Genome-wide analyses using bead-based microarrays

Mark James Dunning

Jesus College

A dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy

Department of Oncology,
Cancer Research Uk, Cambridge Research Institute,
Li Ka Shing Centre Robinson Way,
Cambridge, CB2 0RE,
United Kingdom.

Email: md392@cam.ac.uk

September 4, 2008

Dedicated to my parents

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

This thesis does not exceed the specified length limit of 60,000 words as defined by the Clinical Medicine Degree Committee.

This thesis has been typeset in 12pt font using LaTeX2ε according to the specifications defined by the Board of Graduate Studies and the Clinical Medicine Degree Committee.

Genome-wide analyses using bead-based microarrays

Summary

Mark James Dunning
Jesus College

September 4, 2008

Microarrays are now an established tool for biological research and have a wide range of applications. In this thesis I investigate the BeadArray microarray technology developed by Illumina. The design of this technology is unique and gives rise to many computational and statistical challenges. However, I show how knowledge from other microarray technologies can be used to our advantage.

I describe the beadarray software package, which is now used by researchers around the world. The development of this software was motivated by the fact that Illumina's software (BeadStudio) gives a summarised view of Illumina data and does not gives users any control over several processing steps that were found to be crucial for other microarray technologies. A main feature of beadarray is the ability to access raw data. The advantages of such data include the ability to perform more detailed quality assessment and greater control over the analysis at all stages. The analysis of a control experiment shows that the processing steps used in BeadStudio can be improved. In particular, utilising variances calculated from the raw data can increase the ability to detect genes which have different expression levels between samples, a common goal for microarray studies. The data from the control experiment are made available for other researchers to use and validate their own analysis methods.

One issue discovered during the analysis of the control experiment was that only half of the intended genes could be reliably measured due to problems in the design of the probes targetting particular genes. By considering a large set of publicly available Illumina arrays, I show how such unreliable measurements can affect the analysis of Illumina data. I also show how potential problems can be identified in advance of an experiment and incorporated into an analysis pipeline.

Preface

This thesis describes work undertaken in the Computational Biology group in the Department of Oncology at the University of Cambridge between January 2005 and June 2008. This section of the thesis proved one of the most difficult to write because there are so many people to which I am grateful! During this time I had the good fortune to work on many interesting topics and be surrounded by so many talented individuals that made working in the group thoroughly enjoyable. I would firstly like to thank my supervisor, Simon Tavaré for his patient support and guidance over the past 3 years. The other members of the group have helped me in many ways over the years with scientific discussions and moral support. I am very grateful to Natalie Thorne, Matthew Ritchie, Andy Lynch, Nuno Barbosa-Morais and Christina Curtis for their constructive comments on my work and it has been a real pleasure collaborating with them. My fellow students in the lab have also contributed greatly to the stimulating environment, and I am sure each one has an exciting future ahead of them. I am also grateful to the training I received at University of York and my supervisors Jason Levesley and Garib Murshudov for their encouragement to take up this PhD.

There are many individuals that provided much-needed support outside the world of bioinformatics. In particular my housemates in Malcolm Street (past and present), and especially to Anna Julian for her support over the past few months. I am also quite fortunate to keep in close contact with several school-friends and it is always a real pleasure to meet up and hear about their achievements.

Finally, special thanks must be given to my parents for their continued encouragement throughout the years, and also to my grandparents who are sadly not here to witness the completion of this work.

Contents

Summary **iii**

Preface **iv**

List of abbreviations **xx**

1 Introduction **1**

1.1 Overview of DNA and RNA 1

1.2 Gene expression microarrays 3

1.3 Illumina bead-based microarrays 6

1.4 Pre-processing and analysis of microarray data 7

1.4.1 Image Capture and Processing 9

1.4.2 Background Correction 9

1.4.3 Quality Assessment 10

1.4.4 Normalisation 13

1.4.5 Detecting DE genes 13

1.4.6 Further analysis 14

1.5 Computational challenges 14

1.6 Thesis outline 15

2 Human whole expression profiling using Illumina microarrays **17**

2.1 Introduction 17

2.1.1 Background Correction 18

2.1.2 Normalisation 21

2.1.3 Detecting DE genes 23

2.1.4 Filtering 25

2.1.5 Probe Annotation 26

2.2 Illumina BeadChip arrays 27

2.2.1 Direct hybridisation assay 28

2.2.2 Illumina scanning and analysis software 29

2.3 Illumina analysis methods 30
2.3.1 Image processing and background correction 30
2.3.2 Normalisation 31
2.3.3 Filtering 31
2.3.4 Detecting DE genes 33
2.3.5 Other analysis options supported by BeadStudio 33
2.4 Why Illumina arrays are attractive to researchers 34
2.5 Early uses of Illumina arrays in the literature 35
2.5.1 Publicly available Human6 data 37
2.6 Conclusions 40

3 beadarray: open-source software for Illumina bead-based microarrays 42
3.1 Introduction 42
3.2 The Bioconductor project 43
3.3 Processing bead-level data using beadarray 44
3.3.1 Reading bead-level data into beadarray 45
3.3.2 Representation of bead-level data in beadarray 47
3.3.3 Visualisation of bead-level data 48
3.3.4 Creating bead-summary data 49
3.3.5 Proceeding with bead-summary analysis 50
3.4 Analysing Gene expression bead-summary data 51
3.4.1 Reading BeadStudio output into beadarray 51
3.4.2 Visualisation of bead-summary data 53
3.4.3 Further analysis of bead-summary data 54
3.5 Conclusions 55

4 Investigation into the pre-processing of Illumina data 58
4.1 Introduction 58
4.2 Investigating the BioC07 dataset 59
4.2.1 Image Processing 59
4.2.2 Spatial Plots 61
4.2.3 Outlier Detection 62
4.2.4 Analysis of summarised BioC07 data 64
4.2.5 A simple differential expression analysis 67
4.3 Observations from the HapMap dataset 69
4.4 Discussion 71

5 Analysis of an Illumina spike-in experiment 75
5.1 Introduction 75
5.2 Control experiments for microarrays 77

5.3 The Illumina spike-in experiment 79
5.4 Topics of investigation and methods 79
5.4.1 Image analysis and background correction 80
5.4.2 Summarisation 80
5.4.3 Normalisation 81
5.4.4 Linear models and contrasts 81
5.4.5 Annotation 82
5.5 Results 83
5.5.1 Bead-level issues 83
5.5.2 The effects of pre-processing on differential expression analysis 86
5.5.3 Probe properties and annotation considerations 92
5.6 Discussion 96
5.6.1 Data quality 98
5.6.2 Local background estimation and subtraction 99
5.6.3 Summarisation 100
5.6.4 Normalisation 100
5.6.5 Differential expression analysis 101
5.6.6 Annotation 102
5.6.7 Application to other Illumina technologies 103
5.7 Validating a variance-stabilising transformation 103
5.7.1 Methods 105
5.7.2 Applying VST to the spike-in experiment 106
5.7.3 Discussion 108

