisher, E.R. and Caplan, R., 1988), however more recent studies have shown if PR status is measured correctly than it is an independent predictive factor for benefit from adjuvant endocrine therapies (Bardou et al., 2003).

Another classification that can be applied to breast cancer is whether or not it is
Her2 positive or Her2 negative. Her2 is a protein (produced by the ErbB2 gene) found on the surface of certain cancer cells, and is one of the receptors for human epidermal growth factor. The human epidermal growth factor receptor 2 is a protein involved in regulating normal cell growth, these receptors are located on the surface of the cells. If there is a lot of Her2 present in the breast tumour, the tumour will respond more strongly to growth promoting signals and will grow faster. These tumours tend to be more aggressive and are less sensitive to chemotherapy and hormone therapy. Her2 is present in roughly 25% of all breast cancers. (Dowsett et al., 2000).

Patients who are Her2 positive can benefit from treatment with trastuzumab (colloquially known as Herceptin), a drug that works by attaching itself to the Her2 molecule preventing the human epidermal growth factors attaching themselves. Herceptin is only really effective in patients who exhibit high levels of Her2 (http://www.macmillan.org.uk, 2009). It is also known that Her2 positive cancers do not respond as well to tamoxifen as Her2 negative cancers.

Breast cancers can be classified into four main molecular subtypes, based on the status of the receptors discussed above. Luminal A, where the patient is ER positive, possibly PR positive and Her2 negative, this the most common subtype of breast cancer making up 42-59% of the cases (http://ww5.komen.org/default.aspx, 2009a); Luminal B where the patient is weakly ER positive, possibly PR positive and possibly Her2 positive. Triple negative or basal like, where all three of ER,PR and Her2 are negative, this type is more common amongst younger patients (Haffty et al., 2006). The final type is Her2 positive cancer, where the patient is ER and PR negative but Her2 positive. The ability to be able to accurately classify breast cancers in terms of the molecular characteristics provides the potential for more targeted treatments, and as a result, better patient outcomes. Microarray technology provides a possible means to achieve this goal.

Other clinical information used for prognosis and for making decisions about treatment include the age of the patient, tumour grade, tumour size, and lymph node status. The susceptibility to breast cancer increases with age, and while younger woman certainly do develop breast cancer, over 95% of cases occur in woman aged over 40 (http://ww5.komen.org/default.aspx, 2009b). Lymph node status can tell the physician how advanced the cancer is; if there is cancer in the lymph node, the patient is said to be lymph node positive and if there is no cancer present they are called lymph node
negative. If the cancer is only in the breast and not in the lymph nodes the patient has a better prognosis. While a physical examination can give an estimate of the number of positive lymph nodes, the more accurate way is for a pathologist to exam the nodes removed during surgery and assess the presence of cancerous cells that have migrated from the tumour.

Tumour grade is an important tool in prognostics, with tumours being put into one of three grades. Tumours are graded by three main factors, cancer cells’ homogeneity to normal cells, the rate of division (mitotic rate), and tubule formation. Each of these factors is scored 1-3, with these scores added together to classify the cancer into the grades (Elston, 1984). As cancer arises from the transformation of normal cells, the more closely the cancer cells resemble normal cells, the less advanced the cancer. A cancer cell similar to normal cells is given a score of 1, cells that are highly abnormal score 3. Cancer is also associated with the breakdown in the regulation of cell division and death, if there is higher number of cells in the process of division then the more advanced the cancer. A low mitotic rate is scored as a 1, and a high mitotic rate is scored as a 3. Tubule formation gives a way to assess the tissue organisation, if most of the cells (more than 75%) in the tumour have tubular formation this is considered normal, and is scored as a 1. If less than 10% of the cells are in tubular formation this indicates the cellular structure has been broken down, so these tumours are scored as a 3. These individual scores are then combined to define the grade of the tumour. If the pathologist finds the cells are similar to normal cells, with all tubular formation, and slow growing, the tumour, (a combined score between 3-5), will be graded as “Grade 1” (also known as Well-differentiated), if there are abnormal cells not in a tubular formation, that are fast growing the tumour, (a score of 8 or 9), is defined as “Grade 3” (also known as Poorly-differentiated). Tumours with scores of 6 or 7 are defined as “Grade 2”.

1.3 Biomedical applications of microarrays

The ability to better classify cancers is leading to improvements in treating cancers effectively (Golub et al., 1999). Traditional methods of cancer classification are based on morphological appearances, these methods are limited by the fact that even if two tumours give similar histopathological appearance they can show different responses, and also this analysis relies on biological insight rather than systematic and unbiased
approaches (Golub et al., 1999).

Golub et al. (1999) investigated using microarrays in class prediction, which they defined as “the assignment of particular tumour samples to already defined classes”. The classes they tested were AML (acute myeloid leukemia) and ALL (acute lymphoblastic leukemia). They acknowledged that the distinction between these classes had been well established, but traditional methods of doing this require more than one test. Usually the distinction requires an experienced hematopathologist interpretation of three or four tests that each require their own separate highly specialised laboratories and there are still sometimes errors.

Golub et al. (1999) wished to simplify this process by taking advantage of microarrays’ ability to monitor the expression of thousands of genes. They established using a “neighbourhood analysis” that the observed associations between gene expression and class were stronger than expected by chance. They then used the known samples to develop a class predictor, it was based on each informative gene\(^2\) casting a “vote” towards one class or the other. The vote was weighted depending on the expression level in the new sample and the extent of gene’s correlation with the classes. The votes were used to determine which class the sample related to the most, a prediction strength was also calculated. Only samples whose prediction strength exceeded a predetermined score, would the sample be classified, otherwise it was considered undefined.

Since the initial work of Golub et al. (1999) numerous publications have used microarrays to investigate the molecular characteristics of cancer (West et al. (2001); Huang et al. (2003); Eifel et al. (2001), etc...).

1.4 Microarrays applied to breast cancer

West et al. (2001) were one of the first to develop Bayesian regression models capable of prediction based on gene expression data derived from DNA microarray analysis. Like Golub et al. (1999), West et al. (2001) regarded prognostic and predictive factors as very important in the treatment of tumours. The use of DNA microarray analysis of human tumours can distinguish tumour classes, where traditional histopathological methods cannot. West et al. (2001) used breast cancer, as it is a disease where more

\(^2\)Genes highly correlated with class distinction
classification is needed to improve diagnostic and therapeutic strategies. They sug-
ggested that because of the vast amounts of information extracted by measuring gene
expression using DNA microarrays coupled with the fact there is no assumptions re-
garding what type of pathway could be affecting a particular tumour, the analysis of
gene expression could make clinical decisions more informed, because of the more pre-
cise determination of tumour cell phenotypes. Additional work in this area has shown
that gene expression phenotypes have potential to characterise the complex genetic
alterations within tumours (Huang et al., 2003).

Microarrays have also been used for a more accurate means of prognostication in
breast cancer to improve selection for adjuvant systemic therapy. van de Vijver et al.
(2002) used microarray analysis to classify 295 primary breast carcinomas as having
gene expression associated with either good or bad prognosis. They found that this
was a more powerful predictor than traditional prognosis tools based on clinical and
histologic criteria (e.g age, tumour size, status of axillary lymph nodes etc). This is
important for the selection of patients to receive adjuvant systemic therapy. Adjuvant
systemic therapy involves the administration of therapies such as chemotherapy or ra-
diation therapy following initial surgery on patients with early breast cancer. This
type of treatment is used to prevent, or delay, the spread of cancerous cells. These
treatments are associated with side effects that can lead to a deterioration of quality
of life (Shapiro and Recht, 2001). Patients with poor prognosis benefit the most from
adjuvant therapies (Eifel et al., 2001).

Another application is with the histological grading of breast cancer. With histo-
logical grading there is a sliding scale, and one of the more common complaints with
it is with the interobserver disagreement especially with moderately differentiated or
“Grade 2” tumours. Ivshina et al. (2006a) used microarray analysis to distinguish
between the different grades and found they could accurately classify “Grade 1” and
“Grade 3” tumours and also separate “Grade 2” tumours into two discriminant classes
“Grade 2a”, which are closer to “Grade 1” and “Grade 2b” which were closer to “Grade
3”.

Of most interest in this thesis is the work performed by Bild et al. (2006) where
the authors set out to show that gene expression signatures can be identified to reflect
activation status of several oncogenic pathways. The activation of multiple indepen-
dent oncogenic pathways is integral in the development of an oncogenic state. Human primary mammary epithelial cell cultures (HMEC’s) were used to develop a series of pathway signatures for the known oncogenes β-Catenin, E2F3, Myc, Ras, Src. The generated gene expression signatures gave good distinction between those cells expressing oncogenic activity and the control cells, using principle components to define “metagenes” summarizing pathway activation level. By using the top three principal components they found that they could distinguish not only between the oncogenic cells and the controls, but between the oncogenic cells expressing the different pathways as well.

To verify their signatures, Bild et al. (2006) combined human and mouse cancer datasets. The signatures were regenerated from the genes common to both human and mouse data, and the analysis was trained on the human signatures, and then used to predict the pathway status of all the tumours. Only the E2F3, Myc and Ras pathways were used, since these were the ones for which matching mouse models existed, which could be used for validation. The validation process showed good results, with the predictions showing good correlation with the molecular basis for the tumour induction. The predicted probability for Myc pathway deregulation was highest in the tumours with the (MMTC)-Myc virus, similarly with the other two pathways. Further validation was done using tumours in which Ras activity was spontaneously activated, which more closely mimics the pathway deregulation in human tumours. Again the signatures did well in predicting the pathway deregulation in these tumours and the control lung tissue. These two validation processes were seen as evidence that these signatures can be used to probe the status of pathways.

Using hierarchical clustering Bild et al. (2006) showed that tumours exhibiting low levels of Ras deregulation showed high levels of deregulation in the other pathways (β-Catenin, E2F3, Src and Ras), and the converse was also true. This clustering also identified that patients exhibiting high levels of deregulation of Ras and β-Catenin, Src and Myc, showed poor survival for lung cancer patients. This was done using only the gene expression information independent of the histopathology, indicating that pathway analysis based on multiple signatures can be used to successfully categorize of lung cancer patients. The clustering was also used on breast and ovarian cancer datasets, which also showed clear patterns of pathway deregulation. In all three types of cancer the overwhelming theme was that individual pathway deregulation did not stratify the
patients as well as patterns of multiple pathway deregulation.

1.5 Summary

The first part of this chapter introduced basic molecular biology, defining genes, how they are made up of DNA, and their involvement in the generation of proteins. The “Central Dogma of Molecular Biology” was discussed which gives more detail of how genes are turned into proteins through the processes of “Transcription” and “Translation”. A gene is said to be expressing when they produce proteins, so by measuring the levels of mRNA an idea of the level of gene expression can be obtained. The development of tools that were initially used to measure DNA sequences through to the modern microarrays which measure mRNA expression was discussed. Modern microarrays give the ability to measure ten of thousands of genes per sample in a quick and efficient manner.

Section 1.2 gave an introduction to the biology of breast cancer, as well as an overview of the different types of breast cancer that exist. The histological tools used to classify breast cancer were summarized, including but not limited to, lymph node status, ER and PR receptor status, and tumour grade and size, and their limitations were discussed. Classifying breast cancer correctly is very important as it can aid treatment choices, and to avoid unnecessary adjuvant therapies which can lead to decreased quality of life. Various studies that have explored using microarrays in biomedical applications including ones that deal with breast cancer were reviewed. In the coming chapters the use of Bayesian methods for the analysis of microarray data will be used to examine genomic data from breast cancer patients.
Chapter 2
Bayesian Factor Regression Modelling

2.1 Microarray analysis

The techniques that are used in this thesis to analyse microarray data are based upon the work done in Bild et al. (2006) who in turn used techniques developed by West et al. (2001). West et al. (2001) combined dimension reduction via singular value decomposition (SVD) with class prediction via Bayesian regression: they termed this methodology “Bayesian Factor Regression Modelling” (BFRM). This chapter begins by explaining the idea of dimensionality reduction starting with the early PCA techniques and then moving towards the more modern SVD, as well as the rationale for using it in microarray analysis. Then the univariate probit regression model that was used in the Bayesian regression part of BFRM will be introduced. Once these central ideas have been discussed, more specifics surrounding the details of the Bild et al. (2006) publication will be discussed, including the data used, and how the techniques were applied. Then the application to a large multicohort dataset will be talked about, and some diagnostics done on the model in both R and Python.

2.2 Dimensionality reduction

Alter et al. (2000) state that DNA microarray technology has advanced to the point that it is now possible to monitor gene expression levels on a genomic scale, however analysis of these new data requires mathematical tools that are adaptable to large quantities of data while reducing the complexity of the data to make them comprehen-
The problem is that microarray analysis commonly has a small number of arrays, with a large number of genes, typically at most hundreds of arrays and with a gene count numbering in the tens of thousands. Even in a study where you may have a moderately large number of arrays, the sheer size of the human genome will mean the number of genes will always outweigh the number of arrays. In the context of modelling, where the effect of each gene can be represented by a parameter, statisticians commonly refer to this issue as a “large \( p \) small \( n \) problem”, where \( p \) is the number of parameters, and \( n \) is the number of samples. These types of datasets cause problems because there is not enough information to accurately estimate the parameters, they also lead to overfitting, and overfit models generally have poor predictive qualities (Hawkins, 2004). One of the ways to deal with this sort of data is to use some sort of dimensionality reduction tool, which can reduce the number of parameters from thousands down to more manageable numbers, so improving the ratio of \( p \) to \( n \), and also aiding in the identification of patterns in noisy gene expression data. Two useful tools commonly used in the analysis of multivariate data are principal component analysis (PCA) and singular value decomposition (SVD) (Golub and Reinsch, 1970).

The idea of PCA was first conceived in 1901 (Pearson, 1901) and independently developed by Hotelling (1933). Jolliffe (2002) states that the central idea of PCA is “to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set”. A common visual description of PCA is that, if a multivariate dataset is visualised as a set of coordinates in a high-dimensional data space, then PCA supplies the user with a lower-dimensional picture, or a “shadow” of this object when viewed from its most informative viewpoint. The most informative viewpoint is the one that captures the most variation along the principal axis, that is, the one that best explains the major trend within the data.

Given a \( m \times n \) matrix \( X \), to perform PCA, the first step is to take the mean of each of the \( m \) columns, the mean is then subtracted from each of the respective columns, to give a dataset with a mean of zero. The variance-covariance matrix of the transformed datasets columns is then calculated, it will be of dimension \( m \times m \). From this variance-covariance matrix the eigenvalues and eigenvectors are calculated, the eigen-
vectors provide information about the pattern in the data. The eigenvalues can be used to rank the importance of the information, the eigenvector corresponding to the highest eigenvalue is the first principle component. So the next step is to rank the eigenvalues from highest to lowest, to rank the components in order of proportion of variation explained. To reduce dimensionality the lower ranked components are removed. While some information is lost, if the eigenvalues are low this loss in minimal. More details can be found in Smith (2002)

As mentioned in section 2.1, however, West (2003) suggests using singular value decomposition (SVD) for dimension reduction in microarray experiments. The basic concept of SVD is that given the same \( m \times n \) matrix \( X \) from the PCA explanation, then the singular value decomposition of matrix \( X \) is given by

\[
X = U D V^T
\]  

(2.1)

where \( U \) and \( V^T \) are orthogonal matrices containing the left and right singular values of \( X \) respectively, and \( D \) is a diagonal matrix of singular values. One way to calculate the SVD is to first calculating \( V^T \) and \( D \) by diagonalizing \( X^T X \) into

\[
X^T X = V D^2 V^T
\]  

(2.2)

where the columns of \( V \) are the eigenvectors of \( X^T X \) and the the diagonal entries of \( D \) are the square roots of the associated eigenvalues. Using basic matrix algebra it is easy to show that

\[
V^T = D^{-1} U^T X
\]  

(2.3)

Using the rearrangement from equation 2.3 it is possible to project other matrices onto the same space as \( V^T \).

When principal components are calculated from the covariance matrix there is a direct relation between PCA and SVD (Wall et al., 2003). If the mean of each column of \( X \) is made to be equal to zero, then \( X^T X \) is proportional to the covariance matrix of the rows of \( X \). In equation 2.2 it was shown in the diagonalization of \( X^T X \), the eigenvectors make up the columns of \( V^T \), so the right singular vectors are the same as the principle components.

### 2.3 Probit regression model

In this thesis, a number of probit regression models are discussed, but they can be divided into two overall classes, univariate (this chapter) and multivariate (see Chapter
3.1). The term univariate was applied in situations where the activation probabilities of each pathway are calculated using only information from genes involved in that pathway as defined by Bild et al. (2006), whereas in the multivariate models information from all of the pathways that were investigated is used to calculate the probabilities of activation for each pathway.

Probit regression is a member of a class of models known as generalised linear models (McCullagh and Nelder, 1989). The probit acts as a link function, which in the case of data from a binomial distribution maps the interval (0,1) on to the whole real line. The probit regression models the probability that the binary response variable

\[ y_i = 1 \]

as

\[ \pi_i = \Phi(x_i'\beta) \] (2.4)

where \( \Phi() \) is the cumulative distribution function (CDF) of the standard normal distribution, \( x_i \) is the vector of predictor variables, and \( \beta \) is a vector of regression parameters (West et al., 2000).

The rationale behind using the probit model lies in its ability to exploit the latent variable structure (West et al., 2000). West et al. (2000) state the reason for using this representation is that Bayesian analysis of the binary regression model maybe routinely implemented using MCMC methods that incorporate the inherent latent variables \( y_i \) as missing data to be imputed and inferred along with the parameter \( \beta \).

\[ y_i = 1 \quad \text{if} \quad y_i^* > 0 \] (2.5)

where \( y_i^* \) is a latent quantity defined as

\[ y_i^* = x_i\beta + \epsilon_i \] (2.6)

This ability to estimate values for unknown \( y_i \)'s is exploited in the analysis of microarray data. The univariate probit model can be implemented in R using the \texttt{rbprobit} function which is included as part of the \texttt{bayesm} package in R authored by Peter Rossi. With Bayesian analysis of binary regression, a Gibbs sampler can be used to impute the missing values of \( y_i^* \) along side the \( \beta \) parameters (Albert and Chib, 1993). The details of the univariate Gibbs sampler used will be discussed in more detail in chapter 4.10.

\[^1\text{In West et al. (2000) they use } z_i \text{ as their response variables, however for continuity of notation we will use } y_i \text{'s.}\]
2.4 Microarray analysis

Following the statistical techniques for microarray analysis of breast cancer data used by Bild et al. (2006) the first step is to find the sets of genes whose expression levels are most strongly associated with the activation of oncogenic pathways. For this thesis the gene lists used were the ones published Bild et al. (2006), so as to maintain consistency with the results produced by their methodology.

Once the gene lists have been generated, SVD is performed on these set of genes to create ‘metagenes’\(^2\) (Bild et al., 2006; West et al., 2000; Alter et al., 2000). When West et al. (2000) originally did this they referred to these genes as ‘supergenes’, but the name has been changed since then.

When performing the SVD, the data can be split into two collections, the first being the normal and oncogenic cell lines that have been prepared in vitro, so that all the information about these cell lines are known. The second component is tumour data from patients that have suffered cancer. In the data the activation of the pathways is unknown. Obviously running SVD on all the samples would not give good differentiation due to the mixed nature of the patient samples, creating uninformative singular values. Instead the SVD is only performed on the first collection, as this has the best contrast between each pathway being either active or not. The tumour data can then be projected into this space using equation 2.3.

For the analysis in this thesis the first component, the normal cell line and oncogenic cell lines for five different pathways, comes directly from the experiments of Bild et al. (2006). The cell lines were created from human mammary epithelial cells from breast reduction surgery at Duke University that had been isolated and cultured using techniques developed by Stampfer and Yaswen (1993). These techniques allow for the long-term growth of normal lifetime cells in a serum free environment, providing one of the best ways to examine the differences between normal and transformed cells in expression of cell cycle, and more importantly for this study, differentiation related properties.

\(^2\)This is of course distinct from the metagenes that when subject to extraordinary physiological stress activate, giving people superhuman abilities in the DC Universe (Giffen, K. and Mantlo, B., 1989)
The cells are brought to quiescence, the period were they are no longer growing or in a cell cycle, and then infected with adenovirus expressing the green fluorescent protein (GFP) marker and one of human c-Myc (Myc), activated H-Ras (Ras), human c-Src (Src), human E2F3 (E2F3) or activated β-catenin (Bcat). Adenoviruses are a family of DNA viruses used to deliver genes in many forms of gene therapy (Schiedner et al., 1998). In the normal cells the adenovirus is still applied like a placebo, however the gene that is overexpressing is only the one that produces the GFP marker. GFP is used to help identify cells that are expressing the inserted gene. First the GFP gene is spliced into the gene of the protein of interest, so when the protein is produced, GFP is also produced. Then it is easy to spot if the protein of interest has been activated due to the distinctive fluorescence of the GFP (Chalfie et al., 1994).

There were ten normal cell line arrays that were used, but for each of the oncogenic cells lines there were different amounts of array data available. This is summarized in Table 2.1.

<table>
<thead>
<tr>
<th>Oncogenic Pathway</th>
<th>Number of arrays</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bcat</td>
<td>9</td>
</tr>
<tr>
<td>Myc</td>
<td>10</td>
</tr>
<tr>
<td>Ras</td>
<td>10</td>
</tr>
<tr>
<td>Src</td>
<td>7</td>
</tr>
<tr>
<td>E2F3</td>
<td>9</td>
</tr>
</tbody>
</table>

The second component, the breast tumour data comes from a large “meta-cohort” comprised of 2116 breast cancer patients from 14 distinct studies assembled by Dr Mik Black and his collaborators, Dr Lance Miller (Wake Forest University Medical School) and Dr Heather Cunliffe (Translational Genomics Research Institute). This dataset is referred to BC2116, Table 2.2 lists each of the cohorts, including the number of patients and publications they were involved in.
The data that is produced from these cohorts is stored with each gene examined being a row, and each sample being a column. As well as the gene expression data to be used in the statistical inference, clinical information was collected from the woman, this includes (but is not limited to) estrogen receptor status, lymph node status, grade, age of patient, size of tumour. In this dataset not all of the woman had complete clinical information meaning that there is a lot of missing data. For the purposes of comparing the clinical information versus the gene expression data, the focus here will be on lymph node status, progesterone and estrogen receptor status.
2.5 Generating the metagenes

In this section and in section 3.1 the creation of metagenes is discussed. It is important to note that before the SVD was performed as part of any of these methods, all the genes were standardised to have a mean of zero and a standard deviation of one.

There are various methods to generate the metagenes: in the overall study, three have been used. In the multivariate chapter (Chapter 3), there needs to be information about all the pathways, not just from the pathway of interest, so methods that include this information will be discussed in Chapter 3.1 These methods will be referred to as generating “multivariate metagenes”, not because they use multivariate techniques but because of the models in which they will be used. This nomenclature is used to differentiate those metagenes, from the metagenes discussed here where the only pathway, whose information is included is the one of interest. These metagenes will be referred to as univariate metagenes, again named for the models they will be used in, as opposed to techniques used in their generation.