6 Optimising the analysis of Illumina data by using prior knowledge 112
6.1 Introduction 112
6.2 Data and Methods 114
6.2.1 The MAQC dataset 114
6.2.2 The GEO dataset 114
6.2.3 Filtering Methods Used 115
6.2.4 Use of annotation information 116
6.2.5 Differential expression analysis 116
6.3 Results 117
6.3.1 General Observations 117
6.3.2 Filtering 118
6.3.3 Differential expression 125
6.4 Discussion 133

Conclusions 137

A Papers published during this work 140

B Code for Chapter 4 142

C Supplementary Figures 145

List of Tables

2.1 Summary of GEO datasets derived using Human6 chips. For each entry we list the citation, number of arrays used, type of filtering applied to the data, and normalisation method applied. All datasets had preliminary analysis done using BeadStudio. 37

5.1 Table showing the reannotation of the probes sequences for all non-spikes in the spike-in experiment. The 23,983 probes on Strip 1 and 22,022 probes on Strip 2 are divided into categories (see page 82) describing various annotation problems. 94

6.1 Results of reannotation of the Illumina Human6 Version 1 platform. The 26,091 probes on Strip 1 and 21,198 probes on Strip 2 are divided into categories describing various annotation problems. "Good" denotes probes that had a complete genomic match to the exonic region of a known transcript. . . 117

List of Figures

1.1 This public domain image shows a schematic diagram of a typical two-colour microarray experiment to compare DNA from a cancer cell to that of a normal cell. 5

1.2 Constructing an Illumina “array of arrays”, in this case a SAM: **A**) Each bead sits in a pre-created well on the surface of an array and has probes attached that are complementary to a particular genomic sequence of interest. In this figure, only one sequence is shown, although the bead will have thousands of these sequences attached. The bead also has a unique identifier sequence which is used for decoding purposes. **B**) Each array has around 50,000 beads that are randomly arranged. Around 1,500 distinct bead types are represented around 30 times each. **C**) A matrix of 96 arrays is constructed, each array being uniquely prepared and thus having a different arrangement of beads. Image from (Kuhn *et al.*, 2004). 8

1.3 QA plots for a two-colour microarray experiment provided with the limma user guide. A) Boxplots of the red foreground for four arrays in the experiment, showing the median values and inter-quartile range of each array. Ideally, the foreground measured on each array should have roughly similar distributions. However, the intensities on the first two arrays are generally higher. B) An MA-plot for the red and green channels for a particular array, with the log-ratio (M) plotted on the y-axis and log-average (A) on the x-axis. We would expect that most probes are not DE, and should therefore be centred along $M = 0$. In this example, most points are away from this line and adjustment is required to remove this trend. C) Imageplot of the log-ratios for a particular array, with green and red representing low and high intensities respectively. Ideally, a random scattering of colours should be seen. However, a red streak is seen in the 3rd column of the array. Spots affected by this artefact have log-ratios that are systematically higher and might not be attributed to differences in the biological conditions being studied. 11

2.1 Illustrative example of the "fanning effect" for microarray data. This MA-plot compares two replicates of the same biological sample, and hence no differential expression ($M = 0$) should be expected. However, for A$<$ 7, the M-values are seen to have a wide range of values. The coloured spots indicate control probes on the array. Figure courtesy of Gordon Smyth. 19

2.2 A cartoon representation of a bead in the direct hybridisation assay. Each bead has an address sequence attached for decoding purposes and a 50 base sequence specific to each bead type. The probe sequence is fluorescently labelled and designed to hybridise to a particular genomic sequence. Note that the diagram is not shown to scale and many thousands of sequences are attached to each bead. Figure courtesy of Illumina. 28

2.3 Diagram of the steps used for calculating the foreground and background of a bead. Using the coordinates determined during the decoding step, a centre for the bead is known, indicated by a cross in this figure. The intensities of the four closest pixels in a 3 × 3 square around the centre (red square) are then used in a weighted average to calculate the foreground. The five dimmest pixels within a 17 × 17 square around the bead centre (orange square) are averaged to give the background intensities. In this figure, the five dimmest pixels are indicated by the yellow squares. Figure courtesy of Dr. Matthew Ritchie. 32

3.1 An overview of the beadarray software and the various tasks it can perform compared to BeadStudio. The software can be used to analyse either bead-level data or data exported from BeadStudio, although availability of bead-level data allows a more flexible analysis. 46

4.1 Foreground (A), background (B) and background corrected (C) intensities of all beads in the BioC07 dataset. Arrays are coloured separately for Sample A (blue) and Sample B (red). The foreground intensities are seen to be consistent across the dataset, with the exception of arrays 1 and 6 which are generally have higher intensity. The background intensity has extremely low variability on all arrays. 60

4.2 Imageplots of foreground intensity for the BioC07 dataset, with replicates of sample A in the top row, and sample B in the bottom row. Red and yellow denote high and low intensity regions respectively. Clear spatial artefacts can be seen for arrays 1, 3 and 6. 62

4.3 The locations of beads that are found to be outliers on arrays 1, 3 and 6, which were seen to have spatial artefacts in Figure 4.2. With the exception of Array 3, the locations of outliers correspond well with the spatial artefacts seen by eye. The outlier removal method implemented by Illumina was used, which excludes beads using a cut-off of 3 MADs from the median of the unlogged bead intensities for each bead type. . . . 64

4.4 An overview of the BioC07 dataset after summarising the bead-level data. The summarised expression levels for all arrays are seen to be in good agreement, although arrays 1 and 6 have slightly higher medians. Array 1 also has fewer bead types with extreme high intensities. The number of beads after outlier removal are also shown for all arrays in the BioC07 dataset. The average number of beads for a bead type in a given array is generally around 30, although arrays 1, 3 and 6 have lower numbers of replicates. No bead type on any array has fewer than 10 replicates. 65

4.5 MA plots constructed using selected replicates of Sample A (top row) and replicates of Sample B (bottom row). Comparing Array 1 to arrays that have the same biological sample hybridised yields many genes with log-ratios away from 0 in a non-linear fashion. This trend is not seen for other comparisons of Sample A. The log-ratios generated using Array 6 are more variable than other comparisons of this sample and also systematically greater than 0. However there are many normalisation schemes that might correct for this. 66

4.6 Comparison of the volcano plots produced for a differential expression analysis involving all arrays (left) and with Array 1 removed (right). The y-axis shows a measure of evidence for differential expression (log-odds) and the x-axis shows the estimated coefficients from the linear model. Red dots indicate genes with positive log-odds (roughly corresponding to greater than 50% chance of being DE) in the analysis that excludes Array 1. Removal of Array 1 from the analysis is seen to improve our ability to detect DE genes. 68

4.7 The percentage of outliers for a random selection of 200 arrays from the HapMap dataset, with outliers below and above the median shown in blue and red respectively. On the left, outliers were removed on the unlogged scale using a 3 MAD cut-off, and on the right a $\log_2$ transformation was applied prior to removing outliers. Without applying a $\log_2$ transformation, we find many more outliers above the median than below, whereas we get a more even distribution of outliers after a $\log_2$ transformation. 70

4.8 An overview of the overall array intensities for the HapMap dataset. The median intensities were calculated for each of the 1400 arrays and plotted according to which SAM (indicated by a 7 digit number) the arrays belong to. Clear differences are seen between the chips. 71

5.1 Raw foreground and background intensities for each strip on a typical BeadChip. Each BeadArray is made up of two strips (colour-coded) on the chip surface. The consistency of foreground and background signals between arrays is evident from this plot, as is the tendency for beads from Strip 1 to have higher intensities than those from Strip 2. 84

5.2 The average bias (A) and $\log_2$ variance (B) versus percentage of simulated outliers plotted for each summary method. In panel A, we see that Illumina's summary method can handle up to about 30% of saturated intensities before the bias starts to increase dramatically. The trimmed mean breaks down much earlier, at around 5%. The median is comparable to Illumina's method. Similar trends can be noticed in the variance (B). 85

5.3 Boxplots of the means (A) and variances (B) for the 33 spikes on all arrays in the experiment after outlier removal. The boxplots are arranged in decreasing order of spike concentration, with different background correction methods labeled in different colours. The no background adjustment option shows dramatic attenuation in signal, which begins at a higher concentration than the other background correction options. . . . 87

5.4 MA-plots comparing the bead-summary values for one array with spike concentration 3pM to an array with spike concentration 1pM. An increased density of points is indicated by darker shades of blue. Red points highlight the spike genes. The horizontal line at M=1.73 represents the intended log-ratio for the spike genes, and the line at M=0 is the desired level for the remaining non-spikes. Each panel shows the data processed using different background correction methods. Panel A shows the data with no background adjustment, while in panel B local background has been subtracted and in panel C the data have been background subtracted and background normalised. When the data are background subtracted, the range of M and A-values increases and the spike genes are closer to the true value than for the non-adjusted data. BGN (see page 31) produces the most variable M-values and over-estimates the M-values for the spikes. 89

5.5 The distribution of background subtracted and summarised intensities for 50 negative controls across all arrays in the experiment, ordered by increasing median. Each control is a bead type with a random sequence attached that should not hybridise to any target in the genome. Despite this, some controls clearly appear to show consistently higher intensity than others. 90

5.6 Boxplots of the log-odds scores (A) and log-ratios (B) obtained after fitting a linear model to all genes across all arrays in the spike experiment and making contrasts between 3pM and 1pM. A separate box is shown for each background correction method with a standard linear model and a weighted $\log_2$ analysis. Two separate boxplots are shown for each method and weighting scheme to indicate the log-odds scores for the spikes (bold colours) and non-spikes (transparent). The weighted $\log_2$ analysis improves the log-odds scores for the spikes without increasing the log-odds for the non-spikes, which represents an increase in power to detect true differential expression. In panel B, we show that the log-ratios for the spikes are under-estimated when the data are not background adjusted, whereas the background subtracted and *normexp* processed data recover values much closer to the true log fold-change (dashed line, M=1.73). 91

5.7 The $\log_2$ intensities for the 33 spikes on each array estimated using the linear model. Each spike is indicated by a different colour and line. Despite being added at the same concentration, consistent differences are seen between the spikes, for example, *ela_2* consistently has the lowest intensity. 93

5.8 Boxplots of all bead types intensities on an arbitrary Strip 1 array grouped according to the reannotation of the probe sequence assigned to the bead type. Bead types with reliable annotation (the "Good" category) are seen to have higher intensity compared to others. 96

5.9 Normalised $\log_2$ intensities for all non-spikes on Strip 1 of a particular array in the experiment grouped according to the number of As, Ts, Gs or Cs in the sequence for the probe. The normalised $\log_2$ intensities and bead type variances are also shown in terms of GC content. The width of each box is proportional to the number of observations. Probes with higher GC content are shown to have higher intensity on average and a lower variance. Finally, estimated effect sizes are shown for each base position relative to having a T at that position. The normalised intensities are seen to be higher if a G or C is present at the first base in the sequence and have a lower variance. However, no other systematic trend is seen. . . 97

5.10 The log-odds ranking of all non-spikes on Strip 1 in the contrast between 3pM and 1pM aggregated according to the GC content of each probe. Probes with a GC content of 18-21 are generally ranked higher in the list. The width of each box is proportional to the number of observations. 98

5.11 Demonstrating the VST transformation for an array in the spike-in experiment. A) The undesirable relationship between bead type means and standard deviations is shown. The linear fit shown in green is used to estimate the parameters c_1 and c_2. B) Comparison of VST and $\log_2$ transformed values for this array with the green line representing VST = $\log_2$. Figure created using the lumi package. 105

5.12 Here, we show the MA-plots for an array with spikes at concentration 3pM against spikes at concentration 1pM. In the top row, the arrays were transformed with a $\log_2$ transformation or VST. In the bottom row, the arrays were background normalised before transformation. In all plots, red dots mark the values for the spike probes and the dotted lines indicate the predicted log fold-change of spikes (1.73) and non-spikes (0) respectively. 107

5.13 Comparison of spike log-odds obtained for a particular contrast in the linear model fitted to the entire spike-in experiment of 48 arrays. On the left we show the difference between the log-odds obtained after VST and the log-odds obtained after a $\log_2$ transformation. On the right, we show the difference between VST and a linear model incorporating $\log_2$ variances as weights. In the top panels, we show six independent contrasts with the closest spike concentrations. The bottom panel shows six independent contrasts from the same linear model, but chosen to provide a range in anticipated log-ratios (the finer differences being to the right of the panel). In all cases, a positive value indicates greater log-odds obtained (i.e., more evidence for differential expression) after VST. 109

5.14 Comparison of spike log-odds obtained for a particular contrast in the linear model fitted to the 8 arrays involved in that contrast. On the left we show the difference between the log-odds obtained after VST and the log-odds obtained after a $\log_2$ transformation. On the right, we show the difference between VST and a linear model incorporating $\log_2$ variances as weights. In the top panels, we show six independent contrasts with the closest spike concentrations. The bottom panel shows six independent contrasts chosen to provide a range in anticipated log-ratios (the finer differences being to the right of the panel). In all cases, a positive value indicates greater log-odds obtained (i.e., more evidence for differential expression) after VST. 110

6.1 Two different views of non-normalised Illumina data from the MAQC study. In A, we show data from one location. One array was removed during quality assessment by the MAQC so in total there are 19 arrays with 48,000 observations each. In B, the same arrays are divided into the probes found on Strip 1 and Strip 2. Similar boxplots were seen for all three locations. 119
6.2 For different filtering methods (see Section 6.2.3) we show the percentage of genes belonging to Strip 1 and Strip 2 and different annotation categories retained by the filter. 121
6.3 Average ranks of probes across all Illumina Human6 arrays in GEO. A) shows the average ranks of probes on Strip 1 versus probes on Strip 2. Probes on Strip 1 (roughly equivalent to the lower-density Human8 arrays) are found to be ranked higher than those on Strip 2. In B) the probes on Strip 1 are split into different annotation categories. 122
6.4 Representing the composition of ranked lists of genes obtained from the MAQC dataset after a differential expression analysis. Separate curves are shown for each pairwise contrast of samples. For the linear model fitted to all probes (Fit 1), we show the proportion of Strip 1 or Strip 2 probes encountered along the length of the gene list. For the linear model fitted to probes from Strip 1 only (Fit 2), we show the composition of the gene list in terms of the annotation categories. 126
6.5 Improvements to the detection of DE genes by filtering. On the left, the number of significant findings (after multiple testing correction) for Strip 1 probes using Fit 1 and Fit 2 are shown. For all the contrasts, the number of significant probes is greater using Fit 2. On the right, the number of probes with good annotation is shown under Fit 1 and Fit 2. Fit 3 is also seen to increase the number of significant findings. 128
6.6 Screenshots of the BLAT search for probe GI_42657060-S. The top screen just some of the locations that the probe matched to. The bottom screen shows the genomic location of the top match. The probe sequence is seen to be outside any RefSeq genes and matching a region of the genome with a long-term repeat item. 131