In the univariate model SVD is performed just on the normal cell lines and those cell lines associated with the particular pathway of interest. For example, if the metagene for the BCat pathway is being generated, SVD is performed on the combined normal and BCat cell lines (19 samples in total). Using the transformation matrix generated from SVD, the tumour cell lines are transformed to be in the same space as the normal and BCat cell lines. The two matrices, the cell lines and the transformed tumour, are then combined to give the BCat metagenes.
Figure 2.1: Each individual graph shows a scatterplot of the normal and pathway cell lines for each of the 5 pathways with the primary metagene as the x axis and the secondary metagene as the y axis. The metagenes used were generated using the “univariate method”. The cell line points are colour coded with the normal being green, BCat- Pink, E2F3 - Blue, Myc - Red, Ras - Black, Src - Cyan.

This method provides good distinction between the normal cell lines and the oncogenic cell lines as evidenced in Figure 2.1. The figure shows the normal cell lines at the lower extremities (-0.3) of metagene one, and the oncogenic cell lines (0.3) at the upper extremities of metagene one. This should be expected, as the activation levels of oncogenic cells in the normal cell lines should be zero and the level of activation in the oncogenic cell lines should be higher. The other aspect these graphs highlight is the fact that most of the information needed to differentiate the cell lines is provided in metagene one. If Figure 2.1 is observed in the vertical axis, it is clear that there is virtually no differentiation between the normal and oncogenic cell lines. This
information provides the first evidence to the fact that only the primary metagene is
needed from each pathway for analysis, as that metagene contains most, if not all, of
the information for differentiating the normal and oncogenic cell lines.

2.6 The three models

In the original univariate model the metagenes generated using the univariate method
were used. This means that only the information related to the activation of the
pathway of interest is included. This automatically precludes the possibility that if
there is any correlation between any of the oncogenic pathways activation, it will be
discovered.

Three variations of a univariate model were experimented with to see which would
give the best results. Only three pathways were included, because, later when the
multivariate models were introduced, problems were encountered with the BCat and Src
pathways, so they were removed from analysis. To allow clearer comparisons between
the univariate and multivariate models it was decided to remove the BCat and Src
pathways from the univariate analysis.

With this univariate model the initial model used took the form

$$y_i^* = x_{1i} \beta_1 + \epsilon_i$$  \hspace{1cm} (2.7)

where the $x_{1i}$ was the first, and therefore dominant, in terms of information provided,
metagene. The second variation was the inclusion of the second metagene ($x_{2i}$) into
the simple model to see if the added information aids the model.

$$y_i^* = x_{1i} \beta_1 + x_{2i} \beta_2 + \epsilon_i$$  \hspace{1cm} (2.8)

The third variation was to see whether or not the inclusion of a intercept term
improved the simple model, giving a third model.

$$y_i^* = \beta_0 + x_{1i} \beta_1 + \epsilon_i$$  \hspace{1cm} (2.9)

2.7 Gibbs sampling/Data imputation for the univariate

Gibbs sampling is a special case of the Metropolis-Hastings sampler (Chib and Green-
berg, 1995). It works on the idea that given a multidimensional distribution it is
simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution. Casella and George (1992) give a succinct example using the bivariate case. Given a pair of random variables \((X, Y)\), the Gibbs sampler generates a sample from \(f(x)\) by sampling from the conditional distributions \(f(x|y)\) and \(f(y|x)\). From these distributions it’s possible to generate a Gibbs sequence of random variables. A Gibbs sequence takes the form

\[ Y_0, X_0, Y_1, X_1, Y_2, X_2, \ldots, Y_k, X_k. \tag{2.10} \]

The sampler is normally initiated by giving the value of \(Y_0 = y_0\) where \(y_0\) is an arbitrary starting value, and the rest of the sequence is generated iteratively by alternating between

\[ X_i \sim f(x | Y_i = y_i) \]

and

\[ Y_{i+1} \sim f(y | X_i = x_i) \]

It can be shown that the distribution of \(X_k\) converges to \(f(x)\) as \(k \to \infty\). Therefore for a large enough \(k\), \(X_k\) is a sample point from \(f(x)\).

In section 2.3 the simple univariate probit model was introduced and discussed in general. To refresh, the equations associated with this model are

\[ y_i = 1 \quad \text{if} \quad y_i^* > 0 \tag{2.11} \]

Where \(y_i^*\) is a latent quantity defined as

\[ y_i^* = x_i \beta + \epsilon_i \tag{2.12} \]

In this specific implementation of the model, the \(y_i^*\)’s indicate whether the pathway of interest is activated or not, the only activation data available is for the normal and oncogenic cell lines. The \(y_i^*\)’s for the oncogenic cell lines of the pathway of interest will take a value of 1, and the normal and other oncogenic cell lines take the value of 0. The patient tumour pathway activation data is missing; this of course is the parameter of interest, as it is unknown whether the pathway of interest is activated or not in these samples. Therefore the activation data needs to be estimated for the tumour samples. To achieve this a simple rejection sampler is used to generate new estimates for the
missing response data.

The $x_i$’s are the metagenes that were generated earlier in this chapter. These are available for all the cell lines, as well the tumour data. This information can be used in the rejection sampler to estimate pathway activation in each tumour sample.

Firstly a distribution based on the $x_i \beta$ is generated using the `pnorm` command in R. The $x_i \beta$ for the cell lines for which activation data is available, this is the normal and oncogenic cell lines, are used to estimate the parameters of the distribution. For each $x_i \beta$ a probability value ($\phi$) is returned based on a normal distribution with the assumption that the mean is 0 and standard deviation is 1. A random number ($\psi$) is then drawn from a uniform distribution, if this value exceeds the probability value then the response value is given as 0, if the it is below the probability value then the response value is 1. This can be best represented mathematically as,

$$y_i = 1 \text{ if } \psi_i > \phi_i$$

$$y_i = 0 \text{ if } \psi_i < \phi_i$$

It is important to note only the y’s for which the response data is missing (the tumour data) are generated from the rejection sampler. For the normal and oncogenic cell lines, since the y’s are known they are not replaced.

### 2.8 Simulation

A number of simulations were performed to test various aspects of the univariate model. To simulate the data, the probit data function `rbprobitGibbs` was used. This function implements the Albert and Chib Gibbs sampler (Albert and Chib, 1993) for the binary probit model. It is implemented in the `bayesm` library in R which was written and maintained by Peter Rossi. This package can be found as part of the CRAN archive (http://cran.r-project.org/web/packages/bayesm/index.html, 1997), and more detail can be found in Rossi et al. (2005). The `bayesm` package includes many important models used in marketing and micro-econometrics applications, which this thesis uses for biomedical applications.

The first diagnostic test that was done, was to check that the probabilities of activation are similar between the `rbprobitGibbs` and `rbprobitGibbs.missing` functions. `rbprobitGibbs` is the standard implementation of the univariate probit model and all
the response data is known, so it is expected that the probabilities of activation are true here. `rbprobitGibbs.missing` is an adaptation of the previous function including the Gibbs sampler as discussed in Chapter 2.7 to account for the missing response data.

Since the simulated data has completely known response data, by virtue of the way it was created, this is gives a good way to test the model as the ideal result is known, giving something which can be compared to the results of the model. A set of missing data was artificially created, when doing this it is important that not all of the data is made missing since in the actual breast cancer dataset there is some training data available, the cell line data. In the following example there is a total of 200 responses, of which 100 were missing. Comparing the the probabilities of activation between the two functions is best achieved by graphing them against each other.

![Figure 2.2: Plots comparing the estimates probability of activation using the univariate probit model, when the response is totally known (x axis) and when there are missing values (y axis). Comparing how standardising effects these estimates, plot (a) shows the raw data, plot (b) shows the standardised data.](image)

It is clear from Figure 2.2 that when the data is not standardised to have a mean of zero and a standard deviation of one, the probit model does not give coverage of all possible probabilities of activation. In the above example the original data only returns probabilities of activation between 0.5 and 1. Using the same data but normalising it first, the probabilities of activation now extend from 0 to 1. Another important feature is that it is close to a straight line relationship, indicating that when half of the response data is missing, the Gibbs sampler generating the response data is doing a good job. The next step was to see how the number of missing responses affects
the relationship between the missing response, and full response models simulations predicted probabilities of activation.

![Comparison of activation probabilities between known and missing responses](image)

(a) 190 Missing Responses  (b) 150 Missing Responses  (c) 50 Missing Responses

Figure 2.3: Plots comparing the estimates probability of activation using the univariate probit model, when the response is totally known (x axis) and when there are missing values (y axis). Comparing how a larger quantity of missing values effects the estimates.

Logically one would assume that the more data that needs to be simulated, the more likely a disagreement will be seen. Figure 2.3 certainly backs up this hypothesis. The more missing response data there was, the more the relationship takes the shape of a standard normal CDF, which makes sense as a probit model is used to generate the data. This indicates that when there is not much data available the model takes responses that have a low probability of activation (less than 0.3) and makes them 0, and responses that have high probability for activation (more than 0.7) and makes them 1. This is a relevant issue in the breast cancer data that is analysed in this thesis, as the majority of the data is from patients tumours, where the response is unknown.

### 2.9 Python Implementation

Although much of the work in this thesis was done in R, the univariate model was also implemented in Python, in an attempt to explore the utility of that language for MCMC analysis of microarray data. There were several reasons for this, the first reason is that Python is a popular scripting language used in bioinformatics, so writing an implementation within this language would make it accessible to many bioinformaticians, and more compatible with Python-based workflows. The second reason was
that one of the original supervisors of this work (Dr Chris Fonnesbeck) was heavily involved with the development of the PyMC package. The PyMC package within Python provides a simple approach for running MCMC models, so was readily suited to our needs. A third reason was the current implementation of Bayesian Factor Regression is in MATLAB which is a commercial package, whereas Python is open source.

2.10 Introduction to python

Python\(^3\) is a multi-paradigm programming language conceived and developed in the 1980’s by Guido van Rossum, while working at the National Research Institute for Mathematics and Computer Science in the Netherlands (Venners, 2003). It permits programmers to use a mix of object-oriented and structured programming. It has become very popular among developers for its clean syntax.

When the Python interpreter (iPython (Pérez and Granger, 2007) was used in this thesis) is closed, all definitions are lost, so it is recommended the user uses a text editor to prepare the input as a script. This way definitions can be saved into a file, and used in a script or iteratively in the interpreter, this is called a module and it usually has the suffix .py. In Python a collection of modules is known as a package, and similar to R, Python is open source software and comes with an extensive library of these packages, which perform many common tasks. There are also many additional packages available that extend Python’s capabilities. “Py” is associated with many of these packages in their naming (e.g. SciPy, numPy, PyMC).

Some of the basic Python commands used in the implementation of Python in this thesis are given here and in Chapter 4.10. For a more detailed description on using python the reader is directed to http://docs.python.org/tutorial/.

To perform analysis similar to that done by Bild et al. (2006) in Python, the following packages needed to be used: numPy, Pylab, and PyMC. NumPy is the fundamental scientific programming package, as it provides a multidimensional array type and many useful functions for numerical analysis. Pylab gives plotting features similar to those used in MATLAB.

PyMC is of the most interest, and was part of the motivation to undergo this

---

\(^3\)The name Python is derived from the British Comedic Troupe Monty Python, who were active from 1963-1983
masters thesis. PyMC is a Python package that implements Bayesian statistical models algorithms including Markov chain Monte Carlo. Chris Fonnesbeck began developing PyMC in 2003, with the purpose of generalising the process of building Metropolis-Hastings samplers, and making Markov Chain Monte Carlo (MCMC) more accessible to the non-statistician. In 2005 version 1.0 was released to the public and through regular use by a small group based in the University of Georgia, it was refined. In 2006 David Huard and Anad Patil joined Fonnesbeck on the development team to produce version 2.0, with the aim of increasing flexibility and improving performance to provide a better end-user experience from that of the previous version (Patil et al., 2008).

2.11 Creating metagenes

The same analysis described in Chapter 2.14 will be discussed in this section for the Python implementation, including some of the more important syntax. The first issue to deal with when using Python, was the reformatting of the data into a way that can be feed into Python. It appears Python does not recognise column or row titles on matrices, rather it treats them as extra columns or rows respectively, this makes for some messy and nonparsimonious steps being used in this implementation.

The easiest way to use the datasets from the breast cancer study in Python is to save them as .csv files 4, which is straightforward to achieve using the write.csv function in R. To deal with .csv file in python, the “csv” module can be used; it contains functions to read (csv.reader) and write (csv.writer) .csv files. The syntax to read in the data required is given below

```python
files =
csv.reader(open('files.csv'),delimiter=',',quoting=csv.QUOTE_NONNUMERIC)
```

It is intuitive syntax: the first entry in the brackets, opens the appropriate file, the second entry informs Python to look for commas as the delimiter, similar to the “sep” command in R’s read.table. In both R and Python it is possible to define which character is used to make quotations, using the quotechar command. The quoting section of csv.reader tells python to only put quotes around the entries which are non numeric. Without this command the numeric entries will get quoted as this is the default python setting that treats all variables as characters. For a list of other

---

4The comma-separated values (.csv) file format is a set of file formats used to store tabular data in which numbers and text are stored in plain textual form that can be read in a text editor
quoting methods see http://docs.python.org/tutorial/ (2010) The above command only puts the file into Python as an object which cannot be manipulated easily, so another command is necessary to get the data into a state which is usable.

```python
lfiles=[]
for line in files:
    lfiles.append(line)
```

This command is fairly basic, the first line just creates an empty list, into which the information inside the python object can be placed. The second line starts a for loop that takes every line from the object that was generated from the .csv file. Inside the loop, each line from Python object, is appended to the lfiles array, creating a copy of the .csv file as a list object. Append is one of the features a list objects can use in Python. To use the list features type the list name followed by a fullstop then the feature that is desired, these include but are not limited to `append(x)` which adds the item x to the end of lists, `insert(n,x)` adds x into the nth position in Python, or `remove(x)` which removes x from the list.

This process is very simple, but was required for all the datasets that were used in this analysis. While fairly time-consuming to set up, once it is included the module only requires to be imported.

As Python treats the data as a “list of lists”, it was hard to manipulate the microarray data in their “natural” form. Since the data are formatted with genes as rows and samples as columns, the gene names are the first entry of each row in the Python array. To extract these for use a list of the names can be created using the command

```python
Nameslist = copy.copy([row[0] for row in list])
```

The `copy.copy` and the start of the above command is required, as Python manipulates the original source of data if changes are made to pieces extracted from it, so by making a copy it is possible to avoid problems when the original data is needed to be used again. This command is used frequently throughout the process. Inside the copy brackets a common Python statement is used which eliminates the need for complex loop structures, in this case a “For loop”.

The next step is to create lists of the genes that appear in the overall list, as well as the individual pathway probes. The pathway probes are a list of the genes showing the most relevance in predicting the presence of a particular pathway. This is easily achieved in Python using a list command such as the one below.
\[
\text{Commonlist} = \text{list(set(list1) & set(list2))}
\]

This command creates a common list of all the elements that exist in both list1 and list2. This command is used with each of the individual pathway probes and the list of genes present in the tumour data. Since the common names are generated from the name lists, their positions in the original list need to be found so that the relevant information can be used. To find positions of elements in a list the command \text{Matchlist} is used,

\[
\text{Matchlist} = \text{list(sort([namelist.index(i) for i in Originallist]))}
\]

This again takes advantage of Python’s ability to circumvent complicated loop structures, and the ability to get information out of objects in Python. In this case the \text{.index()} suffix is used, which returns the position in the list of any item that is requested. With the positions of the genes found, these indices can be used in the original list to only include rows that are common between the two lists of interest.

\[
\text{Newlist} = \text{copy.copy([Originallist[j] for j in Matchlist])}
\]

These techniques were also used to generate a new list for each pathway containing only the genes that are in the cell path list and the pathway probes list. The cell path list is the list of genes present in the oncogenic cells used in the study. Once this is done, there are now two lists for each of the pathways, one containing the expression data of the relevant genes in the tumour data, and the other the expression data of the relevant genes in the oncogenic cells. Since these two lists have been created separately the order of the genes does not match up. Therefore these lists need to be reordered so that they match up with each other. It was decided to use the cell path list as the base for selecting the order. This can easily be achieved using a combination of the \text{Nameslist}, \text{Matchlist}, and \text{Newlist} functions discussed previously in the creation of the path probes and cell path lists.

Both these lists need to be standardised to have a mean of zero and a standard deviation of one, this is achieved using a user built function called \text{Astand}, that can be found in the Appendix. Once the lists have been standardised, the names have to be removed from the array so that they can be used in the SVD function in Python. Python’s SVD function comes from the \text{linalg} module, which in turn is from the parent module \text{numPY}. Recall from Chapter 2.2 that the SVD is given by

\[
X = UDV^T
\]
SVD in Python is the same, except that it replaces the D with an S, which is merely a semantic change. So to pull out each of the three components, new variable names are made for each of components, for example when doing the SVD using the BCat pathway, the code would be given as

```
BCatU, BCatS, BCatV = linalg.svd(matBCat, full_matrices=False)
```

where BCatU and BCatV are orthogonal matrices containing the left and right singular values of X respectively, and BCatS is a diagonal matrix of singular values. The `full_matrices` option needs to be changed to “False” to generate the SVD output that is similar to that which is returned in R.

Transforming the tumour cell lines with the SVD outputs for each of the pathways, the metagenes created come out identical to the ones that were generated in R.

### 2.12 Python implementation of univariate model

To implement the univariate model in Python, the PyMC package was used. For the initial model in Python the logistic regression model was used instead of the probit regression that is used in other chapters of this thesis. This is because at the time this work was undertaken, there was no inbuilt probit link function in PyMC, but there was a readily available logistic model.

The logistic regression is defined as

\[
f(y) = \frac{e^y}{e^y + 1} = \frac{1}{1 + e^y}
\]

where

\[
y = \beta_0 + \beta_1 x_1 + \beta_2 x_2
\]

The logistic regression model is such that it accepts inputs from \(-\infty\) to \(+\infty\), but restricts output between 0 and 1: this suits this dataset as the output of interest is the probability of pathway activation.

For the breast cancer data, \(\beta_1\) and \(\beta_2\) are the first and second respective metagenes generated for each of the pathways. To use the PyMC module the data first needs to be uploaded. Only the \(y\) values that are known are uploaded. This number varies from pathway to pathway. For each of the pathways there are ten normal cell lines which are known to have an activation level of zero, and then the number of arrays for each pathway where the activation is known to be one. As a reminder the information from
Chapter 2.5 is repeated here in Table 4.1, with the specific number of arrays available listed.

Table 2.3: Number of Arrays available for Each Oncogenic Pathway

<table>
<thead>
<tr>
<th>Oncogenic Pathway</th>
<th>Number of arrays</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bcat</td>
<td>9</td>
</tr>
<tr>
<td>Myc</td>
<td>10</td>
</tr>
<tr>
<td>Ras</td>
<td>10</td>
</tr>
<tr>
<td>Src</td>
<td>7</td>
</tr>
<tr>
<td>E2F3</td>
<td>9</td>
</tr>
</tbody>
</table>

The x values are the metagenes that were generated in Python, with two metagenes per pathway. After the data have been loaded, priors on the $\beta$’s need to be defined. This is done in PyMC via the syntax

bn = Normal('bn', 0, 0.01, value=0.0)

This defines that ‘bn’, where ‘bn’ is the constant $\beta_0$, comes from a normal distribution with a mean of zero and a standard deviation of 0.01, with an initial value of zero. This is what is known as an non-informative or vague prior, as it contains little information about a prior belief of what the values of $\beta$ may be, this allows the data to speak for itself (Gelman et al., 2004).

Next the model needs to be defined. In PyMC there are two types of functions available, stochastic or deterministic. A stochastic function is one defined by a distribution where the outcome is a random draw from that distribution, whereas a deterministic function is a function where input is provided, and the output is completely “determined” by this. The models that is used to predict the activation probability are deterministic. This is defined in Python by

@deterministic

To deal with the missing data in the model, the model is defined twice, once for the data that is observed and once for the data that is unobserved. By defining a variable obs_index which is the index of the last point where data was observed, this is a simple process. The observed values function is defined as
@deterministic
def p.obs(bn=bn, bo=bo, bt=bt, xone[:obs_index], xtwo[:obs_index]):
    return invlogit(bn + bo*xone + bt*xtwo)

which restricts the xone, and xtwo inputs to be only up to point of where the data was last observed. The missing value function is defined as

def p.pred(bn=bn, bo=bo, bt=bt, xone[obs_index:], xtwo[obs_index:]):

which only uses the xone and xtwo data from the point where the data was last observed. The syntax difference between the two models is just the position of the colon. A colon before the value tells Python to use values up to that point from the beginning. If the colon is after the variable it uses information from this point till the end of the data.

The third line is the same in both observed and missing data functions and tells Python to return the value of the defined model from an invlogit, which converts the output into a value between zero and one.

Because data are missing from the y’s, these also need to be included in the model. They are defined as

    y.obs = Bernoulli('y.obs', p=p.obs, value=y, observed=True)

for the observed y’s and

    y.pred = Bernoulli('y.pred', p=p.pred)

for the y’s that are to be predicted. A Bernoulli distribution returns value 1 with a probability $p$ and value 0 with a probability $q = 1 - p$. This distribution is ideal for this situation where the probability of success $p$ comes from $p.obs$ and $p.pred$ respectively, and the values of y can only take values 0 or 1.

Once the model has been saved with an appropriate name, (e.g. “Model” below), it needs to be run using the pymc syntax.

    import Model
    from pymc import MCMC
    M = MCMC(Model)

The first line imports the model of interest into Python, the second line imports the MCMC function from the pymc module, and the third line defines the variable M as the MCMC of the model. The next line of code is
\texttt{M.isample(iter=10000, burn=1000, thin=10)}

This tells Python to perform a MCMC sample on the model, and defines the number of iterations to use, the size of the burn-in and the thinning parameter. Once this is complete, the information of interest \texttt{p.pred}, can extracted and stored in .csv format for future use. This is achieved with the syntax

\begin{verbatim}
import csv
pred = csv.writer(open('pred.csv','w'),delimiter = ',')
pred.writerow(M.p_pred)
\end{verbatim}

Note that once the csv file is created in the second line, then each row of the csv file is written in individually in the third row. The values \texttt{pred} are the predicted probabilities of activation of the pathway within in each of the tumours. This univariate model process was repeated for each of the five pathways to get the probabilities of activation for all the pathways in each tumour.

2.13 Gradient graphs

Combining the information about the metagenes and the probabilities of activation, gradient graphs can be produced. The metagenes are used to plot the graph to show the distinction between the normal cells, the tumours and the pathway activated cells. The probabilities of activation can be used to colour these graphs on a sliding scale from zero probability of activation to a probability of activation of one. This allows a check on the metagenes and the probabilities of activation to see that they are correctly defining the location of the tumours. If they are both accurate, it is expected that the tumours with low level of activation to have a colour more closely associated with low activation (normal cells) and located near them in the graph. The opposite should also be true. In the graphs low activation is coloured as blue and the colours transition from blue to red, as the activation probabilities increase.
Figure 2.3: Scatterplots of normal, pathway, and tumour cell lines. These graphs have been colour coded on a sliding spectrum from the lowest values (blue) to the highest values (red) of probability of activation.

From Figure 2.3 it is clear that there is a clear horizontal transition in the colours in all of the pathways, this is expected as the horizontal axis is that of the primary metagene. Of note is that in the BCat and E2F3 pathways there appears to be an angle on the colour transition indicating perhaps some extra information is being provided from the secondary metagene, which is graphed on the horizontal axis. This was the last work completed using the python package, before the departure of Chris Fonnesbeck.
2.14 Summary

This chapter introduced Bayesian factor regression modelling. Since microarray data provides far more samples than parameters, some form of dimensionality reduction is required. Bayesian factor regression modelling combines dimensionality reduction through SVD, and class prediction through Bayesian regression. In this chapter the Bayesian regression model that was used is the univariate probit one, and this is given in detail. The dataset that was used for this thesis, BC2116 was introduced, including details about the training data, normal and oncogenic cell line data, and the data which was analysed, the tumours collated from various international studies. The generation of metagenes for the univariate probit regression model was given, and the differentiation between the normal and oncogenic cells was examined, which looked very good, showing clear distinction in between the two for all of the pathways. Three different probit models were introduced, and they will be examined in more detail in Chapter 4. Data were simulated to test how well the rbprobit.missing function coped with various levels of missing data. As expected the less responses that were known, the more inaccurate the prediction of pathway activation became. At this stage the idea of using Python to perform the same analysis was explored, and the same metagenes that were generated through R were able to be created within the Python framework. Using the PyMC package, it was attempted to implement the Bayesian regression model, however due to limitations of the package when working on it, the logistic link function had to be used in place of the probit link.
Chapter 3

Multivariate Models

A major goal of this thesis was to investigate whether the use of a multivariate approach would be advantageous when estimating the probability of pathway activation, as the multivariate approach allows inter-pathway correlation to be included in the model. From a biological perspective, this allows the idea of pathway coactivation to be explored.

The method used remains that of BFRM, except that instead of a univariate probit model, a multivariate probit model will be used for the Bayesian regression. This chapter starts with two methods of calculating the metagenes that include information from all pathways in each metagene, and these methods are then compared. The multivariate probit model is then introduced and the initial implementation in R is investigated. The idea of a multivariate distribution is discussed, along with some of the issues this raises with respect to the Bild et al. (2006) data. A second multivariate model that deals with these issues is then presented.

3.1 Generating the multivariate metagenes

To investigate multivariate model of pathway activation there needs to be some changes to the metagene creation to solve two problems. The first is merely a technicality in that since there are a different number of samples relating to each of the different pathways, the lengths of the various metagenes differ. This makes dealing with the metagenes together within the multivariate models more difficult. The second problem is the main driver of the need to change: in the multivariate model one of the parameters of interest is the correlation between the various oncogenic pathways. In
the original method of Bild et al. (2006) since only the normal cell lines and samples from the pathway of interest were included alongside the tumour data to create the metagenes, all the information about the other pathways was excluded from the metagene. For example, in constructing the metagene for Myc in the original method, the information regarding the expression of Myc was only available for the normal, Myc and the tumour cell lines, there is no information regarding the activation of the Myc pathway in the BCat, E2F3, Ras and Src cell lines: if the suspected correlation exists this information needs to be included so it can be analysed.

In the multivariate model, a method for creating the metagenes is required that produces metagenes of equal length, and that includes information about every pathway. Two different methods of creating metagenes that met these criteria were investigated, both of these are outlined in the next two paragraphs. As previously discussed the naming of these methods as “multivariate” is not to do with their creation, as they do not involve multivariate methods, nor is the application of these methods restricted to multivariate models, as they could equally be used with univariate models, the naming is more to differentiate them from the metagenes generated using the approach described in the univariate chapter (Chapter 2.5).

Multivariate method one, as it shall be referred to from now, involves performing the SVD on just the normal cell lines and pathway of interest arrays just as in the univariate model. However instead of just transforming the tumour data arrays with the transformation matrix, the tumour data and the other pathways that were not used in the SVD are also transformed. For example, if the metagene for BCat is being generated, the initial setup is identical to method one, where the SVD is performed on the normal and BCat cell lines. The difference appears in the second step when the transformation matrix is applied, in this method not only is it applied to the tumour cell lines it is applied to the pathways that were not used in creating the metagene, so in this case Myc, Src, E2F3, Ras. The benefit of including the other cell lines along with the tumour data, is that any correlations between expression of the different pathways will be picked up in this step.

Multivariate method two, as it shall be referred to from now, performs the SVD on the normal cell lines and all the pathways using the genes related to the pathway of interest. So SVD is applied using each of the gene lists associated with the pathways
of interest, but it is done on all the normal and pathways cell lines each time. Then
the transformation matrices created by these decompositions are used to the transform
the tumour cell lines, again getting one metagene for each pathway.

3.2 Comparison of the two multivariate metagenes

Multivariate method one was run, and the metagenes created. Graphing metagene
one and metagene two for only the normal cell lines and the pathway cell lines(Figure
3.1) shows a clear and accurate distinction has been made in the E2F3, Myc, and Ras
graphs between the pathway of interest and the other pathways and cell lines. In con-
trast, the graphs for BCat and Src exhibit poor distinction: the pathways of interest
and the other pathways appear to show quite a lot of overlap. Another point of interest
in this method is the distinction generated between the other pathways from each other
on the vertical axis, suggesting the second metagene may be required in the analysis
performed with these metagenes to take full advantage of the information provided.
Biologically this suggests that the genes for each pathway contain some information
about the activation status of other pathways, implying that co-activation may be oc-
curring.
Figure 3.1: Each individual graph shows a scatterplot of the normal and pathway cell lines for each of the five pathways with the primary metagene as the x axis and the secondary metagene as the y axis. The metagenes used were generated using the “multivariate method one.” The cell line points are colour coded with the normal being green, BCat- Pink, E2F3 - Blue, Myc - Red, Ras - Black, Src - Cyan

Multivariate method two (Figure 3.2) gives the same overall pattern with the E2F3, Myc and Ras pathways still having a clear distinction from the other pathways albeit less distinct than the ones generated via method one. The same problems were also seen in the BCat and Src pathways with there being no clear distinction between the pathways of interest and the others. As well as having less distinct differentiation between the pathway of interest and the other pathways, there is little to no differentiation between the other pathways in this method. Because of these two factors it was decided to proceed using the metagenes generated from multivariate method one.
Figure 3.2: Each individual graph shows a scatterplot of the normal and pathway cell lines for each of the five pathways with the primary metagene as the x axis and the secondary metagene as the y axis. The metagenes used were generated using the “multivariate method two”. The cell line points are color-coded with the normal being green, BCat - Pink, E2F3 - Blue, Myc - Red, Ras - Black, Src - Cyan.

It was also decided based on the evidence of both Figure 3.1 and Figure 3.2 to proceed with the rest of the investigations without BCat and Src. This left three pathways in the study: E2F3, Myc, and Ras. It is interesting to note at this point, that these are the same three pathways used by Huang et al. (2003). It is only in later papers that the introduction of the other pathways occurs.
3.3 Multivariate probit regression model

Since there is the possibility that more than one of the pathways of interest may be active in the same tumour, there is a possibility of correlation in the levels of activation across samples. Lesaffre and Molenberghs (2006) suggest that in medical statistics the multivariate probit model is best used when analysing two or more distinct but correlated binary variables, as it fully exploits such correlations. Here the activation status (on or off) of the pathways would constitute a set of possibly correlated binary variables.

The multivariate probit regression model, is an extension of the univariate probit model given in equation 5.1 and is defined as

\[
y_1 = 1 \quad \text{if } y_1^* > 0 \\
y_2 = 1 \quad \text{if } y_2^* > 0 \\
\vdots = \vdots \\
y_m = 1 \quad \text{if } y_m^* > 0
\]

where \(y_1, y_2, \ldots, y_m\) are binary dependant variables and \(y_1^*, y_2^*, \ldots, y_m^*\) are responses of a set of linear regression models that use a single set of \(p\) predictor variables \((x_1, x_2, \ldots, x_p)\).

\[
y_1^* = \beta_{11}x_1 + \ldots + \beta_{p1}x_p + \epsilon_1 \\
y_2^* = \beta_{12}x_1 + \ldots + \beta_{p2}x_p + \epsilon_2 \\
\vdots = \vdots \\
y_m^* = \beta_{1n}x_1 + \ldots + \beta_{pn}x_p + \epsilon_n
\]

In the more general form of this equation there are intercepts included in the model, but it was decided for the purpose of this study that these added no extra information to the model, as the metagenes were standardised to have a mean of zero. This form is the same as the multivariate linear regression model defined by Johnson and Wichern (1998). In matrix notation this model is the same as the single-response regression model discussed in chapter 2.3, except the matrix quantities have multivariate forms. The equation is given in equation 3.1,

\[
y = X\beta + \epsilon
\]
where $y$ is an $n \times m$ matrix, $n$ is the number of trials and $m$ is the number of response variables.

$$
\begin{bmatrix}
  y_{11} & y_{12} & \cdots & y_{1m} \\
  y_{21} & y_{22} & \cdots & y_{2m} \\
  \vdots & \vdots & \ddots & \vdots \\
  y_{n1} & y_{n2} & \cdots & y_{nm}
\end{bmatrix}
$$

$X$ is a $n \times p$ matrix, where $n$ is the number of trials and $p$ is the number of predictor variables.

$$
\begin{bmatrix}
  X_{11} & X_{12} & \cdots & X_{1p} \\
  X_{21} & X_{22} & \cdots & X_{2p} \\
  \vdots & \vdots & \ddots & \vdots \\
  X_{n1} & X_{n2} & \cdots & X_{np}
\end{bmatrix}
$$

$\beta$ and $\epsilon$ are $n \times p$ matrices also.

$$
\begin{bmatrix}
  \beta_{11} & \beta_{12} & \cdots & \beta_{1p} \\
  \beta_{21} & \beta_{22} & \cdots & \beta_{2p} \\
  \vdots & \vdots & \ddots & \vdots \\
  \beta_{n1} & \beta_{n2} & \cdots & \beta_{np}
\end{bmatrix}
\quad 
\begin{bmatrix}
  \epsilon_{11} & \epsilon_{12} & \cdots & \epsilon_{1m} \\
  \epsilon_{21} & \epsilon_{22} & \cdots & \epsilon_{2m} \\
  \vdots & \vdots & \ddots & \vdots \\
  \epsilon_{n1} & \epsilon_{n2} & \cdots & \epsilon_{nm}
\end{bmatrix}
$$

The $\epsilon$’s have $E(\epsilon_i) = 0$ and $Cov(\epsilon_i, \epsilon_k) = \sigma_{ik}I$ where $i, k = 1, 2, \ldots, m$. The relationships between the $\epsilon$’s is where the correlation between the response variables can be best exploited.

### 3.4 Allenby multivariate probit

To implement the multivariate probit model, the first step was to attempt to adapt the R rmvpgibbs function from the bayesm package (Rossi et al., 2005). The bayesm
package provides various Bayesian models for use in marketing and microeconomics, but these models can be equally used in a bioinformatics setting. The model used here was the Allenby multivariate model that is implemented in R in the \texttt{rmvpGibbs} function. The data needs to be rearranged to accommodate the inputs required by \texttt{rmvpGibbs}.

While the matrix format is the same as the traditional multivariate probit model given in equation 3.1, the Allenby model as implemented in the \texttt{rmvpGibbs} function requires the actual matrices used to be of different dimensions. All the same information is included, just the dimensions of the input are altered. The model requires a $y$ matrix with one column, so in order to use the data, the $y$'s need to be “stacked” one on top of the other, giving a $(m \times n) \times 1$ matrix, as opposed to the more traditional $m \times n$ matrix.

$$y^T = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} & y_{21} & \cdots & y_{m1} & \cdots & y_{mn} \end{bmatrix}$$

As a consequence of the rearrangement of $Y$, the matrix incarnations of $X$, $\beta$ and $\epsilon$ have to be altered accordingly. The $X$ matrix becomes $n \times np$ matrix, each row includes $p$ $x$'s that correspond to the position of the $\beta$'s that should be included in the model, with zeroes in all other entries.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & \cdots & x_{22} & x_{22} & \cdots & x_{2p} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & x_{n(p-1)} & x_{np} \end{bmatrix}$$

The $\beta$ matrix is now a $np \times 1$ column matrix

$$\beta^T = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1p} & \beta_{21} & \beta_{22} & \cdots & \beta_{2p} & \cdots & \beta_{n1} & \beta_{n2} & \cdots & \beta_{np} \end{bmatrix}$$

The $\epsilon$ matrix takes the same format as the reformatted $y$'s

$$\epsilon^T = \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \cdots & \epsilon_{1n} & \epsilon_{21} & \cdots & \epsilon_{m1} & \cdots & \epsilon_{mn} \end{bmatrix}$$

While not difficult, this is a time consuming exercise, hence the need to write a script to automate the process. Once the data has been reformatted to meet the specifications of \texttt{rmvpGibbs}, it is a simple matter of running the function, with no further adjustments required.
### 3.5 Multivariate distribution

Since the three metagenes will be used in conjunction with one another in the multivariate model it would be pertinent at this stage to compare how they related to each other. Under the Allenby multivariate model it is assumed that the metagenes come from a multivariate normal distribution. The simplest multivariate model is the bivariate case. The two variables will be referred to as $X_1$ (horizontal axis) and $X_2$ (vertical axis). If the correlation in the bivariate data is set equal to 0.5 and assuming the mean and standard deviations of both the datasets are set as 0 and 1 respectively, the contours of the distribution become quite oval (Figure 3.3).

![Contour plot of a bivariate Normal with $\mu = 0$ and $\rho = 0.5$](image)

**Figure 3.3:** Contour plot of a bivariate Normal with $\mu = 0$ and $\rho = 0.5$

If the correlation is set to 0, keeping the means and standard deviations as 0 and 1 respectively, the shape of the distribution becomes circular, as shown in Figure 3.4.
The issue is that the predictions of the activation levels in tumour data derived from a circular distribution, will be less precise than those drawn from a distribution closer to an oval shape in the case where a correlation between pathway activation status exists. In the bivariate case where \( \rho \neq 0 \) the use of \( X_2 \) to define the distribution of \( X_1 \), \( f(X_1|X_2) \), will provide more accuracy than using only the marginal distribution \( f(X_1) \). By graphing the metagenes generated by “multivariate method one” against one another, an idea of what the actual shape of the distribution they come from can be approximated.
Figure 3.5: Graphs plotting the three different pathways primary metagenes (from multivariate method one) against one another. The points are colour coded with the normal - Green, E2F3 - Blue, Myc - Red, Ras - Black, tumours - Grey. The main interest in these graphs is the correlation of the tumours between metagenes for the different pathways.

From Figure 3.5 it appears there is some correlation in the tumour data between all pairs of pathways of varying degree, it appears to be strongest in the E2F3 - Myc graph (Pearson correlation of 0.4350), gets weaker in the E2F3- Ras graph (Pearson correlation of 0.2118) and the weakest in the Myc-Ras graph (Pearson correlation of 0.0320). However, in each of the graphs there is no correlation at all between the oncogenic cell lines. This lack of correlation in the oncogenic pathways has the potential to skew the overall correlation of the metagenes towards 0.

Based on the content of Figure 3.5 it would appear that to get more accurate predictions of the activation levels in the tumour data, the correlation of the metagenes needs to be estimated with the normal and oncogenic cell lines removed from the metagenes, to return to a more oval shaped multivariate normal distribution. This inspired the creation of the alternate multivariate model, which is a mixture model where the cell lines are treated as uncorrelated normals, while the tumours are from a multivariate normal with non-zero correlation.

### 3.6 Alternate multivariate model

The use of the inbuilt R package is a valid way of predicting the probabilities of activation for the the tumour data, however the step to estimate the correlation between
parameters uses the values for both the cell lines and the tumour data, so this leads to the problems discussed in section 3.5, which estimates of the correlation very close to zero because of the influence of the uncorrelated pathway samples.

It was decided to attempt to develop a model that would give better estimates for the correlation of the tumour data, without including the normal and oncogenic cell lines to skew these estimates. The idea of the alternate model was to get around the correlation of 0 problem by estimating the $\beta$’s individually using the univariate situation, as they are unaffected by the other pathways. Then the next step was to estimate the correlation using a rejection sampler. The final step is to combine all this information into calculating the response data using the rejection sampler described in chapter 3.8.

The rejection sampler for the correlation between the pathways begins by proposing values for the correlation. These proposed correlations come from a random uniform distribution from negative one to one. Next a likelihood value is calculated for observing the $x_\beta$’s based on a multivariate normal distribution with a mean of zero, and a correlation matrix with ones down the diagonal, and the proposed correlations on the off diagonals.

The likelihoods are then log transformed (likelihoods are far more easily manipulated in this format since addition can be used instead of multiplication), with an epsilon term used to avoid taking the log of zero. A list of log likelihoods is thus generated, one for each of the proposed correlations, and from this list the maximum likelihood is identified. The ratio of each of the likelihood in the list compared to the maximum likelihood is calculated.

A draw of 100 random uniform variables from zero to one is then taken, and compared to the ratios. If at least one of the ratios is larger than the random numbers drawn then the correlation is updated to the first proposed correlation whose likelihood ratio is larger than the random number generated. If none of the ratios exceed the random numbers then the original correlation is retained.
3.7 Testing the correlation estimator

The correlation estimator that was used is designed to estimate the correlation between the probabilities of activation of the oncogenic pathways. It was built from scratch and tested to make sure it could deal with different values of correlation.

The testing that was done was to generate two lists of similar sized data sets as the ones the actual model deals with, that have known correlations. The data sets were generated using the \texttt{rmvnorm} function from the \texttt{mvtnorm} package in R which generates data from a multivariate normal distribution for given means and correlations. The means were set as 0 and various values of correlation were used to test how the model reacted to being close to either end of the proposed correlations, the values that were used were -1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, and 1. Since the data were generated using a random number generator, the values of the actual correlation sometimes differ from the what was proposed, so the titles reflect the actual correlation between the two matrices. The sampler was run 1000 times to give a distribution of the predicted correlations, the posterior mean of these values was used as the estimate of the actual correlation.
Figure 3.6: A plot of the distributions of the predicted correlations from 1000 samples. There were nine different correlations ranging from -1 to 1. A blue line indicates the posterior mean of these predictions, and a red line shows what the actual observed correlation was.

From Figure 3.6 it is apparent that near the boundaries of the distribution the estimates become quite poor, with the distributions being quite skewed. This should
however be expected due to the artificial asymptote in the proposed correlations, as values above 1 or below -1 do not exist. Values in between the boundaries appear to behave very well, with distributions looking very symmetric, and the posterior means looking very close to the actual correlation of the two matrices.

The data that is actually used in the model is not normally distributed, so it was decided to use the correlation estimator to estimate the correlations between the metagenes generated using “multivariate method one” for each pair of the three pathways (E2F3, Myc, Ras). Since the metagenes will have similar distribution to the probabilities of activation and because they are known, they were useful as test data. This is not a major issue though it can be solved by standardising the data before it is put into the estimator.

![Graphs showing predicted correlations between metagenes](image)

**Figure 3.7:** A plot of the distribution of the predicted correlations between the metagenes generated by “multivariate method 1”. The first column is between E2F3 and Myc, the second column is E2F3 and Ras, and the third between Myc and Ras. The top row shows the distributions of the predicted correlations before the metagenes were standardised, the bottom row shows the predicted correlations after the metagenes have been standardised to have mean of 0 and standard deviation of 1.
In Figure 3.7 it is apparent that before the data is standardised (top row) there appears to be a bimodal distribution in the correlations for each of the three pairs of metagenes. In all three pairs, one of the modes is the positive set of numbers and the second mode is in the negative set of numbers, they appear to be quite symmetric, giving an overall observed mean of 0. One of the reasons for this is that the correlation estimator works under the assumption that the data is distributed with a mean of 0 and standard deviation of 1, and since this is not the case, only extreme correlation values are able to maximise the likelihood.

Once the data has been standardised (bottom row) the bimodal nature of the data immediately disappears, and the estimates of correlation become a lot closer to the observed values. From these results it was decided to include a standardising function for the probabilities of activation within alternate multivariate model, to ensure good estimates of correlation.

At this time it is important to remember that when using standardised data the correlation is equal to the covariance. The equation for correlation is given as

$$\rho_{X,Y} = Corr(X,Y) = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$$ (3.2)

When using standardised data $\sigma_X$ and $\sigma_Y$ are equal to 1, making the result

$$\rho_{X,Y} = Corr(X,Y) = Cov(X,Y)$$ (3.3)

In the case of the metagene data, before the standardisation is performed, the data is very small, with means of magnitude approximately $10^{-18}$ and standard deviations of the magnitude $10^{-2}$, resulting in very small estimates of correlation as a result.

### 3.8 Multivariate data imputation

For the multivariate models a simple rejection sampler that was used for the univariate model in Chapter 2.7 does not suffice, as the information from the other pathways is not incorporated into the estimates of the new response variable. The solution to introducing the extra information is to assume the responses come from a multivariate normal distribution.

Given $n$ sets of responses, the posterior distribution of the binary response data can be given by
\[
\begin{pmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_n
\end{pmatrix}
\sim \text{MVN}
\begin{pmatrix}
\mu_1 \\
\mu_2 \\
\vdots \\
\mu_n
\end{pmatrix},
\begin{pmatrix}
\Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1n} \\
\Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma_{n1} & \Sigma_{n2} & \cdots & \Sigma_{nn}
\end{pmatrix}
\] (3.4)

where

\[
\mu_1 = X_1\beta_{11} + X_2\beta_{12}
\]
\[
\mu_2 = X_1\beta_{21} + X_2\beta_{22}
\]
\[
\vdots
\]
\[
\mu_n = X_1\beta_{n1} + X_2\beta_{n2}
\]

so that each run of the probit model generates new estimates of \(\mu\) and \(\sigma\), where \(\sigma_{ij}\) are the covariance-variance value between the \(i^{th}\) and \(j^{th}\) responses. The relationship between the \(\sigma\)’s and \(\Sigma\)’s is given by

\[
\Sigma_{11} = \sigma_{11}I_{n\times n}
\]
\[
\Sigma_{12} = \sigma_{12}I_{n\times n}
\]
\[
\vdots
\]
\[
\Sigma_{ij} = \sigma_{ij}I_{n\times n}
\] (3.6)

From these new estimates it is possible to generate \(\mu\)’s and \(\Sigma\)’s for each response in the model, conditional on all the other responses present in the model. These conditional probabilities are calculated using equation 3.7.

\[
\mu_{1|2,\ldots,n} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)\Sigma_{1|2,\ldots,n} = \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}
\] (3.7)

Using the updated conditional \(\mu\)’s and \(\Sigma\)’s, a binary response for each of the metagenes is generated from the distribution given in equation 3.4. With this new response data the multivariate probit model is run again. The above process of generating new \(\beta\)’s and \(\sigma\)’s from the response data, then generating the response data from the \(\mu\)’s and \(\Sigma\)’s is repeated until convergence is achieved.

As a side note, the way the multivariate model is implemented in \(\mathbb{R}\), it requires a complete set of response data to initiate. Since the data has missing response data, and
the rejection sampler for the multivariate model requires estimates for $\beta$ to generate the response data, a simple estimate is needed to set the model underway. Use of random binomial data can be used but this may increase the amount of time taken for the variables to converge in our Gibbs sampler, so instead “educated starting points” were used. Specifically, a mean is taken of the metagenes used in the multivariate model, then the values for each of the tumour samples is made by comparing to the mean of its corresponding metagene. If the tumour sample value is above the mean, its initial activation indicator takes the value of 1, and 0 otherwise.