6.7 The rank of probe GI_42657060-S across all Illumina Human6 Version 1 arrays in GEO. It can be seen that regardless of sample or processing method, the probe is generally among the highest ranked probes on an array. The different colours represent the different datasets. 132

C.1 Modified TIFF images for arrays 1, 3 and 6 from the BioC07 dataset. 145

List of abbreviations

- BGN - Background normalisation used by Illumina to calibrate arrays to the same baseline.
- BLAT - Basic Local Alignment Tool.
- cDNA - Complementary DNA
- DE - Differentially expressed.
- DNA - Deoxyribonucleic acid
- Human6 - A chip manufactured by Illumina to interrogate 48,000 genes on 6 human samples.
- Human8 - A chip manufactured by Illumina to interrogate 22,000 genes on 8 human samples.
- limma - Linear models for microarrays `R` package
- Mouse6 - A chip manufactured by Illumina to interrogate 48,000 genes on 6 mouse samples.
- mRNA - Messenger ribonucleic acid
- normexp - Normal-exponential convolution method used for background correction.
- RefSeq - Reference sequence database.
- SAM - Sentrix Array Matrx format used by Illumina that has 96 arrays in a 8 × 12 matrix.
- QA - Quality assessment.
- VST - Variance stabilising transformation

Chapter 1

Introduction

This chapter gives a brief introduction to the use of microarrays for medical research and motivates the need for statistical and computational tools to deal with the vast amounts of data generated by such devices. I will also introduce the technology behind bead-based microarrays, which are the subject of investigation in this thesis.

1.1 Overview of DNA and RNA

Deoxyribonucleic acid (DNA) encodes the information required for the development and function of a living organism. The structure of DNA is remarkably simple, being formed of a long chain of smaller units (nucleotides) joined together. Each nucleotide can have one of four bases attached; Adenine (A), Cytosine (C) Thymine (T) or Guanine (G). The order in which these bases occur in a DNA molecule is known as the DNA sequence.

Structurally, a DNA molecule takes the form of a double helix formed by two strands of DNA. This structure is held together by strong hydrogen bonds between the two strands. The bonding takes place in such a way that A base-pairs with a T base in the opposite strand, whereas C base-pairs with G. This is known as the base complementarity property of DNA and effectively means that the sequence of one strand can be determined by the other, a fact that is exploited during DNA replication.

The entire DNA sequence of an organism is known as its genome. The human genome is estimated to have 3.3 billion bases and can be found in the nucleus of every cell in the body. Rather than being one long DNA molecule, a genome is divided into continuous regions of DNA known as chromosomes. There are 24 chromosomes in the human genome, which are numbered 1 to 22 plus the sex chromosomes, X and Y. Most "healthy" cells in the human body contain 46 chromosomes; two copies of chromosomes 1 to 22 (one copy of the chromosome inherited from either parent) and either an X and Y chromosome (for males) or two copies of the X chromosome (for females). Each chromosome is divided into stretches of DNA called genes, which encode the instructions to produce a particular protein. The estimated number of genes in the entire human genome is between 25,000 and 30,000. However, rather than being one continuous sequence of genes, there are many gaps in the chromosome that are not genes and hence do not code for proteins. In fact, only an estimated $5 - 10\%$ of our genome is used to code for proteins. The purpose of remaining "junk DNA" is a source of much debate, but increasingly is considered to have regulatory function.

The instructions encoded in the DNA sequence are stored in the nucleus and must be transported to the cytoplasm, where specialised molecules called ribosomes help produce the required proteins. However, the DNA sequence itself is too valuable to be transported. Therefore, sections of DNA are copied (transcribed) into temporary *messenger RNA* (mRNA) molecules that contain the same information as DNA, but in a slightly different form. The main differences are that mRNA is single-stranded, degrades quickly and has a Uracil (U) base instead of a T. The entire sequence of each gene is transcribed, even though not all parts of the sequence take part in coding for proteins. Such non-coding regions, known as introns, are removed by splicing before the process of translation starts. Translation uses mRNA as a template to assemble previously synthesised amino acids in the correct order to make particular proteins, with groups of 3 successive bases used to specify the amino acid located in that position in the chain.

Although each cell contains copies of the same genome, the cell will require different amounts and combinations of proteins in order to perform its function within the body. Therefore, the genes that control the production of these proteins may be turned on or off to varying degrees. These changes confer unique properties to each cell type. The expression level of a gene refers to the amount of mRNA that is made from the DNA template and subsequently translated into protein.

Given that the genome contains the complete set of instructions required to develop and maintain a living organism, it is little wonder that medical research has invested heavily in methods for studying the genome, and in particular the regulation of gene expression. Being able to understand the differences between healthy and diseased cells, and the mechanisms that bring about these differences is of chief importance. For example, the growth of a cell is tightly regulated by proto-oncogenes which keep a cell dividing and growing, whilst tumour-suppressor genes bring about the death of a cell when required. Clearly, disruptions to the normal activity of these genes could have serious implications for the development of cells, with diseases such as cancer associated with cells that have grown out of control.

In the next section, I describe a popular experimental technique for determining the expression level of a large set of genes in a given sample. The data generated by these experiments require careful processing and statistical analysis in order to draw valid biological conclusions. These issues will also be addressed later in this chapter.

1.2 Gene expression microarrays

A microarray (sometimes referred to as an *array*) is a device for simultaneously measuring the expression level of thousands of genes. The technology makes use of the base-complementarity property of DNA and the fact that single-stranded mRNA is produced in order for a particular gene to be expressed. Thus, by measuring the amount of mRNA we can infer the expression level of the gene.

Microarrays are typically constructed by attaching single-stranded DNA sequences, known as *probes*, to a surface such as a glass slide. Each probe is complementary to the DNA sequence of a particular gene of interest and is placed in spots (or features) at pre-defined locations. Single-stranded mRNA from a sample of interest (called the *target*) is isolated, converted into single-stranded DNA (cDNA) and then transcribed into cRNA. These cRNA are then fluorescently labelled, and exposed to the microarray surface. The target RNA then binds (hybridises) to its complementary probe sequence on the microarray, whereas non-complementary sequences should fail to hybridise. The amount of fluorescence observed at each feature can therefore be used to determine the level of expression for each of the genes represented on the array. In the earliest microarrays (Schena *et al.*, 1995), each feature on the array corresponded to a different gene of interest. However, subsequent developments in microarray production have allowed the same gene to be represented multiple times, thus providing more reliable expression estimates.