### 3.9 Multivariate simulation

Both the existing `rmvpGibbs` and the adapted `rmvpGibbs.missing` can be checked to see how they agree with each other in their estimates of the probabilities of activation. Due to the multivariate model being more complex than the Univariate model, a longer run time is experienced. Also, due to the nature of the `rmvpGibbs` models, missing values can not be included in the models. It was decided to test the effect of missing values by adjusting the number of $y$’s the `rmvpGibbs.missing` function updates. The actual data is just treated as proposed values of $y$. The first test that was done was to see how standardising the data to a mean of 0 and a standard deviation affected the estimates. The `rmvpGibbs.missing` updated values 100 through to 200, replicating the effect of 100 missing values,
Figure 3.8: Plots comparing the estimates probability of activation using the multivariate probit model, when the response is totally known (x axis) and when there are missing values (y axis). (a) and (b) contain non-standardized and standardized data respectively.

Like the univariate simulation (see chapter 2.8) it is apparent that before the data are standardised the full range of probabilities from 0 to 1 is not utilised. In Figure 3.8 it is apparent that the non-standardised data only produces probabilities estimates between 0.5 and 1. The standardised data on the other hand estimates values between 0 and 1. Also of interest is the fact that it appears to be similar to a normal CDF, indicating the Gibbs sampler used to predict missing response data, is giving results pretty consistent with the reality.
Figure 3.9: Plots comparing the estimates probability of activation using the multivariate probit model, when the response is totally known (x axis) and when there are missing values (y axis). (a) has 190 missing responses, (b) has 150 missing responses, and (c) has 50 missing responses.

Figure 3.9 shows that as more and more of the values are treated as missing the worse the correlation between the model with known values and the one with missing values becomes. In plot (a) where 190 values are missing there appears to be no real relationship at all, this is somewhat concerning as the data being used in this study features a high proportion of missing values in the responses.

3.10 Summary

This chapter took the idea of Bayesian factor regression modelling and extended it to a multivariate probit regression function. To achieve this extended multivariate model a new way to generate the metagenes was required. Two of these methods were looked at and the best one was chosen based on the fact that it gave better differentiation amongst the normal and oncogenic cell lines, as well as better differentiation between the various oncogenic cell lines. During this process it was discovered the BCat and Src pathways did not perform very well when used in a multivariate framework, they were therefore excluded from further work done in this thesis. The basic multivariate probit model was introduced as well as an alternate multivariate which took into account that the metagenes did not fit the distributional assumptions of the simple multivariate model. The alternate model required the correlation to be estimated outside the step to calculate activation, this correlation estimator was tested using simulated data. Like
the univariate model, the multivariate model was tested to see how it dealt with missing response data, and as expected the less known data, the less accurate the predictions became. Both the multivariate models will be tested further in Chapter 4.
Chapter 4

Model diagnostics

In sections 2.5 and 3.1 the generation of metagenes was discussed, along with their use in, univariate and multivariate models. In this chapter those models are applied to breast cancer data to see whether or not this information is related to clinical information that is available. The first part of this chapter deals with assessing the convergence of the univariate and multivariate models, to check the validity of using them. The probabilities of activation are also compared for all three of the pathways (E2F3, Myc, Ras) across the three univariate models using the posterior means, to see how they agree. The probabilities of activation were compared between the two multivariate models alongside the simple univariate model, again using the posterior means. The same groups were used to compare the 95% credible intervals on the predictions for the probability of activation. From these results the best univariate and multivariate models were selected to be used with clinical data.

The 14 cohorts combined form a large dataset with over 2116 data points, so it was decided to use just the first cohort, GSE3494 that was used by Miller et al. (2005) to perform diagnostics on the various univariate and multivariate models and select the best representative from each group. Cohort GSE3494 was chosen since it has a representative number of tumours, and compared to some of the other cohorts it contains a relatively complete set of clinical information regarding each tumour including, but not limited to, LN status, ER status, grade, and tumour size.

4.1 Convergence of univariate models

The univariate models are summarized in Table 4.1.
Table 4.1: Univariate models

<table>
<thead>
<tr>
<th>Model No.</th>
<th>Model name</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Simple</td>
<td>$y_i^* = x_{1i} \beta_1 + \epsilon_i$</td>
</tr>
<tr>
<td>2</td>
<td>Intercept</td>
<td>$y_i^* = \beta_0 + x_{1i} \beta_1 + \epsilon_i$</td>
</tr>
<tr>
<td>3</td>
<td>Two Metagenes</td>
<td>$y_i^* = x_{1i} \beta_1 + x_{2i} \beta_2 + \epsilon_i$</td>
</tr>
</tbody>
</table>

Each of these models was run using the `rbprobitGibbs.missing` a Gibbs sampler that not only samples the $\beta$’s but also the missing $y$ values. The MCMC for the univariate models was run with a draw of 100,000, keeping every $20^{th}$ iteration.

With Bayesian inference, where the data is used to come up with posterior means to give a prediction on the parameter of interest, it is important to examine the output from the model to see that convergence has been achieved.

One of the ways that this can be examined is by visually observing the values of the parameter of interest at each iteration, or the trace of the parameter. The trace shows whether or not the model is converging and/or is mixing well (SAS Institute Inc., 2010). A good trace would show a relatively constant mean and variance throughout the iterations. A series of traces were produced for the $\beta$’s in the three univariate models (Simple, Intercept, and Two metagenes) As well as the basic trace, a mean was taken at the end of every 100 iterations, this was plotted as a red line over the trace to help get a better idea of how the overall mean was changing.

![Trace plots](image1.png)

Figure 4.1: Trace plot of the estimated values for the $\beta_1$ terms in each of the respective pathways (E2F3, Myc, Ras) for the simple univariate model. There was a draw of 100,000 and every $20^{th}$ term was kept.

In Figure 4.1 there is some evidence to show that the simple model is mixing
quite well, and has reached a relatively stable posterior distribution in all three of the pathways.

Figure 4.2: Trace plot of the estimated values for the $\beta_0$ and $\beta_1$ terms in each of the respective pathways (E2F3, Myc, Ras) for the intercept univariate model. There was a draw of 100,000 and every 20th term was kept.

In Figure 4.2 there are two $\beta$ walks for each of the pathways, one for the $\beta_0$ (intercept) and one for $\beta_1$ (slope). In this particular model it would appear there are not enough iterations, particularly for the $\beta_0$ term, where the trace appears to be moving around a lot, not settling on a posterior mean, with non-constant variance. The $\beta_1$ term is better behaved, however it is evident in both the Myc and Ras pathways there appears to be strong jumps in the mean towards the end of the trace.
Figure 4.3: Trace plot of the estimated values for the $\beta_0$ and $\beta_1$ terms in each of the respective pathways (E2F3, Myc, Ras) for the two metagene univariate model. There was a draw of 100,000 and every 20th term was kept.

Figure 4.3 again shows two $\beta$ walks for each pathway, this time $\beta_1$ is associated with the primary metagene, and $\beta_2$ with the secondary metagene in each model. For both $\beta$’s in all three of the pathways they appear to have mixed very well, with the mean and the variance remaining relatively constant across all the iterations.

4.2 Geweke diagnostic test

The Geweke diagnostic test (Geweke, 1991) is used to assess convergence in a Gibbs sampler. The Geweke test involves separating a chain of parameter values into two parts, and comparing their means. If the chain is stationary, they should be similar, Geweke (1991) recommends using the first 10% of the values and the last 50%. The
implementation of the Geweke test in R comes from the coda package (http://cran.r-project.org/web/packages/coda/index.html, 2010) for the analysis and diagnosis of Markov chain Monte Carlo (MCMC) simulations. The function `geweke.plot`, which generates a Geweke-Brooks plot (Brooks and Gelman, 1998), is a plot of the Geweke diagnostic $Z$ as an increasing number of iterations are dropped off from the beginning of the chain. The Geweke diagnostic $Z$ has standard normal sampling distribution as $n$ tends to infinity. A plot of the $Z$ values for a chain therefore should mostly lie within an interval of $(-1.96,1.96)$, and if a large number of these fall outside the interval, then it implies a lack of convergence.

Figure 4.4: Geweke diagnostic plots for the simple model $\beta_1$ (var1) terms for each of the three pathways (E2F3, Myc, Ras).

Figure 4.4 indicates that in all three of the pathways the coefficients are converging relatively well in the simple univariate model, as the majority of the points fall within the acceptable region. The Myc plot has all of the points within the acceptable range so the convergence for this pathway is very good. The E2F3 and the Ras pathways have a small number of points falling just outside the acceptable range, so there may be some convergence problems here, but since the majority are in the range, there is acceptable convergence.
For the intercept model there are real convergence problems, particularly in the Ras pathway, where the `geweke.plot` returns an error of non-convergence. For the two pathways that did produce `geweke.plot`’s, it is clear that the intercept term (var1) is not converging well at all, backing up the results garnered from the trace plots (Figure 4.2), with a lot of the points not only falling outside of the acceptable region but a long way outside this region. The β’s related to the metagenes in the E2F3 model seem to be failing to reach convergence as well, however those for the Myc model appear to be acceptable.
Figure 4.6 shows good convergence for both the $\beta$’s in the two metagene model for both the Myc and the Ras pathway. In the E2F3 pathway there appears to be some convergence problems, with a number of the points falling outside the acceptable region.

From the trace plots and the Geweke Brooks plots, it appears that overall the simple and two metagene models appear to converge well with 100,000 iterations keeping every 20th term. There does appear to be some issues with the E2F3 pathway in both models, perhaps indicating there is a problem with the data related to this particular pathway. The intercept model does not appear to converge very well for any of the pathways for the intercept term. There does appear to be some convergence in the Myc pathway for the $\beta$ related to the metagene, this may indicate some sort of stability with the data for this pathway.

### 4.3 Univariate posterior means

In this analysis while the $\beta$’s are parameter of interest, the real value is getting estimates of the probabilities of pathway activation for the tumour data, as these activation probabilities maybe associated with tumours into various histological categories. To estimate the probabilities of activation, the estimates of $\beta$ at each iteration are multiplied by the respective metagenes, these values are then put into the `pnorm` function in \texttt{R} to get a probability. A mean is then taken from all these values to give the posterior mean of probability of activation.

For the each of the three univariate models, the posterior means were calculated and these were compared to see how similar the estimates were given by each of the models.
Figure 4.7: Comparing the predicted probabilities of activation for univariate models. The columns are divided into the three different pairs of models. The first column compares the simple model (x axis) with the intercept model (y axis), the second column compares the simple model (x axis) with the two metagene model (y axis), the third column compares the intercept model (x axis) with the two metagene model (y axis). The rows indicate which pathway is being examined, the top row shows E2F3, the middle row shows Myc, and the bottom row shows Ras.

Figure 4.7 show that the simple model and the two metagene models, in the middle column, give a reasonably linear pattern for the E2F3 and Myc indicating that these models produce similar estimates of activation for these pathways. The Myc pathway is closest to linear with just a little bit of noise, then next the E2F3 pathway has a lot more noise, but still maintains a strong linear relationship. The Ras pathway appears to show no linear pattern at all, with the data spread all over the graph.

Comparing the intercept model to other two (simple and two metagene), it is quite obvious there is a strong disagreement in values of the probability of activation. For
both the Ras and E2F3 pathways there appears to be an asymptote on the tumours, in that there activation levels can not go above around 0.55. The Myc pathway appears to be unaffected in the tumours, since it has tumours at all levels of activation from 0 to 1.

For all the pathways comparing between the intercept model, there is also a strong curve in the graphs, indicating in the intercept model appears to favour a low probability of activation, until the the other two model’s probabilities get very high. Almost all of the probabilities for the intercept model lie between 0 and 0.2. It is only when the probabilities of activation in the other two models get above 0.8, that the intercept probabilities get outside this band, and only in the Myc pathway do the intercepts’ probabilities of activation get higher than 0.4. It is also worth noting that while the two metagene versus the intercept graph has the same pattern as the simple versus the intercept graph, there is more noise in the graph of the two metagene model.

From these graphs it appears the relationships between the intercept model and the other two models are not linear, so to calculate the correlations the Spearman’s rank correlation coefficient (Spearman, 1987) will be used. The Spearman’s rank correlation coefficient takes each of the values in the two datasets, and ranks them in order from lowest to highest. The correlation coefficient assesses how well the relationship between these two ranks can be described using a monotonic function, if the relationship is a perfect increasing monotonic function the coefficient is 1, a perfect decreasing monotonic function gives a coefficient of −1. The correlation coefficient can take any value between −1 and 1, if the value is 0 this means there is no relationship between the ranks at all.

Table 4.2: Spearman’s rank correlation coefficients between probability of activation

<table>
<thead>
<tr>
<th></th>
<th>Simple</th>
<th>Intercept</th>
<th>Two Metagenes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>E2F3</td>
<td>1</td>
<td>0.7673</td>
</tr>
<tr>
<td></td>
<td>Myc</td>
<td>1</td>
<td>0.9683</td>
</tr>
<tr>
<td></td>
<td>Ras</td>
<td>1</td>
<td>0.3916</td>
</tr>
<tr>
<td>Intercept</td>
<td>1</td>
<td>E2F3</td>
<td>0.7673</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Myc</td>
<td>0.9683</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Ras</td>
<td>0.3916</td>
</tr>
<tr>
<td>Two Metagenes</td>
<td>0.7673</td>
<td>0.7673</td>
<td>E2F3</td>
</tr>
<tr>
<td></td>
<td>0.9683</td>
<td>0.9693</td>
<td>Myc</td>
</tr>
<tr>
<td></td>
<td>0.3916</td>
<td>0.3916</td>
<td>Ras</td>
</tr>
</tbody>
</table>
Table 4.2 summarises the Spearman rank correlation coefficients. The correlation coefficients associated with the simple vs intercept models indicates are 1 for all three pathways, indicating that both models are ranking all of the tumours in the same order, so while the values may be different in some places, the overall trend is the same. As a consequence of both these models trending the same then both of there correlation coefficients with the two metagene model will be the same. For both the intercept and the simple models compared to the two metagene model, the correlation coefficient is highest for the Myc pathway, followed by the E2F3 pathway, and lowest correlation comes from the Ras pathway.

4.4 Credible interval widths between the three univariate models

Since the posterior means are being used to calculate the probabilities, there is a degree of uncertainty associated with the estimates. In order to assess this uncertainty 95% credible intervals can be calculated.

![Figure 4.8: Credible intervals for the probabilities of activation in the tumours for the three univariate models are shown as individual boxes within each graph (simple, intercept, two metagene), for each of the three pathways, indicated by the separate graphs (E2F3, Myc, Ras)](image)

Figure 4.8 plots the widths of the credible intervals for the pathway activation probabilities for each tumour in boxplots. This was achieved by first calculating the 95% credible intervals for every tumour at every iteration. These differences are then plotted as boxplots to give an indication of the distribution of credible interval lengths.
produced by each method for each pathway.

From these boxplots it is clear that intercept model gives very wide credible intervals, some as much as 1. When talking about a probability of pathway activation in a single sample, having a credible interval of width one is less than ideal, as this indicates the model is unable to provide information on where the true value lies. Even just considering the interquartile ranges, the width of the credible intervals for the intercept model range is on average 0.4.

The boxplots for the two metagene model shows tighter interquartile ranges, close to 0.2 on average, although note that in the Ras model the interquartile range is actually wider than the one for the intercept model. The overall width is not as wide as intercept model for the E2F3 and Myc pathways, in both these graphs there are quite few outliers at the upper level, in the Ras pathway the overall width for the two metagene model is almost 1, which as mentioned before is not an ideal situation.

The boxplots for the simple model show much tighter intervals, the interquartile ranges for the simple model average about 0.1, and the overall widths are also lot tighter, being only 0.2, and in some cases narrower than the interquartile ranges for the other models.

Looking at the means of the widths of the credible intervals, shows that for E2F3 and Myc the pattern is similar. In the E2F3 graph, the simple model has the lowest median at around 0.15, followed by the two metagene model at 0.2 and the largest median is with the intercept model at 0.65. Similarly with the Myc graph the simple model again has the lowest median at around 0.21, followed by the two metagene model at 0.25, and the intercept model having a median of 0.68. The Myc graph differs from the other two, since in this graph the simple model has a very similar median of around 0.15 compared to the other two pathways, unlike the other two pathways though the intercept and two metagene models have similar medians of around 0.5.

Based upon the results in sections 4.1, 4.3 and 4.4, the simple model converges easily and quickly, mixing early. It gives good coverage of probabilities of activation, going from 0 to 1, the two metagene model gives a similar coverage. However the simple metagene model gives us the tightest 95% credible intervals for the probabilities of activation.
Taken this all into account it was decided the simple model was the most appropriate model to work with, when comparison between a univariate model and a multivariate model was performed.

4.5 Convergence of multivariate models

For the multivariate models there are two options, the original multivariate model, and the alternate multivariate model. As mentioned in section 3.1, in both these models there are now nine $\beta$’s to consider. As well as the $\beta$’s there are also the variance’s and covariances to estimate. Note though that in the alternate multivariate model, the variances do not need to be estimated since they have been scaled to 1.

Similarly to the univariate models the convergence of the parameters of interest was investigated with the multivariate models. This time 25,000 draws were made where every 5th was kept. Again as well as the basic trace, a mean was taken at the end of every 100 iterations, this was plotted as a red line over the trace to help get a better idea of how the overall mean was changing.
Figure 4.9: Trace plot of the estimated values for each of the nine $\beta$ terms for the multivariate model. There was a draw of 25,000 and every 5\textsuperscript{th} term was kept. $\beta_{11}$, $\beta_{12}$, and $\beta_{13}$ relate to the E2F3 pathway, $\beta_{21}$, $\beta_{22}$, and $\beta_{23}$ relate to the Myc pathway, $\beta_{31}$, $\beta_{32}$, and $\beta_{33}$ relate to the Ras pathway.

In Figure 4.9 it appears that all the $\beta$'s are mixing well, with relatively constant variance and constant mean across all the iterations, indicating that a large burn in is not required.
Figure 4.10: Trace plot of the estimated values for each of the nine $\beta$ terms for the alternate multivariate model. There was a draw of 25,000 and every 5$^{th}$ term was kept. $\beta_{11}$, $\beta_{12}$, and $\beta_{13}$ relate to the E2F3 pathway, $\beta_{21}$, $\beta_{22}$, and $\beta_{23}$ relate to the Myc pathway, $\beta_{31}$, $\beta_{32}$, and $\beta_{33}$ relate to the Ras pathway.

Figure 4.10 also shows the $\beta$’s are mixing well, with constant variance, and if anything more constant means than for the original multivariate model.

A note from both Figures 4.9 and 4.10 is that the $\beta$’s are largest on the diagonal of the graphs. This indicates most of the information for the activation of each pathway comes from the metagene relating to that particular pathway.
Figure 4.11: Trace plot of the estimated values for each of the nine $\sigma$ terms for the multivariate model. There was a draw of 25,000 and every 5th term was kept. $\sigma_{11}$, $\sigma_{12}$, and $\sigma_{13}$ relate to the E2F3 pathway, $\sigma_{21}$, $\sigma_{22}$, and $\sigma_{23}$ relate to the Myc pathway, $\sigma_{31}$, $\sigma_{32}$, and $\sigma_{33}$ relate to the Ras pathway.

Figure 4.11 shows good mixing and convergence for all the variances and covariances in the original multivariate model. The upper triangular region is really all that is of interest due to the symmetry of the estimates.
Figure 4.12: Trace plot of the estimated values for the three of the \( \sigma \) terms in the alternate multivariate model. There was a draw of 25,000 and every 5\(^{th}\) term was kept. Only \( \sigma_{11} \) showing the correlation between E2F3 and Myc, \( \sigma_{22} \) showing the correlation between E2F3 and Ras, and \( \sigma_{33} \) showing the correlation between Myc and Ras will be shown, as under the alternate model the other \( \sigma \)'s are set to be 1.

Figure 4.12 shows the mixing and convergence of the correlations in the alternate multivariate model. With the mean and variance remaining relatively constant across all of the iterations. Note that only the three covariance traces were recorded for the alternative multivariate model, due to the symmetric nature of the covariances, and the fact the variances were assumed to be 1 for this model.
4.6 Multivariate Geweke

Figure 4.13: Geweke diagnostic plots for the nine multivariate model $\beta$ terms.

Figure 4.13 shows there is good convergence in all of the parameters in the multivariate model. The only plot where more than one of the points fall outside the acceptable region was for variable 5, which relates to $\beta_{22}$, (the coefficient related to the metagene for Myc when used to measure the activation of the Myc pathway) however there are only a few points outside so this does not demonstrate a severe lack of convergence.
Figure 4.14: Geweke diagnostic plots for the nine alternate multivariate model $\beta$ terms

Figure 4.14 shows good convergence for all of the parameters in the alternative multivariate model. Like the multivariate model, there is only only plot with more than one point outside the acceptable region, this time variable 1, which relates to $\beta_{11}$, (the coefficient relating to the metagene for E2F3 when predicting the activation of the E2F3 pathway) this again is not enough to say there is poor convergence.
Figure 4.15: Geweke diagnostic plots for the nine multivariate model $\sigma$ terms

In Figure 4.15 the covariance and variance for the multivariate model can be seen to be converging very well, with no real areas of concern, only the odd point outside the acceptable region. Figure 4.16 shows the covariance has also converged well for the alternative multivariate model.
Figure 4.16: Geweke diagnostic plots for the three alternate multivariate model $\sigma$ terms which show correlation between the different pathways. (i.e. $\sigma_{11}$ for E2F3 and Myc, $\sigma_{22}$ for E2F3 and Ras, and $\sigma_{33}$ for Myc and Ras)

The trace plots and Geweke Brooks plots for both the multivariate models indicate strong convergence in both models for all parameters. This is in contrast to the univariate plots that exhibited some convergence problems, suggesting that perhaps the inclusion of the other metagenes in the prediction of the pathways is helping with convergence issues. Another point of note is the multivariate model required only 25,000 iterations keeping every 5th draw to reach a good convergence, whereas the univariate models required 100,000 iterations keeping every 20th draw to get close to convergence, and even then there were still some issues. This indicates that in this case the multivariate models were a lot more stable.

4.7 Multivariate compared to best univariate probabilities of activation

Similarly to the univariate model, the probability of activation was used to assess how similar the results the multivariate models produced were. The two multivariate models probabilities of activation were also compared to the that of the simple univariate
model, to determine whether the results were similar.

Figure 4.17: Comparing the predicted probabilities of activation for multivariate models and the simple univariate model. The columns are divided into the three different pairs of models. The first column compares the simple univariate model (x axis) with the multivariate model (y axis), the second column compares the simple univariate model (x axis) with the alternate multivariate model (y axis), the third column compares the multivariate model (x axis) with the alternate multivariate model (y axis). The rows indicate which pathway is being examined, the top row shows E2F3, the middle row shows Myc, and the bottom row shows Ras.

Figure 4.17 shows that both multivariate models give a good spread of probabilities of activation for the tumours. The probabilities of activation for the tumours ranged from 0 to 1. It also shows there was quite good agreement between the univariate model and both the multivariate models, with a linear pattern between all three.

Two points of interest in figure 4.17 are the graph showing the relationship between
between the two multivariate and the univariate models for the Myc and Ras pathways. In the Myc pathways, there appears to be some flattening of the relationship between the univariate and both the multivariate models in the middle probabilities of activation, \((0.2 - 0.8)\).

In the Ras pathway graphs it appears the relationship between the univariate and the original multivariate model is a lot more linear than the one between the univariate and alternative multivariate models. The only real difference between the two multivariate models is that the original multivariate model treats the correlations between the pathways as virtually 0, this gives further evidence to suggest that there is low correlation between the Ras pathway and the others.

Table 4.3: Correlation between multivariate probability of activation

<table>
<thead>
<tr>
<th>Univariate</th>
<th>Multivariate</th>
<th>Alternate Multivariate</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2F3</td>
<td>0.9637</td>
<td>0.9877</td>
</tr>
<tr>
<td>Myc</td>
<td>0.9616</td>
<td>0.9674</td>
</tr>
<tr>
<td>Ras</td>
<td>0.9949</td>
<td>0.9805</td>
</tr>
<tr>
<td>Multivariate</td>
<td>0.9637</td>
<td>E2F3</td>
</tr>
<tr>
<td></td>
<td>0.9616</td>
<td>Myc</td>
</tr>
<tr>
<td></td>
<td>0.9949</td>
<td>Ras</td>
</tr>
<tr>
<td>Alternate Multivariate</td>
<td>0.9877</td>
<td>0.9406</td>
</tr>
<tr>
<td></td>
<td>0.9674</td>
<td>0.9982</td>
</tr>
<tr>
<td></td>
<td>0.9805</td>
<td>0.9681</td>
</tr>
</tbody>
</table>

Table 4.4 backs up the conclusions from Figure 4.17, it also shows that all the correlations are above 0.94 showing that all three of the models give similar probabilities of activation, meaning that probability of activation will not deviate too much regardless of the model that is selected.

### 4.8 Multivariate credible intervals

Similarly to the univariate models, the credible interval widths on the prediction for the probabilities of activation were explored for the multivariate models as well. The simple univariate model was also included in the comparison, to see how the multivariate credible interval widths compared.
Figure 4.18: Credible intervals for the probabilities of activation in the tumours for the two multivariate models and the simple univariate model are shown as individual boxes within each graph (simple univariate, multivariate, alternate multivariate), for each of the three pathways, indicated by the separate graphs (E2F3, Myc, Ras).

Figure 4.18 compare the widths of the credible intervals for the three models across the three different pathways. In the all three of the pathways the original multivariate model has the highest median for the width of the credible intervals. The other two models, the simple univariate and the alternate multivariate model, have very similar medians. In the E2F3 and Ras pathways the simple univariate model has the lowest median, but only just marginally. In the Myc pathway the alternate multivariate model has the lowest median.

In terms of overall width of the boxes there is a clear hierarchy in all three of the pathways. The simple univariate model gives the narrowest credible intervals, followed by the alternative multivariate model, and the original multivariate model gives the widest credible intervals.

Based on credible intervals, it is clear that the alternative multivariate model is the best of the multivariate models, but that the simple univariate model still performs best overall.

4.9 Multivariate correlation estimates

In the multivariate models the correlations between the three pathways were also estimated. Although not an accurate estimate it was felt that perhaps the correlations
between the actual metagenes, may give an indication of how the probabilities of activation should be related as well, so these were also calculated.

Table 4.4: Correlation between multivariate probability of activation

<table>
<thead>
<tr>
<th></th>
<th>E2F3</th>
<th>Myc</th>
<th>Ras</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2F3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metagenes</td>
<td>0.4350</td>
<td>0.03199</td>
<td>0.0127</td>
</tr>
<tr>
<td>Multivariate</td>
<td>0.0145</td>
<td>0.0133</td>
<td>0.0644</td>
</tr>
<tr>
<td>Alternative Multivariate</td>
<td>0.3213</td>
<td>0.0644</td>
<td>0.0644</td>
</tr>
<tr>
<td>Myc</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E2F3</td>
<td>0.4350</td>
<td>0.03199</td>
<td>0.0127</td>
</tr>
<tr>
<td>Multivariate</td>
<td>0.0145</td>
<td>0.0133</td>
<td>0.0644</td>
</tr>
<tr>
<td>Alternative Multivariate</td>
<td>0.3213</td>
<td>0.0644</td>
<td>0.0644</td>
</tr>
<tr>
<td>Ras</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E2F3</td>
<td>0.2118</td>
<td>0.03199</td>
<td>0.0127</td>
</tr>
<tr>
<td>Myc</td>
<td>0.0127</td>
<td>0.0133</td>
<td>0.0644</td>
</tr>
<tr>
<td>Ras</td>
<td>0.4762</td>
<td>0.0644</td>
<td>0.0644</td>
</tr>
</tbody>
</table>

Table 4.4 show that all three of the correlation estimates from the multivariate model were relatively low, when compared to the other correlations from the metagenes and the alternate multivariate model. While this does not necessarily mean the estimates from the multivariate model are wrong, given the evidence provided by both the other correlation estimates and additional exploratory work, it seems likely that the multivariate model is under-estimating the correlation.

Looking at only the alternate multivariate model and the metagenes the correlation between Myc and Ras both these methods show that these two pathways have the lowest correlations, around 0.05. The other two pairs are a little more confusing, the correlation between E2F3 and Ras in the alternate multivariate model has the highest correlation of any of the pairs at around 0.48, followed by the correlation between E2F3 and Myc (0.32). In the metagene correlations this is reversed with the correlation between E2F3 and Myc the highest (0.44), and the correlation between E2F3 and Ras being the second highest (0.21).

So, while the alternate multivariate model gives slightly differing correlation estimates to that of the predictions from the metagenes, it give the correlations that are more in line of what is expected under the multivariate normal distribution, from which the tumours came. The original multivariate model gives correlations a lot lower than
observed in the metagene, however this is not too unexpected since in the R function the correlations are calculated with the normal and oncogenic cell line data still in the metagenes, which creates the distorted multivariate normal discussed in Chapter 3.5.

4.10 Summary

The convergence of the three univariate models was assessed using traceplots and Geweke diagnostic plots, with the models all run 100,000 times and every 20\textsuperscript{th} draw kept. Using these tools it was discovered that in the intercept model there was lack of convergence, in both the $\beta_0$ (intercept) and $\beta_1$ terms. In the other two models (simple, and two metagene models) convergence was reached for all the parameters. The next step was to examine the means and 95\% credible intervals for the probabilities of activation, from the box plots it was clear that the intercept model gave the widest confidence intervals, and that the simple model gave the tightest intervals. Because of the quick convergence and the tighter confidence intervals it was decided that the simple model was the best univariate model.

For both of the multivariate models (Multivariate and Alternate Multivariate) there were no convergence issues. Another point to note was convergence in these two models was reached a lot faster than the univariate models, only requiring to be run 20,000 times keeping every 5\textsuperscript{th} draw. Both multivariate models estimated the correlation between the metagenes. Based on other exploratory work it was decided that the estimates being given by the multivariate model were underestimates of the actual correlations, where the alternative multivariate model gave better estimates of these correlations. Because of this it was decided to use the alternate multivariate model as the multivariate model.

It was also noted that the medians and interquartile ranges of the probabilities of activation between the simple univariate and alternate multivariate models were very similar, indicating that there may not be any advantage between using either model over the other when comparing probabilities of activation with clinical information, apart from the fact that the multivariate models converge quicker.
Chapter 5

Pathway activation and clinical information

One of the main motivations of this thesis was not the creation of the univariate and multivariate models, but rather to investigate how these models differed in their ability to give information about various clinical features related to breast cancer, (see Chapter 1.2 for more information). From the results in Chapter 4 the best univariate and multivariate models were selected to be used with clinical data. Both of these models were used to see if there was an association between the probabilities of activation for the pathways and various clinical variables: lymph node (LN) status, estrogen receptor (ER) status, and progesterone receptor (PR) status. To see how the two different models compare in their predictions, medians and 95% credible intervals were used as comparison. The full BC2116 dataset was used for the comparison between pathway activation and clinical information. Then, using only the multivariate model, comparison between using the values at each MCMC iteration for each tumour, compared to just using the posterior median of these values was done using ER status to see how the inclusion of the intra-tumour variation effected the predictions.

5.1 Tools

To examine the the association between pathway activation and clinical information, boxplots and Bayesian probit models were used. For each of the different clinical variables, boxplots were created comparing the probability of activation between the various levels for the clinical variables (two levels for lymph node, estrogen receptor,
and progesterone receptor status) for each of the three pathways (E2F3, Myc, Ras). These boxplots allow us to compare the median levels of activation to see if there are any major differences in the probability of activation of the different pathways between the various clinical variables’ levels. The 95% credible intervals show how much variability in the activation probabilities exist within each level of the clinical variable. The boxplots are also annotated with red dots to indicate the location of the posterior mean probability of activation.

The second tool used was a univariate probit model for each of the pathways, for the lymph node, estrogen, and progesterone status. As there are only two levels for each clinical variable (either positive or negative) the simple binomial case of this model can used, which takes the form

\[ y_i = 1 \text{ if } y_i^* > 0 \]

where \( y_i^* \) is a latent quantity defined as

\[ y_i^* = x_i \beta + \epsilon_i \]

In this case the dependant variable \( y_i \) is the level of the clinical variable, and predictor variable \( x_i \) is the median posterior probability of activation (as calculated in Chapter 4) of the relevant pathway. This is implemented in R using the \texttt{rbprobitGibbs} function which is part of the \texttt{bayesm} package.

This model was run 25000 times and every 5\textsuperscript{th} draw was kept, giving 5000 estimates for the coefficient (\( \beta_1 \)). Taking the posterior median of these values gives an estimate of the direction of the relationship between the the probability of activation and the clinical variable. That is indicating whether increased probability of activation of a pathway mean the lymph node status is more or less likely to be positive. The 95% credible interval of these estimates of \( \beta_1 \) give us an idea whether this relationship is significant or not, if the credible interval includes 0, then this means there is not a significant relationship between the probability of activation and the clinical variable, if 0 is excluded then there is a significant relationship.

Each of the tumours have different levels of clinical information available, so for the probit models only the tumours that have the relevant clinical information were included. For example, for lymph node status only the tumours for which lymph node status was available were used.
5.2 Lymph node status

The first clinical variable that was observed was lymph node status, this is a binary response variable with a tumour either being LN+ or LN-. In the BC2116 dataset the information for lymph node status was available for 1983 tumours.

Figure 5.1: Lymph node status and pathway activation probabilities generated by the multivariate model

Figure 5.2: Lymph node status and pathway activation probabilities generated by the univariate model

Looking at the boxplots the mean levels of activation for both the all the pathways appear to be quite similar for both LN+ and LN- for both the multivariate and univariate models. There is very little evidence to show that increased presence of any of the pathways could help distinguish between the LN+ and LN- patients. The 95%
credible intervals appear to be quite similar for the Ras and E2F3 pathways, while the Myc pathway has a much wider credible interval for both the groups, suggesting large variability within the groups.

Table 5.1: Coefficients for the univariate probit model with lymph node status as the dependant variable, and the respective pathways as the predictor variable.

<table>
<thead>
<tr>
<th>Multivariate</th>
<th>Mean(95% Credible Interval)</th>
<th>Credible Interval Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2F3</td>
<td>0.2201 (−0.9996, 0.5387)</td>
<td>0.6387</td>
</tr>
<tr>
<td>Myc</td>
<td>0.1032 (−0.1430, 0.3436)</td>
<td>0.4867</td>
</tr>
<tr>
<td>Ras</td>
<td>−0.1032 (−0.4194, 0.2069)</td>
<td>0.6262</td>
</tr>
<tr>
<td>Univariate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E2F3</td>
<td>0.2473 (−0.0951, 0.5855)</td>
<td>0.6806</td>
</tr>
<tr>
<td>Myc</td>
<td>0.0914 (−0.1355, 0.3138)</td>
<td>0.4493</td>
</tr>
<tr>
<td>Ras</td>
<td>−0.1677 (−0.4917, −0.1619)</td>
<td>0.6535</td>
</tr>
</tbody>
</table>

The results from the univariate probit model back up what was seen in the boxplots, with all of the 95% credible intervals, except the univariate model for the Ras pathway, including 0 indicating there is no evidence to suggest there is a significant association between pathway activation and LN node status. Since the univariate model for the Ras pathway does not include 0 and the interval is wholly negative, this suggests that an increase in the probability of activation of the Ras pathway is more common in patients whose lymph node status is LN-, however given the overlap of the boxplot in Figure 5.2, it seems unlikely that this difference will be clinically relevant.

### 5.3 Estrogen receptor status

The next clinical variable that will be looked at is the Estrogen receptor status. Like lymph node status this is a binary response variable with the tumour either being ER+ or ER-. In the dataset estrogen receptor status is available for 1797 tumours.
Figure 5.3: Estrogen receptor status, pathway activation probabilities generated by the multivariate model

Figure 5.4: Estrogen receptor status, pathway activation probabilities generated by the univariate model

In both the univariate and multivariate boxplots the pattern appear quite similar, the credible intervals are very wide and do not give much information. The E2F3 pathway appears to have the least variation of the three pathways, having the smallest interquartile range, the Ras pathway has a slightly wider interquartile range, and the Myc pathway has the widest. If the focus is put on the means, in all three of the pathways the ER- appear to have a higher average level of activation. The difference appears to be most pronounced in the Myc pathway, followed by the Ras pathway. In the E2F3 pathway there appears to be very little difference at all between the two groups.
Table 5.2: Coefficients for the univariate probit model with estrogen receptor status as the dependant variable, and the respective pathways as the predictor variable.

<table>
<thead>
<tr>
<th>Multivariate</th>
<th>Mean (95% Credible Interval)</th>
<th>Credible Interval Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2F3</td>
<td>$-0.5926 (-0.9359, -0.2614)$</td>
<td>0.6745</td>
</tr>
<tr>
<td>Myc</td>
<td>$-0.8887 (-1.1471, -0.6357)$</td>
<td>0.5114</td>
</tr>
<tr>
<td>Ras</td>
<td>$-2.2796 (-2.6389, -1.9282)$</td>
<td>0.7107</td>
</tr>
</tbody>
</table>

Looking at the results from the univariate probit model, the results from the box-plots are backed up with E2F3 having the weakest association with ER status followed by Myc and then Ras with the strongest. Each of the values is significant as 0 is not included with in the credible interval. All of the intervals are wholly negative too, indicating that increased probability of activation for all of the pathways indicates that a tumour is more likely to be ER- than ER+. It would appear from the magnitudes of the coefficients that Ras activation is by far the strongest indicator of ER status, with the effect of Myc and E2F3 activation being similar to one another, as well as a lot lower than that for Ras. The reason that the Myc coefficient is smaller in magnitude to that of Ras, despite them both having similar differences in the means, is perhaps due to the wider interquartile range in the Myc pathway causing a reduction in the size of the coefficient.

5.4 Progesterone receptor status

The next clinical variable is progesterone receptor status. It was noted in Chapter 1.2 that this is highly correlated with ER status, so it should give similar patterns. Once again the response variable is binary either PR+ or PR-. In the BC2116 dataset the progesterone receptor status was available for only 946 of the tumours.
Figure 5.5: Progesterone receptor status, pathway activation probabilities generated by the multivariate model

Figure 5.6: Progesterone receptor status, pathway activation probabilities generated by the univariate model

As expected from the boxplots for PR status, the results have an identical pattern as the ER status in both means, medians and interquartile ranges. With all three pathways median level of activation being highest for the PR- tumours, with the difference being most pronounced in Myc, followed by Ras, with E2F3 having the smallest difference between the two levels of PR status.
Table 5.3: Coefficients for the univariate probit model with progesterone receptor status as the dependant variable, and the respective pathways as the predictor variable.

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Mean(95% credible Interval)</th>
<th>credible Interval Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2F3</td>
<td>$-0.7838 (-1.2405, -0.3267)$</td>
<td>0.9138</td>
</tr>
<tr>
<td>Myc</td>
<td>$-0.9966 (-1.3306, -0.6604)$</td>
<td>0.6702</td>
</tr>
<tr>
<td>Ras</td>
<td>$-1.4240 (-1.8767, -0.9807)$</td>
<td>0.8960</td>
</tr>
</tbody>
</table>

The rbprobit model results back up the findings in the boxplots, with the E2F3 having the lowest effect followed by the Myc pathway, and the Ras pathway having a far stronger effect than the other two pathways. Again noting that all the 95% credible intervals exclude 0, indicating there is a significant association between the activation of all the pathways and progesterone receptor status.

5.5 Intra-tumour variation

One of the issues with looking at the clinical variables is the width of the credible intervals associated with them. Not only do the tumours vary with each other but also have a lot of intra-tumour variations as the probability of activation is calculated at each iteration.

To look at this variation, one cohort was chosen, that had a manageable number of tumours and relatively complete set of data for the ER status. Cohort GSE7390 which contains 198 tumours, with only one tumour without information about ER status. The boxplots were produced for each of the 198 tumours, after they had been separated into ER- and ER+ tumours, and ordered within those groups.
Figure 5.7: (a) probability of activation of the E2F3 pathway classified by Estrogen Receptor (ER) status, for all tumours. (b) probability of activation of the E2F3 pathway for all tumours in cohort GSE7390, including the 95% credible intervals, split by classification of ER status.

Figure 5.8: (a) probability of activation of the Myc pathway classified by Estrogen Receptor (ER) status, for all tumours. (b) probability of activation of the Myc pathway for all tumours in cohort GSE7390, including the 95% credible intervals, split by classification of ER status.
From all of the boxplots, it is apparent that both ER+ and ER- have tumours with a median levels ranging from 0 to 1, with a lot of variability across their iterations. Overall however the ER- group have a higher proportion of tumours with higher median levels of activation, compared to the ER+ group. This highlights the fact that there is a lot of variability within the tumours as well as within the status groups when the predictions are made.

The features of the boxplots showing the means and 95% credible intervals for the probability of activation for the two overall groups ER+ and ER- have been discussed previously in Section 5.3. Looking at the boxplots for the individual tumours from cohort GSE7390, showing their medians and 95% credible intervals, in both the E2F3 and the Ras pathways the majority of the medians are located on or around the overall mean, indicating there is not a lot of variation between the tumours, where in the Myc pathway there are very few medians for the individual tumours laying on or around the overall mean. This indicates that there may be more variation within the tumours for the Myc pathway then the other two pathways. The second feature of note is that the widths of the 95% credible intervals are relatively narrow in the Myc pathway compared to the E2F3 and the Ras pathway suggesting more certainty in the individual tumour probability of activation.
variability was to run a glm (generalized linear model) with a probit link function with ER status as the dependent variable and the probability of activation for a particular pathway as the predictor variable, for each of the 4000 iterations of the probabilities of activation that were generated in Chapter 4. From these the mean of the coefficient associated with the pathway activation and the 95% credible interval was taken; these values were compared to the coefficients that were generated in Section 5.3 using the rbprobit model. The probabilities of activation used in this comparison come from the multivariate model. This information is summarised in Table 5.4.

Table 5.4: Table comparing the mean and 95% credible intervals of the pathway coefficients, between the glm and the rbprobit models for Estrogen receptor status.

<table>
<thead>
<tr>
<th>Pathway</th>
<th>GLM model</th>
<th>Mean (95% Credible Interval)</th>
<th>Rbprobit Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2F3</td>
<td>$-0.5420 \ (-0.8731, -0.1367)$</td>
<td>$-0.5926 \ (-0.9359, -0.2614)$</td>
<td>0.6745</td>
</tr>
<tr>
<td>Myc</td>
<td>$-0.8647 \ (-1.0011, -0.7237)$</td>
<td>$-0.8887 \ (-1.1471, -0.6357)$</td>
<td>0.5114</td>
</tr>
<tr>
<td>Ras</td>
<td>$-2.0957 \ (-2.3670, -1.7949)$</td>
<td>$-2.2796 \ (-2.6389, -1.9282)$</td>
<td>0.7107</td>
</tr>
</tbody>
</table>

Table 5.4 shows that when using the rbprobit model, the means remain similar for all the pathways, but the credible interval width reduces for the E2F3 pathway, and increases for the other two (Myc, Ras) pathways, compared to the intervals for the glm. This suggests that in the E2F3 pathway the variation within the individual tumours probability of activation is wider than the variation between the tumours with the two groups ER+ and ER-, whereas in the Myc and Ras pathways the variation is stronger between the groups than it is within the tumours.

5.6 Probability of recurrence or survival

The DFS (Disease Free Survival) event measures whether or not after a treatment the tumour has had any recurrence of the disease. The measure is binary: a tumour is scored as a 1 is the disease recurs, if the tumour was disease free (i.e, non-recurrent) at the time of their last followup, they are scored as a 0, but censored, as their disease may have recurred since their last recorded followup. The analysis comparing the
intra-tumour variation was repeated using DFS as the dependant variable in a logistic regression, again in cohort GSE7390.

Figure 5.10: (a) probability of activation of the E2F3 pathway classified by disease free survival (DFS) status, for all tumours. (b) probability of activation of the E2F3 pathway for all tumours in cohort GSE7390, split by classification of DFS status.

Figure 5.11: (a) probability of activation of the Myc pathway classified by disease free survival (DFS) status, for all tumours. (b) probability of activation of the Myc pathway for all tumours in cohort GSE7390, split by classification of DFS status.
Figure 5.12: (a) probability of activation of the Ras pathway classified by disease free survival (DFS) status, for all tumours. (b) probability of activation of the Ras pathway for all tumours in cohort GSE7390, split by classification of DFS status.

From the median levels of pathway activation, there is evidence to suggest slightly elevated levels of activation in all pathways is associated with disease recurrence. The difference between the two groups, however is not as pronounced as in the ER status graphs. Looking at the spread of the patients in cohort GSE7390, it is clear that the spread is narrowest for the Myc pathway, which has the clearest centre. There is also evidence looking at the widths of the 95% credible intervals that the intra-patient variation is the smallest in the Myc pathway when compared to the other two pathways.

Table 5.5: Table comparing the mean and 95% credible intervals of the pathway coefficients, between the glm and the rbprobit models for disease free survival.

<table>
<thead>
<tr>
<th>GLM model</th>
<th>Mean(95% Credible Interval)</th>
<th>Credible Interval Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2F3</td>
<td>0.7978 (0.6297, 0.9452)</td>
<td>0.3155</td>
</tr>
<tr>
<td>Myc</td>
<td>0.4768 (0.4049, 0.5499)</td>
<td>0.1450</td>
</tr>
<tr>
<td>Ras</td>
<td>0.5810 (0.3851, 0.7530)</td>
<td>0.3679</td>
</tr>
<tr>
<td>Rbprobit Model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E2F3</td>
<td>0.8649 (0.4568, 1.2808)</td>
<td>0.8240</td>
</tr>
<tr>
<td>Myc</td>
<td>0.4906 (0.1939, 0.7831)</td>
<td>0.5892</td>
</tr>
<tr>
<td>Ras</td>
<td>0.6378 (0.2345, 1.0216)</td>
<td>0.7871</td>
</tr>
</tbody>
</table>

From Table 5.5 the mean for the coefficients are similar in both the GLM and rbprobit models, the width of the 95% credible intervals are much larger in the rbprobit
model in all three of the pathways. This suggests that the variation within the two
groups (DFS+ and DFS-) is larger than the variation within the patients’ estimates
for probability of activation.