Two-colour microarrays are used to compare two samples (e.g. cancer and normal cells) on the same microarray. The RNA from the two samples is extracted separately and fluorescently labelled with different dyes, usually red and green. Therefore, after hybridisation, each feature is a mixture of red and green fluorescence. A completely red or green feature indicates that a particular gene is expressed in one sample, but not the other. In practice, the mixture of red and green observed at each feature is not so clear-cut and statistical methods are required to quantify the contribution of each colour, as described later. A "differential expression" analysis aims to find genes that have significantly different expression levels between different conditions under investigation. Such genes are said to be differentially expressed (DE). See Figure 1.1 for an illustration of a typical two-colour microarray experiment.

Microarrays have become an invaluable tool for medical research (Allison *et al.*, 2006) and provide a wealth of data that was previously unobtainable. The production of microarrays is a rapidly growing industry, with many companies supplying variations of the technology for a wide range of applications. Each company has a different method of manufacturing microarrays, the major differences being the production of the probe sequences used and the method of depositing these sequences onto the array surface. For instance, different length probe sequences (usually measured in the number of base-pairs) can be used as well as mRNA or cDNA probes, rather than the cRNA probes described above.

Single-channel microrrays can also be produced to measure the absolute expression level of every gene of interest in a given sample. Therefore, the fluorescence of each feature is a measure of the expression level of a particular gene. Until recently, arguably the most popular single-channel microarray technology was that of Affymetrix (Lockhart *et al.*, 1996). These arrays

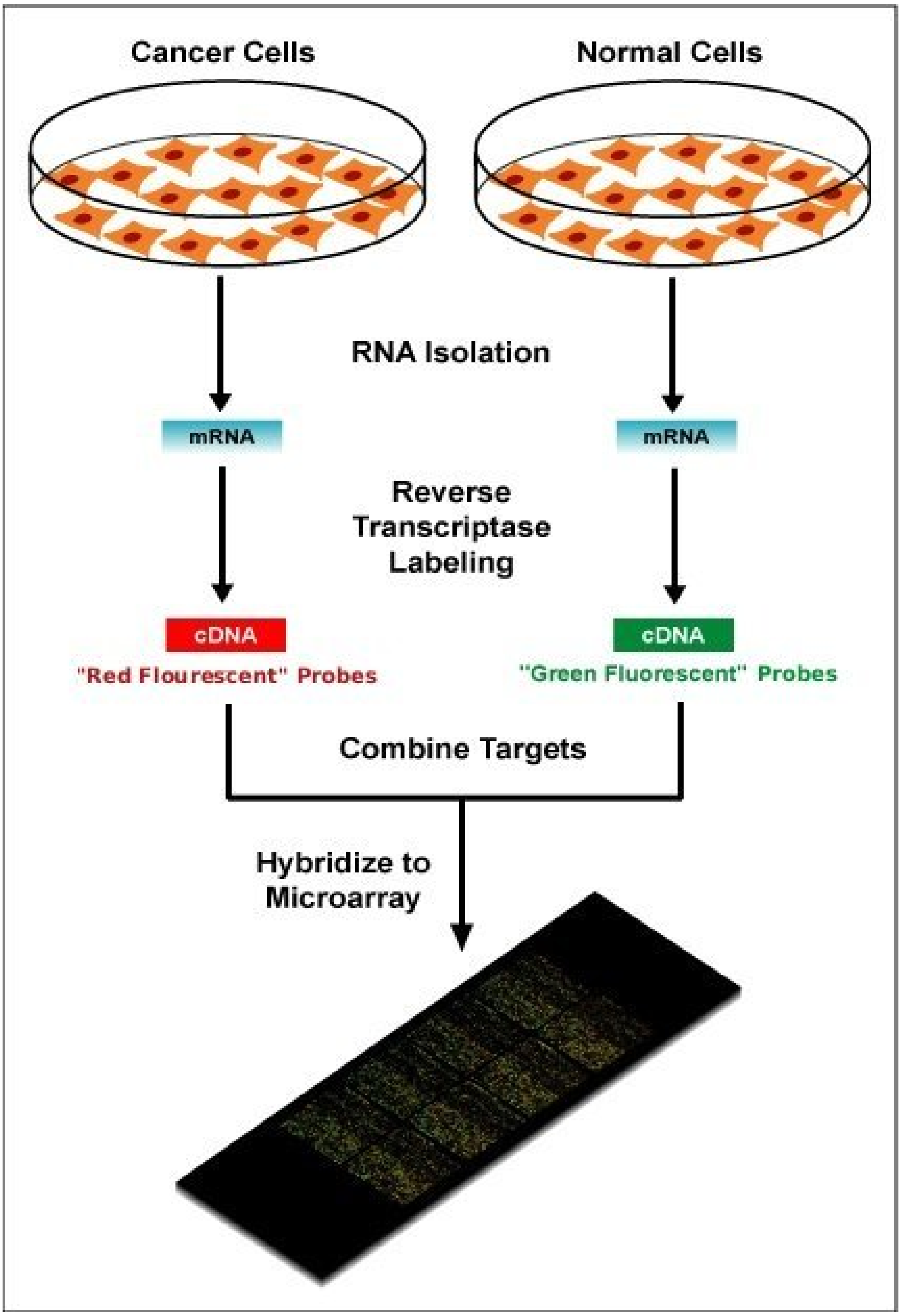


Figure 1.1: This public domain image shows a schematic diagram of a typical two-colour microarray experiment to compare DNA from a cancer cell to that of a normal cell.

use 25 base-pair probes that are synthesised on the array surface. Each gene of interest is interrogated by a collection of 11-20 probe pairs, known as a *probe set*. The expression level for a gene is then derived by combining all measurements from a particular probe set.

Additionally, microarrays are manufactured for applications other than gene expression. For instance, microarrays can be used to interrogate regions of the genome where differences in a single base (Single Nucleotide Polymorphisms, or SNPs for short) are observed in a population, or regions where long stretches of DNA are gained or lost (Copy Number Variation or CNV). Adaptations of these technologies can also investigate changes to the genome, such as methylation, that alter the structure of DNA but not the sequence, and the locations where proteins might bind to the genome in order to encourage or impede expression

1.3 Illumina bead-based microarrays

In this thesis, I will concentrate on the BeadArray microarray technology developed by Illumina, which is becoming widely used and offers many potential benefits over other technologies. Rather than attaching probes onto a microarray at known locations, BeadArrays are self-assembling arrays of minute beads with probes attached. Each array is produced separately by exposing an array surface (either a glass slide or fibre-optic bundle) to a large collection of pre-prepared beads. This causes the beads to be randomly sampled and assembled into wells on the surface of the array (Fan *et al.*, 2006). A specific DNA sequence is assigned to each *bead type*, which is replicated on about 30 beads on an array. Each bead is 3 microns in diameter and has many thousands of copies of the same probe sequence. Both the number and location of the replicates for the same bead type are random on an array (Kuhn *et al.*, 2004). Therefore, an extra address sequence (an *IllumiCode*) is attached to each bead for decoding (Gunderson *et al.*, 2004), with beads of the same type also having the same IllumiCode. Each IllumiCode is designed to hybridise in a predictable way to a series of specially designed dye-labelled sequences. After each hybridisation, each bead is assigned to one of two states (e.g. red or green) depending on the amount of hybridisation. Thus, after a number of such hybridisations, a binary sequence is determined for each bead. This binary sequence should then uniquely correspond to the predicted response of an IllumiCode. These decoding hybrisidations are performed by Illumina, with the guarantee that no array will be supplied to the user with a bead type that has less than five replicates.