5.7 Summary

From sections 5.2, 5.3, 5.4 there is evidence to suggest from the box plots and the
\textit{rbprobit} models that the probability of activation of the three pathways, E2F3, Myc
and Ras are strongly associated with the estrogen receptor and the progesterone recep-
tor status of a tumour. There is no evidence however that the probabilities of activation
of the pathways are associated with the lymph node status of a patient.

In both the clinical variables where the coefficients were significant (ER and PR
status) the Ras pathway contributes the most information about which group the
tumours fall into, and E2F3 contributes the least. For E2F3, while it appears to be
statistically significant there are doubts whether it is clinically significant. For all of the
pathways an increased probability of activation was associated with the tumour being
ER- and PR-. There appears to be not a lot of difference in the widths of the 95%
credible intervals between the univariate and multivariate models suggesting neither of
the models is clearly superior to the other.

For the E2F3 model there is evidence to suggest that there is more variation within
the individual tumours’ probabilities of pathway activation, when compared to the
variation within the two ER groups. This could be one the reasons that there is a
statistical difference between the two groups, but there may not be a clinical one.
For the other pathways the variation within the individual tumours is less than that
between the groups.

For the disease free survival, it appears that the variation within the individual
tumours is less than that between the groups, those that had disease recurrence and
those that remained disease free. There is also evidence to suggest that increased
probabilities of activation in any of the pathways is associated with an increased chance
of disease recurrence.
Chapter 6

Discussion

There were four main aims of this thesis. The first was the generation of the metagenes and comparing the normal and oncogenic cells. The second was to implement a Bayesian factor regression model within Python utilising the PyMC package. The third aim was to introduce a multivariate model, to see if the activation of pathways is affected by the activation of the other pathways in the study, and whether or not knowing this relationship alters the prediction of pathway activation. The fourth aim was to see if pathway activation was associated with histological features of breast cancer, this was done by combining the pathway activation and clinical variable information (ER, PR, and Lymph node status) for the tumours in the BC2116 dataset.

Metagenes

In sections 2.5 and 3.1 various methods of generating metagenes for both the univariate and multivariate models. This work is similar to that already done by Bild et al. (2006). In Bild et al. (2006) they used three metagenes for each pathway, in this thesis the number of metagenes was restricted to two. This thesis began with five pathways Bcat, Src, Ras, Myc, and E2F3, but as will be discussed, the Bcat and Src pathways were dropped from the analysis. One of the obvious drawbacks to this method is that the only pathways included are those that have been selected, perhaps a pathway that is important in breast cancer has been ignored. One way to address this would be use unsupervised clustering methods to attempt to identify novel groups of functionally related genes (Timm, 2002).

Using two metagenes it was found that one could distinguish between the onco-
genic cells (pathways) and the controls (normal cell lines) in both the univariate and multivariate metagenes generated in this thesis. The metagenes show clear distinction between the normal and oncogenic cell lines for all the pathways of interest. However using just the two metagenes for both implementations of the multivariate metagenes, there was not a clear difference between certain pathways. It was found that the multivariate metagenes had clear ability to distinguish between the E2F3, Myc and Ras pathways, but for BCat and Src it was difficult to see any difference between the pathway cell lines. Because of this it was decided to drop BCat and Src from the analysis.

Python

The implementation of Python worked well, and gave the same results as those obtained from R. One of the drawbacks of using Python was the way it used .csv data, when brought into the Python environment the row names and column names are not imported as such, rather as an extra row and extra column in the array. This made manipulating the arrays when creating the metagenes more difficult then it needed to be, especially compared with the ease that R deals with these types of data. However this issue can be attributed to the fact that R has a more mature and robust approach to mathematical and statistical environments than Python does, and once the code has been written it no longer has to be worried about.

In terms of the actual Bayesian regression modelling, the code was a lot more simple for the Python implementation due to the fact PyMc is designed to be adaptable to many applications, where as rbprobit is a specialised function within R making adaption a little harder.

Multivariate

Developing a multivariate model was one of the main aims of the thesis, to see if adding the other metagenes into the model led to the probability of pathway activation being altered. The main issue that came up was the calculation of the correlation. Initially it was attempted to model the whole dataset on the basis that it was distributed with a multivariate normal, however during the course of this thesis it was determined that this was not appropriate, due to the presence of the cell lines in the dataset. To combat this, the correlation estimate step was included within the overall model, where
the predicted probabilities of activation for only the tumours were used to estimate the correlation and these values were feed back into the overall model. Comparing the predicted probabilities of activation between the simple univariate model and the alternative multivariate model, the medians seem very similar suggesting that introducing more information via the other pathways, did not have a large effect in this setting. There was no strong evidence to suggest there were any major gains by using a multivariate Bayesian regression model over a univariate one, apart from the fact that the multivariate model seemed to converge faster than the univariate one. In other situations, however, for example, where stronger correlations exist, the multivariate approach may be more valuable. Another area where this method could be investigated would be other studies that include metagenes generated by other approaches, for example siRNA knockdowns (Hannon and Rossi, 2004).

**Clinical information**

While it is interesting to be able to calculate the probabilities of pathway activation, of more relevance is to see whether or not knowledge of the these probabilities will aid in improving our understanding of breast cancer. For the tumour data in BC2116 there was an incomplete set of data for various clinical variables, Lymph node, ER and PR status. So when each of the clinical variables was examined, only tumours that had information about that particular clinical variable was included. For each of the three clinical variables that were looked at, they could be split into two groups, positive or negative (ER+/ER-, PR+/PR-, LN+/LN-). Through use of boxplots and a univariate Bayesian probit model, the difference in probability of activation for the three pathways between these groups could be examined. There appeared to be no difference between the probabilities of activation for any of the pathways in helping predict lymph node status. For ER and PR status there does appear to be some difference between the two groups, for all the pathways there appears to be higher probability of activation in the ER+ and PR+ compared to ER- and PR-. The difference is most pronounced for the Myc pathway, followed by the Ras pathway, in the E2F3 pathway the model indicates that there is a significant difference between the two groups however examining the boxplots it does not appear that the difference for E2F3 would be clinically significant. For DFS (disease free survival), there is evidence that an increase in pathway activation for all of the pathways, is associated with those patients in which the tumour recurred.
The intra-tumour variation was also examined. Since the probabilities of activation were calculated from a Bayesian regression model there is variation associated with these probabilities. Looking at this for both ER receptor status and DFS, it was found that the variation between the groups was larger than the intra-tumour variation for all the pathways, except for E2F3 for the ER receptor status, this was also the pathway where there appeared to be a statistical significant difference but not a clinical one, so this is a possible reason for this.

**Future Direction**

In both the multivariate models only the primary metagene from each of the pathways was utilised, if time allowed it could be worth seeing whether or not a secondary and tertiary metagene could aid in the multivariate models. While there was no evidence that the secondary metagenes aided in the univariate model, looking in the metagene graphs there is evidence that the second metagene helps distinguish between the different pathways, something of interest in the multivariate model. Another issue was that both models do not perform as well when the relative sample size of the cell lines (pathway activation status known) is very small compared to the number of tumour samples (pathway activation status unknown). This could be addressed by increasing the ratio of cell lines to tumour samples in future studies. This however may not be practical, so another method would be to use more informative priors, favouring the cell lines.

Due to the unforeseen departure of my co-supervisor the Python work was cut short. A lot of future work could be put into extending the Python implementation. The first extension would be to turn the various pieces of code that were used to get the results into a single package that could be loaded by anyone, to do similar analysis using their own datasets. Also it would be good to be able to utilise a probit link function in the univariate model instead of the logit that was used. Finally the next step would be to use the multivariate models in the Python implementation.

In terms of utilising the knowledge of probability of pathway activation and the clinical variables, one of the obvious extensions would be to combine two or more pathways in the prediction of breast cancer classification. For example instead of looking at E2F3 pathway activation between the two ER groups, it might be more useful to look at tumours where both E2F3 and Myc are activated, and see how this relates to
whether a tumour is ER+ or ER-. A similar approach was taken by Bild et al. (2006) using hierarchical clustering, however a more sophisticated model-based approach may provide better results.
References


Appendix A

R code

A.1 Generating univariate metagenes

load('celldata-setup.RData')
load('mas5-qnorm-probes-symbols.RData')

onc.path<-list()

for(kk in 1){
  dat<-mas5.pdat[[kk]]
  dn<-paste("cohort",kk,sep='')

  ps<-T
  filt<-F

  breast.trim<-dat
  if(filt){
    low.int<-apply(dat<5,1,sum)/66 > .9
    breast.trim<-dat[!low.int,]
  }

  svd.all<-svd.du<-list()
  breast.path<-list()
  breast.meta<-list()
  comb.path<-list()
  xx<-list()
  yy<-list()

  for(i in 1:length(files)){
    probes<-intersect(path.probes[[i]],rownames(breast.trim))
    xx[[i]]<-breast.trim[match(probes,rownames(breast.trim)),]
    breast.filt<-xx[[i]]
    probes<-intersect(path.probes[[i]],rownames(breast.filt))
    breast.path[[i]]<-breast.filt[match(probes,rownames(breast.filt)),]
    xx[[i]]<-t(apply(xx[[i]],1,function(x) (x-mean(x))/sd(x)))
    yy[[i]]<-t(apply(yy[[i]],1,function(x) (x-mean(x))/sd(x)))
breast.path[[i]]<-xx[[i]]

svd.all[[i]]<-svd(yy[[i]],nv=3)
svd.du[[i]]<-diag(1/svd.all[[i]]$d)%*%t(svd.all[[i]]$u)

breast.meta[[i]]<-rbind(svd.all[[i]]$v,t(svd.du[[i]]%*%xx[[i]])[,1:3])

rownames(breast.meta[[i]])<-(c(colnames(cell.path[[i]]),colnames(breast.path[[i]])))

A.2 Generating multivariate metagenes method 1

load('celldata-setup.RData')
load('mas5-qnorm-probes-symbols.RData')

onc.path<-list()

for(kk in 1){

dat<-mas5.pdat[[kk]]

dn<-paste("cohort",kk,sep='')

ps<-T
filt<-F

breast.trim<-dat

if(filt){
    low.int<-apply(dat<5,1,sum)/66 > .9
    breast.trim<-dat![low.int,]
}

n= 5
mg <- list()
zz <- list()
ord <- t(matrix(c(2:5,1,3:5,1,2,4,5,1:3,5,1:4),4,5))
just.paths<-list()
yy<-list()
xx<-list()
test<-svd.all<-svd.du<-list()
breast.path<-list()
breast.meta<-list()
comb.path<-list()
breast2.meta2<-list()
yytest<-list()

gse2 <- gse3151[,-c(0,56:64)]
colnames(gse2) <- c(rep("GFP",10),rep("Myc",10),rep("Src",7),rep("Bcat",9),
    rep("E2F3",9),rep("RAS",10))
del<- t(matrix(c(rep(0,20),21:55,rep(0,27),28:55,rep(0,36),37:55,rep(0,45),
        46:55,rep(0,55)),55,5))
if(n < 5){gse <- gse2[,-c(del[n,])]}
if(n==5){gse <- gse2}

path.probes2 <- path.probes[-6]

for(i in 1:n)
{
  probes<-intersect(path.probes2[[i]],rownames(gse))
  mgprobes<-lapply(path.probes2,function(x) x[!is.na(match(x,rownames(mas5.pdat[[kk]])))]
  xx[[i]]<-breast.trim[match(mgprobes[[i]],rownames(breast.trim))]
  yy[[i]]<-cell.path[[i]][match(mgprobes[[i]],rownames(cell.path[[i]]))]
}

mgprober <- append(mgprobes[[1]],mgprobes[[2]])
mgprober <- append(mgprober,mgprobes[[3]])
mgprober <- append(mgprober,mgprobes[[4]])
mgprober <- append(mgprober,mgprobes[[5]])

gse4 <- gse[match(mgprober,rownames(gse))]
gse5 <- t(apply(gse4,1,function(x) (x-mean(x))/sd(x)))

filter <- list(c(1:10,28:36),c(1:10,37:45),c(1:10,11:20),c(1:10,46:55),c(1:10,21:27))

for(i in 1:n)
{
  yytest[[i]]<-gse5[match(mgprobes[[i]],rownames(gse5)),filter[[i]]]
  xx[[i]]<-t(apply(xx[[i]],1,function(x) (x-mean(x))/sd(x)))
  breast.path[[i]]<-xx[[i]]
  mg[[i]]<-gse5[match(mgprobes[[i]],rownames(gse5))]
  svd.all[[i]]<-svd(yytest[[i]],nv=3)
  svd.du[[i]]<-diag(1/svd.all[[i]]$d)%*%t(svd.all[[i]]$u)
}

for(i in 1:n)
{
  test[[i]]<-t(svd.du[[i]]%*%mg[[i]])
  breast2.meta2[[i]]<-rbind(svd.all[[i]]$v,t(svd.du[[i]]%*%xx[[i]])[,1:3])
  breast.meta[[i]]<-rbind(test[[i]][,1:3],t(svd.du[[i]]%*%xx[[i]])[,1:3])
  rownames(breast.meta[[i]])<-c(colnames(gse2),colnames(breast.path[[i]]))
}

A.3 Generating multivariate metagenes method 2

load('celldata-setup.RData')
load('mas5-qnorm-probes-symbols.RData')

onc.path<-list()

for(kk in 1){
dat<-mas5.pdat[[kk]]
dn<-paste("cohort",kk,sep='')

ps<-T
filt<-F

breast.trim<-dat
if(filt){
    low.int<-apply(dat<5,1,sum)/66 > .9
    breast.trim<-dat[!low.int,]
}

n=5

mg<-list()
zz<-list()
ord<-t(matrix(c(2:5,1,3:5,1,2,4,5,1:3,5,1:4),4,5))
just.paths<-list()

yy<-list()
xx<-list()
test<-svd.all<-svd.du<-list()

breast.path<-list()
breast.meta<-list()
comb.path<-list()

gse2<-gse3151[-c(0,56:64)]
colnames(gse2)<-c(rep("GFP",10),rep("Myc",10),rep("Src",7),rep("Bcat",9),
rep("E2F3",9),rep("RAS",10))
del<-t(matrix(c(rep(0,20),21:55,rep(0,27),28:55,rep(0,36),37:55,rep(0,45),
46:55,rep(0,55)),55,5))

if(n<5){gse<-gse2[-c(del[n,])]} if(n==5){gse<-gse2}

path.probes2<-path.probes[-6]

for(i in 1:5){
    probes<-intersect(path.probes[[i]],rownames(gse))
    mgprobes<-lapply(path.probes,function(x)x[!is.na(match(x,rownames(mas5.pdat[[kk]])))]))
    xx[[i]]<-breast.trim[match(mgprobes[[i]],rownames(breast.trim)),]
    breast.filt<-xx[[i]]
    probes<-intersect(path.probes[[i]],rownames(breast.filt))
    breast.path[[i]]<-breast.filt[match(mgprobes[[i]],rownames(breast.filt))],]
    xx[[i]]<-breast.path[[i]]
    yy[[i]]<-cell.path[[i]][match(mgprobes[[i]],rownames(cell.path[[i]]),],]
    xx[[i]]<-t(apply(xx[[i]],1,function(x)(x-mean(x))/sd(x)))
    yy[[i]]<-t(apply(yy[[i]],1,function(x)(x-mean(x))/sd(x)))
    breast.path[[i]]<-xx[[i]]
    mg[[i]]<-t(apply(gse[match(mgprobes[[i]],rownames(gse))],1,function(x)(x-mean(x))/sd(x)))
svd.all[[i]]<-svd(mg[[i]],nv=3)
svd.du[[i]]<-diag(1/svd.all[[i]]$d)%*%t(svd.all[[i]]$u)
}

for(i in 1:5){

  breast.meta[[i]]<-rbind(svd.all[[i]]$v,t(svd.du[[i]]%*%xx[[i]])[,1:3])
  rownames(breast.meta[[i]])<-c(colnames(gse),colnames(breast.path[[i]]))
}

A.4 Univariate Bayesian regression model

rbprobitGibbs.mik<-function (Data, Prior, Mcmc){
  breg1 = function(root, X, y, Abetabar) {
    cov = crossprod(root, root)
    betatilde = cov %*% (crossprod(X, y) + Abetabar)
    betatilde + t(root) %*% rnorm(length(betatilde))
  }
  pandterm = function(message) {
    stop(message, call. = FALSE)
  }
  if (missing(Data)) {
    pandterm("Requires Data argument -- list of y and X")
  }
  if (is.null(Data$X)) {
    pandterm("Requires Data element X")
  }
  X = Data$X
  if (is.null(Data$y)) {
    pandterm("Requires Data element y")
  }
  y = Data$y
  nvar = ncol(X)
  nobs = length(y)
  if (length(y) != nrow(X)) {
    pandterm("y and X not of same row dim")
  }
  if (sum(unique(y) %in% c(0,1,NA)) < length(unique(y))) {
    pandterm("Invalid y, must be 0,1,NA")
  }
  if (missing(Prior)) {
    betabar = c(rep(0, nvar))
    A = 0.01 * diag(nvar)
  } else {
    if (is.null(Prior$betabar)) {
      betabar = c(rep(0, nvar))
    } else {
      betabar = Prior$betabar
    }
  }
}

117
if (is.null(Prior$A)) {
    A = 0.01 * diag(nvar)
} else {
    A = Prior$A
}

if (ncol(A) != nrow(A) || ncol(A) != nvar || nrow(A) != nvar) {
    pandterm(paste("bad dimensions for A", dim(A)))
}

if (length(betabar) != nvar) {
    pandterm(paste("betabar wrong length, length= ", length(betabar)))
}

if (missing(Mcmc)) {
    pandterm("requires Mcmc argument")
} else {
    if (is.null(Mcmc$R)) {
        pandterm("requires Mcmc element R")
    } else {
        R = Mcmc$R
    }
    if (is.null(Mcmc$keep)) {
        keep = 1
    } else {
        keep = Mcmc$keep
    }
}

cat(" ", fill = TRUE)
cat("Starting Gibbs Sampler for Binary Probit Model", fill = TRUE)
cat(" with ", length(y), " observations", fill = TRUE)
cat("Table of y Values", fill = TRUE)
print(table(y))
cat(" ", fill = TRUE)
cat("PriorParms:", fill = TRUE)
cat("betabar", fill = TRUE)
print(betabar)
cat("A", fill = TRUE)
print(A)
cat(" ", fill = TRUE)
cat("MCMC parms: ", fill = TRUE)
cat("R= ", R, " keep= ", keep, fill = TRUE)
cat(" ", fill = TRUE)
betadraw = matrix(double(floor(R/keep) * nvar), ncol = nvar)
ydraw = matrix(double(floor(R/keep) * length(y)), ncol = length(y))
pdraw = matrix(double(floor(R/keep) * length(y)), ncol = length(y))
beta = c(rep(0, nvar))
sigma = c(rep(1, nrow(X)))
root = chol(chol2inv(chol((crossprod(X, X) + A))))
Abetabar = crossprod(A, betabar)
itime = proc.time()[3]
MCMC Iteration (est time to end - min) 
fsh()
for (rep in 1:R) {

  pp = pnorm(X%*%beta)
  yy = y
  xx = 1.*(runif(length(yy))<pp)
  yy[is.na(y)] = xx[is.na(y)]
  yout = yy

  a = ifelse(yy == 0, -100, 0)
  b = ifelse(yy == 0, 0, 100)
  mu = X %*% beta
  z = rtrun(mu, sigma, a, b)
  beta = breg1(root, X, z, Abetabar)
  if (rep%%10000 == 0) {
    ctime = proc.time()[3]
    timetoend = ((ctime - itime)/rep) * (R - rep)
    cat(" MCMC Iteration (est time to end - min) ", fill = TRUE)
    fsh()
  }
  if (rep%%keep == 0) {
    mkeep = rep/keep
    betadraw[mkeep, ] = beta
    ydraw[mkeep,]= t(yout)
    pdraw[mkeep,]=t(pp)
  }

  ctime = proc.time()[3]
  cat(" Total Time Elapsed: ", round((ctime - itime)/60, 2), 
       "\n")
  attributes(betadraw)$class = c("bayesm.mat", "mcmc")
  attributes(betadraw)$mcpar = c(1, R, keep)
  return(list(betadraw = betadraw,ydraw=ydraw,pdraw=pdraw))
}

A.5 Multivariate model

rmpGibbs.Aarontest<-function (Data, Prior, Mcmc, Imp)
{
  pandterm = function(message) {
    stop(message, call. = FALSE)
  }
  if (missing(Data)) {
    pandterm("Requires Data argument -- list of p, y, X")
  }
  if (is.null(Data$p)) {
    pandterm("Requires Data element p -- number of binary indicators")
  }
  p = Data$p
  if (is.null(Data$y)) {

119
pandterm("Requires Data element y -- values of binary indicators")
}
y = Data$y
if (is.null(Data$X)) {
  pandterm("Requires Data element X -- matrix of covariates")
}
X = Data$X
levely = as.numeric(levels(as.factor(y)))
bady = FALSE
for (i in 0:1) {
  if (levely[i + 1] != i) {
    bady = TRUE
  }
}
cat("Table of y values", fill = TRUE)
print(table(y))
if (bady) {
  pandterm("Invalid y")
}
if (length(y)%%p != 0) {
  pandterm("length of y is not a multiple of p")
}
n = length(y)/p
k = ncol(X)
if (nrow(X) != (n * p)) {
  pandterm(paste("X has ", nrow(X), ", rows; must be = p*n"))
}
if (missing(Prior)) {
  betabar = rep(0, k)
  A = 0.01 * diag(k)
  nu = p + 3
  V = nu * diag(p)
}
else {
  if (is.null(Prior$betabar)) {
    betabar = rep(0, k)
  } else {
    betabar = Prior$betabar
  }
  if (is.null(Prior$A)) {
    A = 0.01 * diag(k)
  } else {
    A = Prior$A
  }
  if (is.null(Prior$nu)) {
    nu = p + 3
  } else {
    nu = Prior$nu
  }
  if (is.null(Prior$V)) {
    V = nu * diag(p)
  }
}
else {
    V = Prior$V
}

if (length(betabar) != k)
    pandterm("length betabar ne k")
if (sum(dim(A) == c(k, k)) != 2)
    pandterm("A is of incorrect dimension")
if (nu < 1)
    pandterm("invalid nu value")
if (sum(dim(V) == c(p, p)) != 2)
    pandterm("V is of incorrect dimension")
if (missing(Mcmc))
    pandterm("Requires Mcmc argument -- at least R must be included")
if (is.null(Mcmc$R)) {
    pandterm("Requires element R of Mcmc")
} else {
    R = Mcmc$R
}
if (is.null(Mcmc$beta0)) {
    beta0 = rep(0, k)
} else {
    beta0 = Mcmc$beta0
}
if (is.null(Mcmc$sigma0)) {
    sigma0 = diag(p)
} else {
    sigma0 = Mcmc$sigma0
}
if (length(beta0) != k)
    pandterm("beta0 is not of length k")
if (sum(dim(sigma0) == c(p, p)) != 2)
    pandterm("sigma0 is of incorrect dimension")
if (is.null(Mcmc$keep)) {
    keep = 1
} else {
    keep = Mcmc$keep
}
cat(" ", fill = TRUE)
cat("Starting Gibbs Sampler for MVP", fill = TRUE)
cat(" ", n, " obs of ", p, " binary indicators; ", k, " indep vars (including intercepts)", fill = TRUE)
cat(" ", R, " reps; keeping every ", keep, "th draw", fill = TRUE)
cat(" ", fill = TRUE)
cat("Prior Parms:", fill = TRUE)
cat("betabar", fill = TRUE)
print(betabar)
cat("A", fill = TRUE)
print(A)
cat("nu", fill = TRUE)
print(nu)
```r
V <- print(V)

MCMC Parms:
R = R
initial beta = initial beta
initial sigma = initial sigma

sigma0 = print(sigma0)

sigmadraw = matrix(double(floor(R/keep) * p * p), ncol = p)
betadraw = matrix(double(floor(R/keep) * k), ncol = k)
ydraw = matrix(double(floor(R/keep) * length(y)), ncol = length(y))
wnew = double(nrow(X))

betaold = beta0
C = chol(solve(sigma0))
drawwMvpC = function(w, mu, y, sigi) {
  p = ncol(sigi)
  .C("draww_mvp", w = as.double(w), as.double(mu), as.double(sigi),
     as.integer(length(w)/p), as.integer(p), as.integer(y))$w
}
drawwMvp = function(w, X, y, beta, sigmai) {
  Xbeta = as.vector(X %*% beta)
  drawwMvpC(w, Xbeta, y, sigmai)
}
itime = proc.time()[3]

cat("MCMC Iteration (est time to end - min) ", fill = TRUE)
	x<-(t(matrix(rep(1:p,p),p,p)))
  ord2<-matrix(0,p,p)
  for(rr in 1:p){
    ord2[rr,]<-c(rr,x[rr,-rr])
  }

  for (rep in 1:R) {
    sigmai = crossprod(C)
    sepy = matrix(Data$y,ncol=p)
    #rajy = matrix(yy2,ncol=p)
    mu <- matrix(X%*%betanew,ncol=p)
    for (i in 1:p){
```
ord3 <- ord2[i,]

sigoo <- sigmai[ord3[1],ord3[1]]
sigot<-c()
sigto<-c()
sigtt <- c()
cc<-c()
for(aa in 2:p){
sigot<-cbind(sigot,sigmai[ord3[1],ord3[aa]])
sigto<-rbind(sigto,sigmai[ord3[aa],ord3[1]])
xxx <- c()
for(jj in 2:p){
xxx<-cbind(xxx,sigmai[ord3[aa],ord3[jj]])
}
sigtt <- rbind(sigtt,xxx)
cc<-cbind(cc,sepy[-c(1:40),ord3[aa]]-pnorm(mu[-c(1:40),ord3[aa]]))
}
a= qnorm(abs(cc))

for(kk in 1:length(a)){
  if(a[kk]=="-Inf"){a[kk]<-sign(cc[kk])*20}else
  if(a[kk]=="Inf"){a[kk]<-sign(cc[kk])*20}
}
mubar= 0 + sigot%*%(solve(sigtt))%*%t(a)
sigbar= sigoo + sigot%*%(solve(sigtt))%*%(sigto)

py<- pnorm(mu[,i],t(mubar),sigbar)

for (j in Imp$start:Imp$end){
  sepy[j,ord3[1]] = 1.*(runif(1)<py[j])
}

y = matrix(sepy,ncol=1)

wnew = drawMvp(wold, X, y, betaold, sigmai)
zmat = matrix(cbind(wnew, X), nrow = p)
\[ zmat = C \times zmat \]
\[ zmat = \text{matrix}(zmat, \text{nrow} = \text{nrow}(X)) \]
\[ \text{betanew} = \text{breg}(zmat[, 1], \text{zmat[}, 2:(k + 1)], \text{betabar}, A) \]
\[ \text{epsilon} = \text{matrix}((\text{wnew} - X \times \text{betanew}), \text{nrow} = p) \]
\[ S = \text{crossprod}(\text{t(epsilon)}) \]
\[ W = \text{rwishart}(\text{nu} + n, \text{chol2inv(chol(V + S)))} \]
\[ C = W \times C \]
\[
\text{if (rep\%100 == 0)} {
\begin{align*}
\text{ctime} &= \text{proc.time()[3]} \\
\text{timetoend} &= ((\text{ctime} - \text{itime})/\text{rep}) \times (R - 1 - \text{rep}) \\
\text{cat(" ", rep, " ", round(timetoend/60, 1), ", ")"}, \\
\text{fill = TRUE}) \\
\text{fsh()} 
\end{align*}
\]
\[
\text{if (rep\%100 == 0)} {
\begin{align*}
\text{mkeep} &= \text{rep}/\text{keep} \\
\text{betadraw[mkeep,]} &= \text{betanew} \\
\text{sigmadraw[mkeep,]} &= \text{as.vector}(W \times I) \\
\text{ydraw[mkeep,]} &= \text{t(y)} 
\end{align*}
\]
\[
\text{if (rep\%100 == 0)} {
\begin{align*}
\text{ctime} &= \text{proc.time()[3]} \\
\text{cat(" Total Time Elapsed: ", round((\text{ctime} - \text{itime})/60, 2), ", ")"}, \\
\text{attributes(betadraw)$class = c("bayesm.mat", "mcmc")} \\
\text{attributes(betadraw)$mcpa = c(1, R, \text{keep})} \\
\text{attributes(sigmadraw)$class = c("bayesm.var", "bayesm.mat", "mcmc")} \\
\text{attributes(sigmadraw)$mcpa = c(1, R, \text{keep})} \\
\text{return(list(betadraw = betadraw, sigmadraw = sigmadraw, ydraw=ydraw))}
\end{align*}
\]

A.6 Alternative Multivariate Model

nmlist <-c("E2F3","Myc","RAS")
pred<-list()  
breast.prob<-matrix(0,ncol(breast.path[[1]]),length(nmlist)) 
n = 3  
skip<-1  
nn<-25000 
burn<-1000 
its<-nn*skip 

yy<-list() 
for(i in 1:n){ 
xx<-breast.meta[[i]] 
yy[[i]]<-rep(c(0,NA),c(39,ncol(breast.path[[i]])))) 
yy[[i]][grep(nmlist[i],rownames(xx))]<-1 
}
\[ p = n \]

Generator = \( \text{rbind}(\text{breast.meta}[1][,1],\text{breast.meta}[2][,1],\text{breast.meta}[3][,1]) \)

proMg = \( \text{cbind}(\text{breast.meta}[1][,1],\text{breast.meta}[2][,1],\text{breast.meta}[3][,1]) \)

\[
\text{proMg1}\leftarrow\text{cbind}(\text{proMg},\text{matrix}(0,\text{nrow(proMg)},2*p))\\
\text{proMg2}\leftarrow\text{cbind}(\text{matrix}(0,\text{nrow(proMg)},p),\text{proMg},\text{matrix}(0,\text{nrow(proMg)},p))\\
\text{proMg3}\leftarrow\text{cbind}(\text{matrix}(0,\text{nrow(proMg)},2*p),\text{proMg})
\]

proX = \( \text{rbind}(\text{proMg1},\text{proMg2},\text{proMg3}) \)

\[
\text{Data1}\leftarrow\text{list(X=proMg,y=yy[1])}\\
\text{Data2}\leftarrow\text{list(X=proMg,y=yy[2])}\\
\text{Data3}\leftarrow\text{list(X=proMg,y=yy[3])}\\
\text{Mcmc1}\leftarrow\text{list(R=1000,keep=10)}
\]

\[
x\leftarrow\text{t(matrix(rep(1:p,p),p,p))}\\
\text{ord2}\leftarrow\text{matrix(0,p,p)}\\
\text{for(rr in 1:p){}\\
\text{ord2[rr,]}\leftarrow\text{c(rr,x[rr,-rr])}}
\]

\[
\text{newy}\leftarrow y2\\
\text{vara1}\leftarrow\text{vara2}\leftarrow1\\
\text{betadraw}\leftarrow\text{matrix(0,25000,9)}\\
\text{sigdraw}\leftarrow\text{matrix(0,25000,3)}\\
\text{sigdraw2}\leftarrow\text{matrix(0,25000,3)}\\
\text{ydraw}\leftarrow\text{matrix(0,25000,891)}\\
\text{probdraw}\leftarrow\text{matrix(0,25000,774)}\\
\text{codecovar}\leftarrow\text{matrix(0,25000,3)}\\
\text{py}\leftarrow\text{matrix(0,3,258)}
\]

\[
\text{Strasse}\leftarrow\text{rbprobitGibbs.mik(Data=Data1,Mcmc=Mcmc1)}\\
\text{Links}\leftarrow\text{rbprobitGibbs.mik(Data=Data2,Mcmc=Mcmc1)}\\
\text{Strand}\leftarrow\text{rbprobitGibbs.mik(Data=Data3,Mcmc=Mcmc1)}
\]

\[
\text{betaE}\leftarrow\text{Strasse}\$\text{betadraw}[100,]\\
\text{betaM}\leftarrow\text{Links}\$\text{betadraw}[100,]\\
\text{betaR}\leftarrow\text{Strand}\$\text{betadraw}[100,]\\
\text{newy}\leftarrow\text{c(Strasse}\$\text{ydraw}[100,],\text{Links}\$\text{ydraw}[100,],\text{Strand}\$\text{ydraw}[100,])
\]

\[
\text{for(j in 1:25000){}\\
\text{y2}\leftarrow\text{newy}\\\
\text{betaE}\leftarrow\text{betaE}\\\
\text{betaM}\leftarrow\text{betaM}\\\
\text{betaR}\leftarrow\text{betaR}}
\]

\[125\]
Data1=list(X=proMg,y=y2[1:297])
Data2=list(X=proMg,y=y2[298:594])
Data3=list(X=proMg,y=y2[595:891])

Mcmc1=List(R=10,keep=2)
Prior1 = list(betabar=c(betaE[1],betaE[2],betaE[3]))
Prior2 = list(betabar=c(betaM[1],betaM[2],betaM[3]))
Prior3 = list(betabar=c(betaR[1],betaR[2],betaR[3]))
mugen <- rbprobitGibbs(Data=Data1,Prior= Prior1,Mcmc=Mcmc1)
mugen2 <- rbprobitGibbs(Data=Data2,Prior= Prior2,Mcmc=Mcmc1)
mugen3 <- rbprobitGibbs(Data=Data3,Prior= Prior3,Mcmc=Mcmc1)

mu1 <- (proMg%*%mugen$betadraw[5,])
mu2 <- (proMg%*%mugen2$betadraw[5,])
mu3 <- (proMg%*%mugen3$betadraw[5,])
mu <- as.matrix(cbind(cbind(mu1,mu2),mu3))

mu1a <- (proMg%*%mugen$betadraw[5,])[40:297,]
mu2a <- (proMg%*%mugen2$betadraw[5,])[40:297,]
mu3a <- (proMg%*%mugen3$betadraw[5,])[40:297,]

stdmu1 <- (mu1a - mean(mu1a))/sd(mu1a)
stdmu2 <- (mu2a - mean(mu2a))/sd(mu2a)
stdmu3 <- (mu3a - mean(mu3a))/sd(mu3a)
stdmu <- as.matrix(cbind(cbind(stdmu1,stdmu2),stdmu3))

mucomp <- list()
mucomp[[1]] <- cbind(stdmu1,stdmu2)
mucomp[[2]] <- cbind(stdmu1,stdmu3)
mucomp[[3]] <- cbind(stdmu2,stdmu3)
cov(mucomp[[1]])
cov(mucomp[[2]])
cov(mucomp[[3]])

for(ff in 1:p){

    vara1 <- vara2 <- vara3 <- 1
    sz <- 100

    propcov <- runif(sz,-1,1)

    likely <- matrix(0,sz,258)
    for (iq in 1:sz) {
        likely[iq,] <- dmvnorm(mucomp[[ff]],c(0,0),

126
\[
\sigma = \text{matrix}(c(1, \text{propcov}[i], \text{propcov}[j], 1, 2, 2))
\]

\[
\text{llikely} <- \log(\text{llikely} + 0.00000000000001)
\]

\[
\text{llikew2} <- \text{apply}(\text{llikely}, 1, \text{sum})
\]

\[
\text{maxlik} <- \text{max}(\text{llikew2})
\]

\[
\text{rat} <- \exp(\text{llikew2} - \text{maxlik})
\]

\[
u <- \text{runif}(|\text{rat}|)
\]

\[
\text{if}(\text{sum}(u < \text{rat}) > 0) \quad \text{codecovar}[j, ff] <- (\text{propcov}[\text{u < rat}])[1]
\]

\[
\text{print}(\text{c}(j, \text{sum}(u < \text{rat})))
\]

\[
\sigma_i <- \text{matrix}(c(\text{vara1}, \text{codecovar}[j, 1], \text{codecovar}[j, 2], \text{codecovar}[j, 1],
\quad \text{vara2}, \text{codecovar}[j, 3], \text{codecovar}[j, 2], \text{codecovar}[j, 3], \text{vara3}), 3, 3)
\]

\[
\text{sepy} <- \text{matrix}(y_2, \text{nrow} = p)
\]

\[
\text{for} (i \in 1:p) \{
\quad \text{ord3} <- \text{ord2}[i,]
\quad \text{print}(\text{ord3})
\quad \sigma_{oo} <- \sigma_i[\text{ord3}[1], \text{ord3}[1]]
\quad \sigma_{ot} <- c()
\quad \sigma_{tt} <- c()
\quad \sigma \leftarrow c()
\quad \text{for}(i \in 2:p)\{
\quad \sigma_{ot} <- \text{cbind}(\sigma_{ot}, \sigma_i[\text{ord3}[1], \text{ord3}[i]])
\quad \sigma_{tt} <- \text{cbind}(\sigma_{tt}, \sigma_i[\text{ord3}[i], \text{ord3}[1]])
\quad \text{xxx} \leftarrow c()
\quad \text{for}(j \in 2:p)\{
\quad \text{xxx} <- \text{cbind}(\text{xxx}, \sigma_i[\text{ord3}[i], \text{ord3}[j]])
\quad \}
\quad \sigma_{tt} <- \text{cbind}(\sigma_{tt}, \text{xxx})
\quad \sigma \leftarrow \text{cbind}(\sigma, \text{sepy}[\text{c}(1:40), \text{ord3}[aa]] - \text{pnorm}(\mu[\text{c}(1:40), \text{ord3}[aa]])
\quad \}
\]

\[
a = \text{qnorm}(\text{abs}(\sigma))
\]

\[
\text{for}(\text{kk} \in 1:|\text{length}(a)|)\{
\quad \text{if}(\text{a[kk]} == "\text{-Inf}"\{\text{a[kk]} <- \text{sign}(\text{cc[kk]}) * 20\}) \text{else}
\quad \text{if}(\text{a[kk]} == "\text{Inf}"\{\text{a[kk]} <- \text{sign}(\text{cc[kk]}) * 20\})
\quad \}
\]

\[
\text{mubar} = 0 + \text{sigot} \% \% \text{(solve}(\sigma_{tt})) \% \% \text{t}(a)
\]
sigbar = sigoo + sigot %*% (solve(sigtt)) %*% sigto

py[i,] <- pnorm(stdmu[,i], t(mubar), sigbar)

fred <- c(rep(1,39),1:258)

for (k in 40:297){
    sepy[k,ord3[1]] = 1.*(runif(1)<py[ord3[1],fred[k]])
}

sigdraw[j,] <- c(cov(mucomp[[1]])[1,2], cov(mucomp[[2]])[1,2], cov(mucomp[[3]])[1,2])

sigdraw2[j,] <- c(codecovar[j,1], codecovar[j,2], codecovar[j,3])

newy = matrix(sepy,ncol=1)

betaE <- mugen$betadraw[5,]
betam <- mugen2$betadraw[5,]
betar <- mugen3$betadraw[5,]
probdraw[j,] <- cbind(cbind(py[1,],py[2,]),py[3,])
betadraw[j,] <- cbind(cbind(mugen$betadraw[5,],mugen2$betadraw[5,]),mugen3$betadraw[5,])
ydraw[j,]<- newy

}
Appendix B

Python code

B.1 Creating Metagenes

# Aaron Python Code

# Transcribe Data in CSV format to play with in Python
# mas5.gdat, mas5.pdat are lists of cells that need to be lined up with pathprobes?
# Path.probes is a list need to make separate file for each one maybe piece them
# back together later....."B-catenin" "E2F3" "Myc" "Ras" "Src" "PI3K"

# Need to import csv and csv reader

# make list of oncopathways using mas5.pdat and a for loop. Seems to go through mas5.pdat,
# component by component creates dat file as such

# for i in range(len(mas5.pdat):

from numpy import *
import csv
import copy
from pylab import plot,show,scatter,savefig,close,figure

files = csv.reader(open('files.csv'),delimiter=','),quoting=csv.QUOTE_NONNUMERIC)
lfiles = []
for line in files:
lfiles.append(line)

mas5pdat = csv.reader(open('mas5.pdat.csv'),delimiter=','),quoting=csv.QUOTE_NONNUMERIC)
lmas5pdat = []
for line in mas5pdat:
lmas5pdat.append(line)

# pp is path probe
ppBCat = csv.reader(open('path.probes.BCat.csv'),delimiter=','),quoting=csv.QUOTE_NONNUMERIC)
lppBCat = []
for name in ppBCat:
lppBCat.append(name)

ppE2F3 = csv.reader(open('path.probes.E2F3.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lppE2F3 = []
for line in ppE2F3:
lppE2F3.append(line)

ppMyc = csv.reader(open('path.probes.Myc.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lppMyc = []
for line in ppMyc:
lppMyc.append(line)

ppPI3K = csv.reader(open('path.probes.PI3K.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lppPI3K = []
for line in ppPI3K:
lppPI3K.append(line)

ppRas = csv.reader(open('path.probes.Ras.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lppRas = []
for line in ppRas:
lppRas.append(line)

ppSrc = csv.reader(open('path.probes.Src.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lppSrc = []
for line in ppSrc:
lppSrc.append(line)

# cp is cell path
cpBCat = csv.reader(open('cell.path.BCat.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lcpBCat = []
for line in cpBCat:
lcpBCat.append(line)

cpE2F3 = csv.reader(open('cell.path.E2F3.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lcpE2F3 = []
for line in cpE2F3:
lcpE2F3.append(line)

cpMyc = csv.reader(open('cell.path.Myc.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lcpMyc = []
for line in cpMyc:
lcpMyc.append(line)

cpPI3K = csv.reader(open('cell.path.PI3K.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lcpPI3K = []
for line in cpPI3K:
lcpPI3K.append(line)

cpRas = csv.reader(open('cell.path.Ras.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lcpRas = []
for line in cpRas:
lcpRas.append(line)

cpSrc = csv.reader(open('cell.path.Src.csv'), delimiter='\'', quoting=csv.QUOTE_NONNUMERIC)
lcpSrc = []
for line in cpSrc:
lcpSrc.append(line)

# cpa is cell path all
cpaBCat = csv.reader(open('cell.path.all.BCat.csv'), delimiter=',', quoting=csv.QUOTE_NONNUMERIC)
lcpaBCat = []
for line in cpaBCat:
lcpaBCat.append(line)

cpaE2F3 = csv.reader(open('cell.path.all.E2F3.csv'), delimiter=',', quoting=csv.QUOTE_NONNUMERIC)
lcpaE2F3 = []
for line in cpaE2F3:
lcpaE2F3.append(line)

cpaMyc = csv.reader(open('cell.path.all.Myc.csv'), delimiter=',', quoting=csv.QUOTE_NONNUMERIC)
lcpaMyc = []
for line in cpaMyc:
lcpaMyc.append(line)

cpaPI3K = csv.reader(open('cell.path.all.PI3K.csv'), delimiter=',', quoting=csv.QUOTE_NONNUMERIC)
lcpaPI3K = []
for line in cpaPI3K:
lcpaPI3K.append(line)

cpaRas = csv.reader(open('cell.path.all.Ras.csv'), delimiter=',', quoting=csv.QUOTE_NONNUMERIC)
lcpaRas = []
for line in cpaRas:
lcpaRas.append(line)

cpaSrc = csv.reader(open('cell.path.all.Src.csv'), delimiter=',', quoting=csv.QUOTE_NONNUMERIC)
lcpaSrc = []
for line in cpaSrc:
lcpaSrc.append(line)

cpaPI3K = csv.reader(open('cell.path.all.PI3K.csv'), delimiter=',', quoting=csv.QUOTE_NONNUMERIC)
lcpaPI3K = []
for line in cpaPI3K:
lcpaPI3K.append(line)

# data all now successfully in Python start extraction process

# create vector of names in python for the lmas5pdat datafile

nameslmas5pdat = copy.copy([row[0] for row in lmas5pdat])

# create a vector of the intersections between gene names, in probes and total list

BCatprobes = list(set(nameslmas5pdat) & set(lppBCat[0]))
E2F3probes = list(set(nameslmas5pdat) & set(lppE2F3[0]))
Mycprobes = list(set(nameslmas5pdat) & set(lppMyc[0]))
Rasprobes = list(set(nameslmas5pdat) & set(lppRas[0]))
Srcprobes = list(set(nameslmas5pdat) & set(lppSrc[0]))
PI3Kprobes = list(set(nameslmas5pdat) & set(lppPI3K[0]))
# create vector of names in python for the cpBcat datafile

def create_names_vetor(cpBcat):
    nameslcpBCat = [row[0] for row in cpBcat]
    nameslcpE2F3 = [row[0] for row in cpBcat]
    nameslcpMyc = [row[0] for row in cpBcat]
    nameslcpRas = [row[0] for row in cpBcat]
    nameslcpSrc = [row[0] for row in cpBcat]
    nameslcpPI3K = [row[0] for row in cpBcat]

    # create vectors of positions of the gene names
    yyBCatmatch = sorted([nameslcpBCat.index(i) for i in BCatprobes])
    yyE2F3match = sorted([nameslcpE2F3.index(i) for i in E2F3probes])
    yyMycmatch = sorted([nameslcpMyc.index(i) for i in Mycprobes])
    yyRasmatch = sorted([nameslcpRas.index(i) for i in Rasprobes])
    yySrcmatch = sorted([nameslcpSrc.index(i) for i in Srcprobes])
    yyPI3Kmatch = sorted([nameslcpPI3K.index(i) for i in PI3Kprobes])

    # create new list of only genes that are contained in both lists
    yyBCat = [cpBcat[j] for j in yyBCatmatch]
    yyE2F3 = [cpBcat[j] for j in yyE2F3match]
    yyMyc = [cpBcat[j] for j in yyMycmatch]
    yyRas = [cpBcat[j] for j in yyRasmatch]
    yySrc = [cpBcat[j] for j in yySrcmatch]
    yyPI3K = [cpBcat[j] for j in yyPI3Kmatch]

    def create_names_vector(cpBcat):
        nameslmas5pdat = [row[0] for row in cpBcat]
        namesBCatmatch = sorted([nameslmas5pdat.index(i) for i in BCatprobes])
        namesE2F3match = sorted([nameslmas5pdat.index(i) for i in E2F3probes])
        namesMycmatch = sorted([nameslmas5pdat.index(i) for i in Mycprobes])
        namesRasmatch = sorted([nameslmas5pdat.index(i) for i in Rasprobes])
        namesSrcmatch = sorted([nameslmas5pdat.index(i) for i in Srcprobes])
        namesPI3Kmatch = sorted([nameslmas5pdat.index(i) for i in PI3Kprobes])

        testBCatbreastpath = [lmas5pdat[j] for j in namesBCatmatch]
        testE2F3breastpath = [lmas5pdat[j] for j in namesE2F3match]
        testMycbreastpath = [lmas5pdat[j] for j in namesMycmatch]
        testRasbreastpath = [lmas5pdat[j] for j in namesRasmatch]
        testSrcbreastpath = [lmas5pdat[j] for j in namesSrcmatch]
        testPI3Kbreastpath = [lmas5pdat[j] for j in namesPI3Kmatch]