Along with the high degree of replication within an array, Illumina also offer the capability of processing BeadArrays in parallel, making this technology desirable for high-throughput experiments. A Sentrix BeadChip is a glass slide (chip) that allows a very high number of observations to be made for a particular sample. Depending on the configuration of the chip, between 1 and 16 samples can be processed simultaneously with tens of thousands of genes profiled per sample. A more detailed description of this chip technology is given in Chapter 2. The Sentrix Array Matrix (SAM) contains 96 arrays, each of which is a hexagonal fibre-optic bundle with approximately 50,000 beads and around 1,500 distinct bead types. Thus, 96 samples can be interrogated simultaneously on a single SAM. See Figure 1.2 for a summary of how these arrays are constructed.

1.4 Pre-processing and analysis of microarray data

Despite differences in array production, the common goals of any gene expression study are roughly the same and one has to deal with similar statistical issues when analysing microarray data. For instance, the intensities of the features on a microarray are influenced by many sources of noise and repeated measurements made on different microarrays may also appear to disagree. Therefore, a number of data-cleaning, or pre-processing steps, must take place before being able to draw valid biological conclusions from a microarray experiment (Quackenbush, 2002; Smyth *et al.*, 2003; Allison *et al.*, 2006).

These steps are well-understood for established microarray technologies (e.g. Affymetrix or older two-colour arrays). However, at the start of my PhD there was little coverage of the processing of Illumina data in the literature. Therefore, the main theme of this thesis is to apply knowledge acquired from

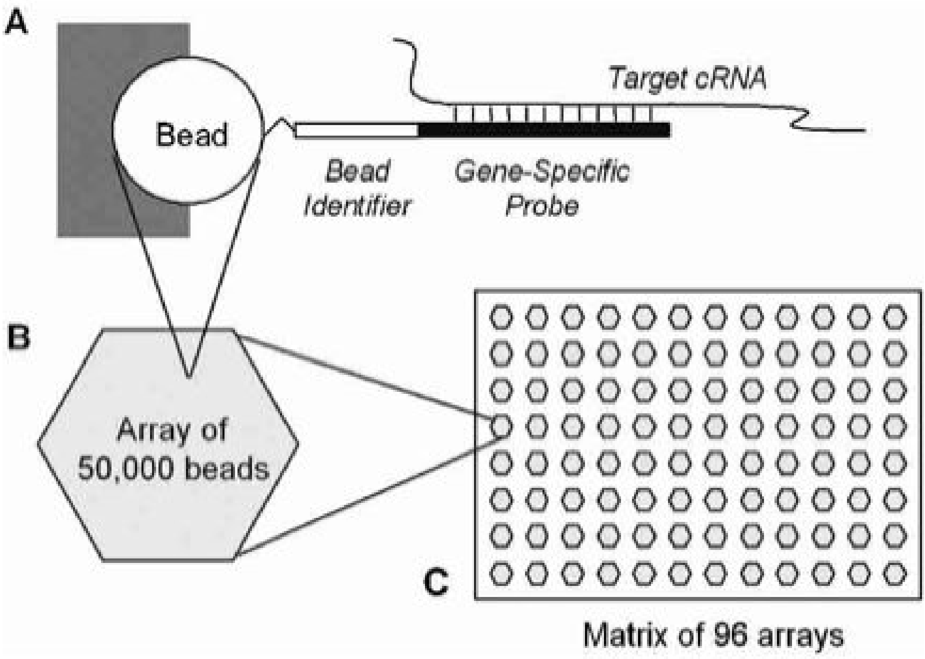


Figure 1.2: Constructing an Illumina "array of arrays", in this case a SAM: **A**) Each bead sits in a pre-created well on the surface of an array and has probes attached that are complementary to a particular genomic sequence of interest. In this figure, only one sequence is shown, although the bead will have thousands of these sequences attached. The bead also has a unique identifier sequence which is used for decoding purposes. **B**) Each array has around 50,000 beads that are randomly arranged. Around 1,500 distinct bead types are represented around 30 times each. **C**) A matrix of 96 arrays is constructed, each array being uniquely prepared and thus having a different arrangement of beads. Image from (Kuhn *et al.*, 2004).

other microarray technologies to the emerging Illumina technology.

1.4.1 Image Capture and Processing

A microarray surface is typically scanned by a laser to produce an image representation of the fluorescence emitted by it. Thus, depending on the resolution of the scanner, each feature will be represented by a number of pixels. For two colour microarrays, separate red and green images are produced. These are known as the raw images and are usually in the 16-bit TIFF image format. Therefore, the intensity of each pixel is a value in the range $0 - [2^{16} - 1]$. These images are usually processed by the manufacturers' software, which involves locating all the features on the image and then calculating foreground intensities using the pixels that make up each feature. However, the pixel intensities measured on the image may be influenced by factors other than hybridisation, such as optical noise from the scanner or foreign items deposited on the array. Therefore, a background intensity is estimated for each feature to account for such factors. The background and foreground estimates generally act as a starting point for statistical analysis.

1.4.2 Background Correction

The aim of background correction is to reduce the impact of non-specific or random contributions to the observed intensity for each feature on an array. If the foreground (X_f) and background intensities (X_b) of each feature on an array have been obtained via image processing, then the simplest form of background correction to give background corrected intensities (X) is:

$$X = X_f - X_b. \tag{1.1}$$

However, background correction must be applied with care, as the background values X_b are not guaranteed to be greater than the foreground and can yield negative intensities with this simple equation. Such negative intensities become difficult to interpret in further analysis. Potential solutions to this problem are discussed in Chapter 2.

A different approach to background correction is provided by Affymetrix. Each pair in the probe set has one perfect match (PM) probe which is complementary to the gene of interest, and one mismatch (MM) probe which is identical to the PM probe except for one base. The purpose of the MM probes is to measure the background noise of the microarray. The PMs and MMs for each probe set are combined into a single measurement for the gene.

1.4.3 Quality Assessment

Quality assessment (QA) is a crucial part of the analysis process as it can help identify sources of technical variation, and arrays for which the hybridisation failed to work and needs to be repeated. Figure 1.3 shows example QA plots generated using data that accompanies the limma microarray analysis software (Smyth, 2005). The data in question (the “swirl” dataset) compare zebrafish RNA from samples with a mutation in an important gene to RNA from a normal sample.