        # reorder list so it is the same order as the one above yy... MUST BE BETTER WAY
        namesBCatbreastpath = sorted(testBCatbreastpath)
        namesE2F3breastpath = sorted(testE2F3breastpath)
        namesMycbreastpath = sorted(testMycbreastpath)
        namesRasbreastpath = sorted(testRasbreastpath)
        namesSrcbreastpath = sorted(testSrcbreastpath)
        namesPI3Kbreastpath = sorted(testPI3Kbreastpath)
BCatbreastpath = copy.copy([testBCatbreastpath[j] for j in testBCatmatch])

namesE2F3breastpath = [row[0] for row in testE2F3breastpath]
namesyyE2F3 = [row[0] for row in yyE2F3]
testE2F3match = list([namesE2F3breastpath.index(i) for i in namesyyE2F3])
E2F3breastpath = copy.copy([testE2F3breastpath[j] for j in testE2F3match])

namesMycbreastpath = [row[0] for row in testMycbreastpath]
namesyyMyc = [row[0] for row in yyMyc]
testMycmatch = list([namesMycbreastpath.index(i) for i in namesyyMyc])
Mycbreastpath = copy.copy([testMycbreastpath[j] for j in testMycmatch])

namesRasbreastpath = [row[0] for row in testRasbreastpath]
namesyyRas = [row[0] for row in yyRas]
testRasmatch = list([namesRasbreastpath.index(i) for i in namesyyRas])
Rasbreastpath = copy.copy([testRasbreastpath[j] for j in testRasmatch])

namesSrcbreastpath = [row[0] for row in testSrcbreastpath]
namesyySrc = [row[0] for row in yySrc]
testSrcmatch = list([namesSrcbreastpath.index(i) for i in namesyySrc])
Srcbreastpath = copy.copy([testSrcbreastpath[j] for j in testSrcmatch])

namesPI3Kbreastpath = [row[0] for row in testPI3Kbreastpath]
namesyyPI3K = [row[0] for row in yyPI3K]
testPI3Kmatch = list([namesPI3Kbreastpath.index(i) for i in namesyyPI3K])
PI3Kbreastpath = copy.copy([testPI3Kbreastpath[j] for j in testPI3Kmatch])

# Standardize Function

def Astand(seq):
    for i in range(len(seq)):
        j = (len(seq[i]))
        m = mean(seq[i][1:j])
        s = std(seq[i][1:j])
        for k in range(1,len(seq[i])):
            t = seq[i][k]
            seq[i][k] = (t - m)/s
    return seq

# Standardize yy and breastpath
yysBCat = Astand(yyBCat)
yysE2F3 = Astand(yyE2F3)
yysMyc = Astand(yyMyc)
yysRas = Astand(yyRas)
yysSrc = Astand(yySrc)
yysPI3K = Astand(yyPI3K)

BCatsbreastpath = Astand(BCatbreastpath)
E2F3sbreastpath = Astand(E2F3breastpath)
Mycsbreastpath = Astand(Mycbreastpath)
Rassbreastpath = Astand(Rasbreastpath)
Srcsbreastpath = Astand(Srcbreastpath)
PI3Ksbreastpath = Astand(P13Kbreastpath)
# svd attempt get in only numbers format....

```
matBCat = copy.copy([row[1:] for row in yysBCat])
matE2F3 = copy.copy([row[1:] for row in yysE2F3])
matMyc = copy.copy([row[1:] for row in yysMyc])
matRas = copy.copy([row[1:] for row in yysRas])
matSrc = copy.copy([row[1:] for row in yysSrc])
matPI3K = copy.copy([row[1:] for row in yysPI3K])

xxBCat = copy.copy([row[1:] for row in BCatsbreastpath])
xxE2F3 = copy.copy([row[1:] for row in E2F3sbreastpath])
xxMyc = copy.copy([row[1:] for row in Mycsbreastpath])
xxRas = copy.copy([row[1:] for row in Rassbreastpath])
xxSrc = copy.copy([row[1:] for row in Srcsbreastpath])
xxPI3K = copy.copy([row[1:] for row in PI3Ksbreastpath])
```

```
# svd test
BCatU,BCatS,BCatV = linalg.svd(matBCat,full_matrices=False)
E2F3U,E2F3S,E2F3V = linalg.svd(matE2F3,full_matrices=False)
MycU,MycS,MycV = linalg.svd(matMyc,full_matrices=False)
RasU,RasS,RasV = linalg.svd(matRas,full_matrices=False)
SrcU,SrcS,SrcV = linalg.svd(matSrc,full_matrices=False)
PI3KU,PI3KS,PI3KV = linalg.svd(matPI3K,full_matrices=False)
```

```
svdduBCat = dot(diag(1/BCatS), transpose(BCatU))
svdduE2F3 = dot(diag(1/E2F3S), transpose(E2F3U))
svdduMyc = dot(diag(1/MycS), transpose(MycU))
svdduRas = dot(diag(1/RasS), transpose(RasU))
svdduSrc = dot(diag(1/SrcS), transpose(SrcU))
svdduPI3K = dot(diag(1/PI3KS), transpose(PI3KU))
```

```
svddu = open('svddu.csv','w')
for row in svdduPI3K:
    svddu.write(','.join([str(i) for i in row])+'
')
```

```
# Breast.meta....
```

```
# make similar number eigen vectors to mik
BCatV2 = array(copy.copy([row[0:3] for row in transpose(BCatV)]))
E2F3V2 = array(copy.copy([row[0:3] for row in transpose(E2F3V)]))
MycV2 = array(copy.copy([row[0:3] for row in transpose(MycV)]))
RasV2 = array(copy.copy([row[0:3] for row in transpose(RasV)]))
SrcV2 = array(copy.copy([row[0:3] for row in transpose(SrcV)]))
PI3KV2 = array(copy.copy([row[0:3] for row in transpose(PI3KV)]))
```

```
# create the matrix multiplication of the appropriate vectors
```
BCatbreastcombo = transpose(dot(svdduBCat,xxBCat))
BCatbreastcombo2 = array(copy.copy([row[0:3] for row in BCatbreastcombo]))

E2F3breastcombo = transpose(dot(svdduE2F3,xxE2F3))
E2F3breastcombo2 = copy.copy([row[0:3] for row in E2F3breastcombo])

Mycbreastcombo = transpose(dot(svdduMyc,xxMyc))
Mycbreastcombo2 = copy.copy([row[0:3] for row in Mycbreastcombo])

Rasbreastcombo = transpose(dot(svdduRas,xxRas))
Rasbreastcombo2 = copy.copy([row[0:3] for row in Rasbreastcombo])

Srcbreastcombo = transpose(dot(svdduSrc,xxSrc))
Srcbreastcombo2 = copy.copy([row[0:3] for row in Srcbreastcombo])

PI3Kbreastcombo = transpose(dot(svdduPI3K,xxPI3K))
PI3Kbreastcombo2 = copy.copy([row[0:3] for row in PI3Kbreastcombo])

# rbind v and combo
BCatbreastmeta = []
for line in BCatV2:
    BCatbreastmeta.append(line)

for line in BCatbreastcombo2:
    BCatbreastmeta.append(line)

E2F3breastmeta = []
for line in E2F3V2:
    E2F3breastmeta.append(line)

for line in E2F3breastcombo2:
    E2F3breastmeta.append(line)

Mycbreastmeta = []
for line in MycV2:
    Mycbreastmeta.append(line)

for line in Mycbreastcombo2:
    Mycbreastmeta.append(line)

Rasbreastmeta = []
for line in RasV2:
    Rasbreastmeta.append(line)

for line in Rasbreastcombo2:
    Rasbreastmeta.append(line)

Srcbreastmeta = []
for line in SrcV2:
    Srcbreastmeta.append(line)

for line in Srcbreastcombo2:
    Srcbreastmeta.append(line)

PI3Kbreastmeta = []
for line in PI3KV2:
    PI3Kbreastmeta.append(line)

for line in PI3Kbreastcombo2:
    PI3Kbreastmeta.append(line)

BCatmetagene1 = [row[0] for row in BCatbreastmeta]
BCatmetagene2 = [row[1] for row in BCatbreastmeta]

E2F3metagene1 = [row[0] for row in E2F3breastmeta]
E2F3metagene2 = [row[1] for row in E2F3breastmeta]

Mycmetagene1 = [row[0] for row in Mycbreastmeta]
Mycmetagene2 = [row[1] for row in Mycbreastmeta]

Rasmetagene1 = [row[0] for row in Rasbreastmeta]
Rasmetagene2 = [row[1] for row in Rasbreastmeta]

Srcmetagene1 = [row[0] for row in Srcbreastmeta]
Srcmetagene2 = [row[1] for row in Srcbreastmeta]

PI3Kmetagene1 = [row[0] for row in PI3Kbreastmeta]
PI3Kmetagene2 = [row[1] for row in PI3Kbreastmeta]

#Pcol = [(i,0,1-i) for i in lPpred[0]]

#y= zeros(len(PI3Kmetagene1))
#y[c=='r']=1

### cmolnames = lcpMyc[0]

# cmolnames = []

# cmolindex = [i = -1
# try:
#     while 1:
#         i = cmolnames.index('GFP', i+1)
#         cmolindex.append(i)
# except ValueError:
#     pass

# cmolindex2 = [i = -1
# try:
#     while 1:
#         i = cmolnames.index('Myc', i+1)
#         cmolindex2.append(i)
# except ValueError:
#     pass

# cm = repeat('b',len(Mycmetagene1))
for i in cmolindex:
    cm[i]='r'

for j in cmolindex2:
    cm[j] = 'g'

figure(num=1)
scatter(Mycmetagene1,Mycmetagene2,c=cm)
savefig("Myc.png")
close(1)

######################################################################
colnames = lcpPI3K[0]
colindex = []
i = -1
try:
    while 1:
        i = colnames.index('GFP', i+1)
        colindex.append(i)
except ValueError:
    pass

colindex2 = []
i = -1
try:
    while 1:
        i = colnames.index('PI3K', i+1)
        colindex2.append(i)
except ValueError:
    pass

c = repeat('b',len(PI3Kmetagene1))

for i in colindex:
    c[i]='r'

for j in colindex2:
    c[j] = 'g'

figure(num=2)
scatter(PI3Kmetagene1,PI3Kmetagene2,c=c)
savefig("PI3K.png")
close(2)

######################################################################
cbolnames = lcpBCat[0]
cbolindex = []
i = -1
try:
    while 1:
        i = cbolnames.index('GFP', i+1)
        cbolindex.append(i)
except ValueError:
    pass

cbolindex2 = []
i = -1
try:
    while 1:
        i = cbolnames.index('Bcat', i+1)
        cbolindex2.append(i)
except ValueError:
    pass

cb = repeat('b',len(BCatmetagene1))

for i in cbolindex:
    cb[i] = 'r'

for j in cbolindex2:
    cb[j] = 'g'

figure(num=3)
scatter(BCatmetagene1,BCatmetagene2,c=cb)
savefig("BCat.png")
close(3)

#########################################################################

ceolnames = lcpE2F3[0]

ceolindex = []
i = -1
try:
    while 1:
        i = ceolnames.index('GFP', i+1)
        ceolindex.append(i)
except ValueError:
    pass

ceolindex2 = []
i = -1
try:
    while 1:
        i = ceolnames.index('E2F3', i+1)
        ceolindex2.append(i)
except ValueError:
    pass
ce = repeat('b', len(E2F3metagene1))
for i in ceolindex:
    ce[i] = 'r'
for j in ceolindex2:
    ce[j] = 'g'

figure(num=4)
scatter(E2F3metagene1, E2F3metagene2, c=ce)
savefig("E2F3.png")
close(4)

#########################################################################

csolnames = lcpSrc[0]
csolindex = []
i = -1
try:
    while 1:
        i = csolnames.index('GFP', i+1)
        csolindex.append(i)
except ValueError:
    pass

csolindex2 = []
i = -1
try:
    while 1:
        i = csolnames.index('Src', i+1)
        csolindex2.append(i)
except ValueError:
    pass

cs = repeat('b', len(Srcmetagene1))
for i in csolindex:
    cs[i] = 'r'
for j in csolindex2:
    cs[j] = 'g'

figure(num=5)
scatter(Srcmetagene1, Srcmetagene2, c=cs)
savefig("Src.png")
close(5)

#########################################################################
crolnames = lcpRas[0]
crolindex = []
i = -1
try:
    while 1:
        i = crolnames.index('GFP', i+1)
        crolindex.append(i)
except ValueError:
    pass


crolindex2 = []
i = -1
try:
    while 1:
        i = crolnames.index('RAS', i+1)
        crolindex2.append(i)
except ValueError:
    pass


cr = repeat('b',len(Rasmetagene1))

for i in crolindex:
    cr[i]='r'

for j in crolindex2:
    cr[j] = 'g'

figure(num=6)
scatter(Rasmetagene1,Rasmetagene2,c=cr)
savefig("Ras.png")
close(6)


B.2 Example Bayesian Regression Code in Python
(Ras Pathway)

from pymc import *
from numpy import *

# Data
y= array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])

observed_index = 20

xone = array([-0.22427114, -0.22219351, -0.23725514, -0.2170428 , -0.24724411,
              -0.19691461, -0.20526798, -0.22568055, -0.22726782, -0.22943258,
              0.21391197, 0.22156098, 0.23147122, 0.22984244, 0.20074775,
              0.22484066, 0.21114924, 0.23922962, 0.22567339, 0.23414297,
              0.00578448, 0.00428332, -0.00185819, 0.0495441 , -0.24724411])
-0.0269777, 0.03701261, 0.0007988, 0.05947868, 0.02633287,
-0.06747807, -0.0616408, -0.05368694, -0.4878321, -0.0688179,
-0.04313803, -0.00806406, 0.0186102, -0.1376469, 0.00122958,
-0.04780591, -0.02710229, 0.06684333, 0.18996713, 0.09112801,
-0.00791606, 0.02323257, -0.3064534, 0.3259145, 0.0703436,
-0.02413609, 0.05981528, 0.0226439, -0.02710229, 0.06684333,
-0.04780591, -0.0312503, 0.02397471, 0.04230821, -0.0356212,
-0.0088578, 0.00244951, 0.01794727, -0.01581704, -0.01916152,
-0.02219598, 0.05489548, 0.02150343, 0.00791606, 0.04230821,
-0.1221254, 0.0091002, 0.0235848, 0.07059088, -0.06384625,
-0.02938994, -0.03402935, -0.0312503, -0.01382412, -0.02312966,
-0.0300719, -0.03770875, 0.11328218, -0.00372694, -0.04209748,
-0.03225413, 0.00765887, 0.018927, -0.2158717, -0.6694475,
-0.00072591, -0.05980477, 0.1935539, 0.1069264, -0.05511605,
-0.0376318, -0.04497781, -0.0124304, 0.05198186, 0.2399578,
-0.00059292, -0.0129429, -0.0646054, -0.02693713, -0.00420748,
-0.02219598, 0.05489548, -0.00173622, -0.01720415, -0.0315467,
-0.02790954, 0.2059975, -0.0477873, 0.3376796, 0.0302095,
-0.01742408, -0.01187471, 0.3381157, 0.13385764, 0.06870042,
-0.02106084, -0.0165555, -0.3375693, 0.0801994, 0.0335541,
-0.04989781, -0.00633939, -0.3384627, 0.0027695, -0.1155179,
-0.05689027, -0.0009806, 0.8006187, 0.184478, -0.08235274,
-0.05427639, 0.01149281, -0.292336, 0.06457339, 0.0113641,
-0.0097398, -0.00457849, 0.4368648, -0.02063964, 0.1997797,
-0.05830206, -0.0151283, 0.00407275, -0.0088312, -0.00579994,
-0.05415974, 0.7848909, 0.3506884, 0.8994684, -0.0069064,
-0.03092761, -0.09180504, -0.07216172, 0.00227367, 0.04920907,
-0.04542373, 0.0157101, -0.00813563, 0.06113122, -0.02532782,
-0.0340282, -0.03191419, -0.0597324, 0.02614901, -0.00973314,
-0.03758378, -0.06817221, -0.05315095, 0.03459772, -0.1110725,
-0.0199977, -0.02209424, 0.06194841, -0.08151535, 0.07132078,
-0.01361474, -0.00731969, 0.01348054, -0.04255992, 0.00494305,
-0.05629142, -0.04851146, -0.02270391, -0.0949619, -0.06625074,
-0.02625221, -0.0132983, 0.02018319, 0.01809097, -0.03629124,
-0.0609188, -0.00957787, 0.1957395, 0.01934093, -0.0234828,
-0.02523073, 0.0180606, 0.0128988, 0.0167574, 0.00239643,
-0.00522379, -0.03209383, -0.03572582, -0.01867795, 0.05292795,
-0.04774053, 0.01242366, 0.01733256, 0.01802618, -0.05486986,
-0.0081973, 0.07494989, 0.0120898, 0.01049469, 0.01985969,
-0.09175909, -0.00933447, 0.06721472, -0.02947765, 0.06178928,
-0.00823307, -0.02193248, 0.09735301, -0.1208444, -0.03823121,
-0.01300596, -0.0093294, 0.04421773, 0.00975264, -0.06276979,
-0.02027889, -0.02247204, -0.00934289, -0.06737807, -0.00162077,
-0.0095886, -0.02570203, -0.05215871, 0.02304654, -0.09385014,
-0.00386185, -0.00986231, 0.04926357)

xtwo = array([(0.02555819, -0.09441296, -0.12233287, 0.11114263, -0.1343338,
-0.00373936, -0.11272966, -0.08945883, 0.12925169, 0.0703436,
0.08314385, 0.03820605, 0.46162831, 0.31456271, -0.15003913, 0.141])
-0.26300136, -0.18930507, 0.41771235, -0.30216267, -0.4346994,
0.03283248, 0.19035644, -0.30957872, 0.36628788, 0.09762075,
0.01797934, 0.00962488, -0.21702204, 0.13596708, 0.29015815,
-0.1977448, 0.08399693, 0.15365796, 0.00404452, 0.19938205,
-0.06968412, -0.27956694, -0.0377842, 0.11878927, -0.04865747,
0.03948791, -0.15777297, 0.38095712, -0.03484625, 0.49869963,
-0.0131692, -0.22970652, 0.00228078, 0.00082576, -0.06498996,
-0.25314661, 0.07495867, 0.18618342, 0.03433335, 0.27122713,
0.00451356, -0.17614144, -0.04444548, -0.07869295, 0.1149652,
-0.04061362, -0.12387936, 0.3467527, -0.15526677, -0.08103984,
-0.13578859, 0.10348971, 0.03948971, -0.15377287, 0.04986963,
0.09761622, -0.25937816, -0.17672635, 0.13150027, -0.06561514,
-0.0176755, 0.35136121, 0.03442652, -0.04954285, 0.07219877,
-0.03436522, -0.27609565, -0.03267359, 0.2457237, -0.0344865,
-0.12905084, 0.47385113, -0.09836301, -0.19366602, -0.10713225,
-0.1640797, -0.06150876, 0.07197258, -0.24206509, -0.01239478,
0.03342663, -0.1038635, 0.0860635, 0.16337073, -0.24472292,
-0.09531006, -0.1008941, -0.1783049, 0.17080562, -0.16198101,
-0.09152119, -0.20491286, -0.20706466, 0.0605963, -0.12887408,
-0.06078585, 0.30405735, 0.25855659, 0.09244613, 0.04808279,
0.1498437, 0.42782861, -0.18670077, -0.28148358, -0.02306072,
-0.07306573, -0.04171641, 0.01906465, -0.1113604, 0.0041914,
-0.20329661, 0.02464289, 0.47915316, 0.12188683, -0.08843703,
-0.10801556, -0.09948856, 0.0693740, -0.31529842, -0.35040723,
0.05140099, -0.25043052, 0.10162419, -0.46204376, 0.0081883,
0.51291551, -0.25045774, 0.30271959, 0.3501231, -0.09792423,
-0.03615941, -0.2209129, -0.28688613, 0.12241176, -0.05627316,
-0.1606944, -0.05839408, -0.12292477, 0.04655582, 0.12738286,
-0.12817529, -0.01563087, -0.02859197, 0.18753746, 0.21889312,
0.03032913, -0.02533598, 0.03921847, -0.09504487, -0.12176761,
0.09907219, 0.04819306, -0.06938761, 0.05143124, 0.04776197,
0.1062374, -0.23420102, 0.0868184, -0.06803224, 0.004506373,
-0.06396234, -0.27470668, -0.02182806, -0.11986894, -0.05909866,
-0.29221358, -0.03148188, -0.03573913, 0.06521201, -0.10660902,
-0.04508364, 0.02102538, 0.12017861, 0.15328641, 0.00415297,
0.13920616, -0.20582218, 0.00443898, 0.18543876, 0.17768956,
0.25129211, -0.45375072, -0.17800103, -0.05930513, -0.15052322,
0.06800044, -0.11399009, 0.27187284, 0.25140125, -0.14725482,
0.01085773, 0.02084672, 0.02647413, -0.0900204, 0.03175714,
0.29392622, 0.14900335, 0.26274429, 0.20168994, -0.0115241,
0.22118494, 0.03158669, -0.05164429, 0.06870483, 0.30909432,
0.13403904, -0.0609131, 0.32102078, -0.14084734, 0.03632979,
-0.09753426, 0.23846874, -0.06189227, 0.23470644, 0.27038813,
0.17756855, 0.19303907, -0.18212965, 0.40307841, 0.08035904,
-0.2092487, 0.06096169, 0.19721575, -0.07087476, -0.10846263,
-0.04936461, 0.00486787, 0.06418254, -0.032734, 0.04286672,
-0.03230949, -0.1306703, 0.10041157, -0.04454406, 0.42367123,
0.35916688, 0.29969708, -0.1499844, 0.13394995, -0.05974322,
0.16081446, 0.11943366, -0.05272278]}

#priors

142
bn = Normal('bn', 0, 0.01, value = 0.0)
bo = Normal('bo', 0, 0.01, value = 0.0)
btt = Normal('bt', 0, 0.01, value = 0.0)

#model
@deterministic
def p_obs(bn=bn, bo=bo, bt=bt, xone=xone[:observed_index], xtwo=xtwo[:observed_index]):
    """Probabilities for observed outcomes""
    return invlogit(bn + bo*xone + bt*xtwo)
@deterministic
def p_pred(bn=bn, bo=bo, bt=bt, xone=xone[observed_index:], xtwo=xtwo[observed_index:]):
    """Predicted probabilities""
    return invlogit(bn + bo*xone + bt*xtwo)

y_obs = Bernoulli('y_obs', p=p_obs, value=y, observed=True)
y_pred = Bernoulli('y_pred', p=p_pred)