A typical QA includes *boxplots*, which give a convenient visual representation of the distribution of quantities of interest measured by the array. These can include foreground and background intensities of each feature. For each array, a box is constructed using the 25th, 50th and 75th quantiles. Thus, the length of the box represents the inter-quartile range (IQR). Values that are more than 1.5 IQR above the 75th quantile or 1.5 IQR below the 25th quantile are usually plotted as individual points. When arranged side-by-side, boxplots give a rough guideline of how the overall distributions on each array differ. Figure 1.3A shows boxplots for the red foreground intensities of the swirl dataset after applying a $\log_2$ transformation. This transformation is usually applied for QA plots as it reduces the spread of the data and makes them easier to visualise (Smyth *et al.*, 2003). In this figure we can see the the first two arrays (*swirl.1* and *swirl.2*) have a median intensity around 12, whereas arrays three and four (*swirl.3* and *swirl.4*) have median intensity around 11. Therefore, genes on the first two arrays might be considered more expressed than on arrays three and four. The purpose of QA is to determine whether this difference arises for biological or technical reasons.

MA-plots (Dudoit *et al.*, 2002) are a common visual tool for comparing arrays in a single-channel experiment, or the two channels in a two-colour experiment. For two given samples (k_1 and k_2) the intensities for a given gene, y_{k_1} and y_{k_2}, are used to calculate log-ratios M where

$$M = \log_2(y_{k_1}) - \log_2(y_{k_2}) \tag{1.2}$$

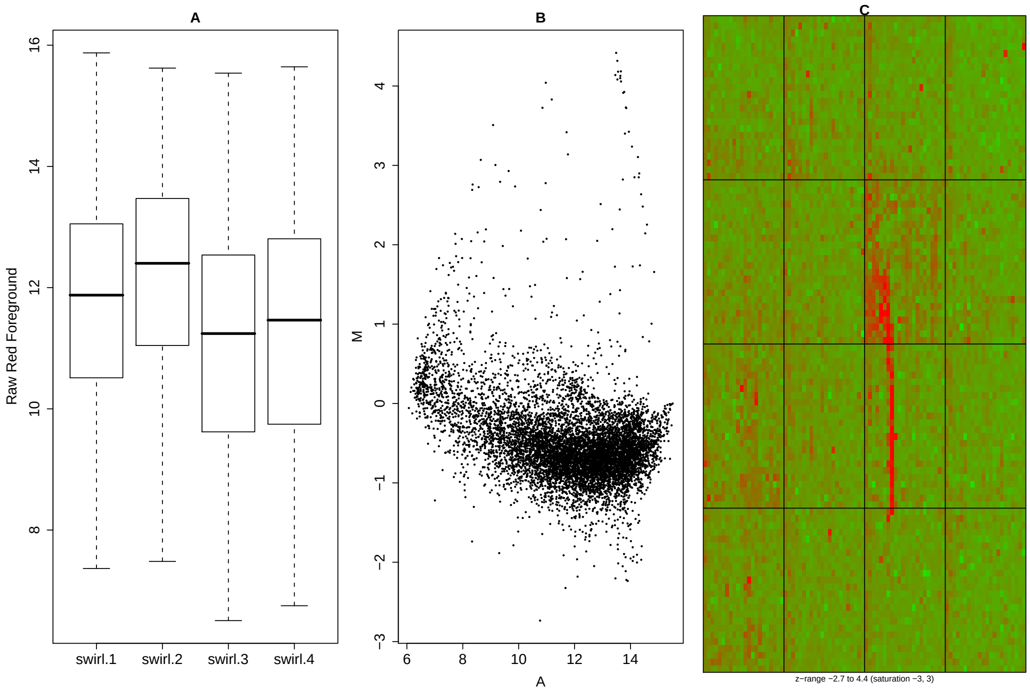


Figure 1.3: QA plots for a two-colour microarray experiment provided with the limma user guide. A) Boxplots of the red foreground for four arrays in the experiment, showing the median values and inter-quartile range of each array. Ideally, the foreground measured on each array should have roughly similar distributions. However, the intensities on the first two arrays are generally higher. B) An MA-plot for the red and green channels for a particular array, with the log-ratio (M) plotted on the y-axis and log-average (A) on the x-axis. We would expect that most probes are not DE, and should therefore be centred along $M = 0$. In this example, most points are away from this line and adjustment is required to remove this trend. C) Imageplot of the log-ratios for a particular array, with green and red representing low and high intensities respectively. Ideally, a random scattering of colours should be seen. However, a red streak is seen in the 3rd column of the array. Spots affected by this artefact have log-ratios that are systematically higher and might not be attributed to differences in the biological conditions being studied.

and average log intensities A, where

$$A = \frac{1}{2}[\log_2(y_{k_1}) + \log_2(y_{k_2})]. \tag{1.3}$$

The M- and A-values for all genes are then plotted on the y and x axes respectively, with a value of M=0 indicating that a gene is not DE between arrays. Generally, it is assumed that most genes are not DE between different samples and therefore most genes should have M-values near to 0. Figure 1.3B shows an MA plot for *swirl.1*, where the M and A values were calculated using intensities from the red and green channels for the array. Most points deviate from the line $M = 0$ and this deviation is seen to increase as the average intensity (A) increases. Such intensity-dependent effects are common for two-colour microarrays and are attributed to different properties of the dyes used to label the samples. Hence this effect is often referred to as dye-bias (Smyth *et al.*, 2003).

False-colour images (know as *imageplots*) can also be used to visualise the intensities of all features on a given array, and to look for trends caused by errors in the manufacturing of the array, rather than by biological differences. Each point on the plot is coloured according to some measurement that we want to compare, such as the foreground, background or log-ratios. Ideally we would like to see a random scattering of colours across the image. Figure 1.3C shows an imageplot for the log-ratios of array *swirl.1*, with green and red indicating low and high intensities respectively. This array was constructed by spotting the probes in pre-defined locations on the array in a grid pattern of four rows and four columns, and each grid cell further divided into 22 rows and 24 columns. On the imageplot, we can see a red streak on the middle rows of the 3rd column. Thus, points inside this so called spatial artefact have a log-ratio that is probably due to manufacturing problem (e.g. scratch or dust on the array surface) rather than biological variation. Microarray manufacturers take many steps to ensure such artefacts do not occur. However, these can also occur during sample processing and an important step in QA is identifying artefacts and ensuring that they do not influence the conclusions drawn from the experiment.

1.4.4 Normalisation

The process of normalisation involves reducing sources of systematic variation and making the intensities observed within the same array, and between arrays in the same experiment, comparable (Smyth and Speed, 2003). For a within-array normalisation, one might correct for any spatial effects detected using imageplots. In the simplest case, a global constant (usually the median or mean of all intensities on the array), may be subtracted from each observed intensity. Or one might take the grid structure of the array into account and normalise all genes in a location-dependent manner.

Between-array normalisation strategies may be used to correct intensity-dependent trends such as that observed in the MA-plot of Figure 1.3. Such methods usually make the assumption that the majority of genes are not DE and correct $M-$values so they are centred around the line $M = 0$. Alternatively, the popular method of quantile normalisation (Bolstad *et al.*, 2003) assures that each array has the same distribution.

1.4.5 Detecting DE genes

One of the main goals of a gene expression experiment is usually to derive a list of DE genes, whose expression level is significantly different between the samples under investigation. Various statistical approaches are applied to the detection of DE genes, and these are described in more detail in Chapter 2. Generally, these involve calculating a test statistic for each gene, and then ordering all statistics according to decreasing significance. A threshold may then be used to identify which genes are most likely to be DE. With such large numbers of genes being tested simultaneously, there is naturally the associated problem of multiple testing and the possibility of many false positive results. Another complication is that the number of repeated observations for a particular gene can be low, making it more difficult to obtain reliable estimates of expression level and the associated variability. Several strategies to overcome this problem are also discussed in Chapter 2.

1.4.6 Further analysis

With the high number of features on a microarray, the researcher can sometimes be given a daunting list of genes declared to be DE. Being able to experimentally validate each result can be expensive and time-consuming, and lead to many blind alleys. Therefore, further analysis of the list of DE genes is becoming increasingly common (Allison *et al.*, 2006). To make sense of a gene list, intuitively we look for similarities among the list. These similarities can be found by considering prior biological knowledge of the genes. Many online resources are available to allow researchers to gain extra insight into their results. For instance, the gene ontology (GO) project (The Gene Ontology Consortium, 2000) attempts to maintain a hierarchical model representing the involvement of genes in biological processes. A particular GO category or term is a set of genes who share a common biological function. The tree-like structure of GO means that each GO term may have a number of "parent" terms that are more general instances of that term. Similarly, each term may have a number of "child" terms that describes a more specific function. The Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000) is another comprehensive database for relating genes with similar function.

Another common analysis technique applied to microarray data is clustering, the purpose of which is to group together arrays or genes that show similar expression patterns and to discover previously unknown similarities (unsupervised) or to categorise according to pre-defined criteria (supervised). A common application for clustering is to discover sets of genes that can be used to classify samples from different patients into pre-defined clinical groups in order to obtain accurate predictions about prognosis and most effective treatment course based on prior knowledge of the groups (van 't Veer *et al.*, 2002).

1.5 Computational challenges

As our knowledge of the human genome improves and the cost of microarray production decreases, microarrays are being produced with increasing numbers of features and studies are being conceived with larger sample sizes. Therefore, there is a great need for computational tools to handle the vast amount of data generated by microarray experiments. From the brief introduction given in this chapter, it should be apparent that there are a number of complex steps involved in converting the scanned images from a microarray into valid biological information. To maxmise the utility of microarray experiments, the steps taken in the analysis should therefore be documented so that they can be validated by external parties if required.

Although many commercial products are available for the analysis of microarray data, the open-source Bioconductor project (Gentleman *et al.*, 2004) remains a popular choice. This is a repository of software written using the R programming language (Ihaka and Gentleman, 1996) for the analysis of microarrays and similar high-throughput technologies. The project promotes transparent, reproducible research and collaboration between the software developers and end-users.

1.6 Thesis outline

This thesis deals with many aspects in the analysis of microarray data obtained using Illumina bead-based technology. Chapter 2 describes the use of this technology for whole genome gene expression studies, where up to 48,000 measurements can be made simultaneously. I will discuss how the Illumina technology potentially overcomes some common issues in the analysis of other microarrays. During a review of publications using Illumina arrays, the wide range of analysis methods applied and reliance on Illumina's proprietary software (BeadStudio) becomes apparent. This software provides a summarised overview of the data and does not utilise some of the unique features of the technology. Also, it does not easily lend itself to integration with existing microarrays analysis tools in Bioconductor.

In Chapter 3, I describe the development of the beadarray open-source software tool that provides access to the data required to perform a full analysis. This software has proved extremely popular and is used by researchers around the world. In Chapter 4, I describe the use of the software on arrays from a pilot study on Illumina technology. This enabled the investigation of several low-level properties of Illumina arrays that were not previously reported. Chapter 5 describes the analysis of a specially designed "spike-in" experiment where the "truth" about a small number of genes was known in advance of the experiment. Thus, the processing methods of Illumina could be investigated in more detail to see how well they recover the predicted results. It is shown that measures presented by BeadStudio are reasonable, and can be used in a successful analysis. However, access to the raw data allows the variability of each gene to be taken into account to derive a more powerful test for differential expression. The method of normalisation implemented in BeadStudio is shown to be incompatible with some current analysis approaches in Bioconductor. Alternatives to this normalisation are discussed and an independently developed method is validated using the spike-in experiment.

Analysis of the probe sequences used in spike-in experiment revealed that around half of 48,000 probes are not optimally designed for their intended genes, thus complicating the conclusions that can be drawn from an analysis. In Chapter 6, I discuss how the identity of such misannotated probes can be predicted and the consequences of not taking this information into account by giving examples from previous studies. It will be shown that this problem can be reduced with appropriate filtering or by using low density chips from Illumina that include only reliable probes.

A list of publications that I have been involved with during the course of this thesis is presented as an appendix.

Chapter 2

Human whole expression profiling using Illumina microarrays

2.1 Introduction

Expression profiling studies aim to ascertain the expression level of many genes simultaneously under different conditions of interest. In a simple case, one might try to find genes that have aberrant differential expression between normal and healthy tissue. In order to avoid biases in the results, it is advisable to make measurements for all possible genes that might be expressed in the samples of interest. This is especially important when we are investigating a system for which we have little prior knowledge. A popular use for microarrays is to look for genes that are associated with a particular disease, in which case testing a large number of samples is advisable to obtain meaningful conclusions. Ideally, microarrays should offer the flexibility to interrogate a large number of genes without compromising on the number of samples that can be investigated.

Typically, the outcome of a gene expression study will be a list of candidate genes that are believed to be associated with the biological differences being studied. Candidate genes may often be tested or validated using further experimental techniques that are time-consuming and expensive (Chuaqui *et al.*, 2002). Therefore a great deal of attention needs to be paid to the results of a microarray experiment and one must be confident of their accuracy in order to minimise the number of false positives taken forward for further analysis.

In this chapter, I give a more detailed description of the microarray analysis steps introduced in Chapter 1 and some problems commonly encountered during the analysis of such data. Then, after describing the technology Illumina use for expression analysis, I discuss the analysis steps they propose and how their technology might overcome some of the issues found for other technologies. Finally, I discuss examples of how Illumina microarrays have been used elsewhere in the literature for gene expression studies. I will first introduce two key studies using older versions of the technology before moving on to studies that performed whole-genome expression profiling on the human genome.

2.1.1 Background Correction

Image processing is an important consideration for conventional microarrays and incorporates locating each feature on the array and estimating the true amount of hybridisation. Locating features, known as segmentation, is usually performed according to the microarray manufacturers' guidelines and typically results in an estimate of the foreground and background signal for each feature. The background estimate is generally obtained from the mean or median of pixel intensities close to the feature. In other words, these values are a local estimate for the underlying array intensity in a particular region of the array. It has been shown that the segmentation method has less impact than background correction (YANG *et al.*, 2002b), and hence I will concentrate on background correction here. The simplest method of background correction, as given by (1.1), is to subtract the background estimate for each feature from the corresponding foreground. However, this method does not guard against the negative values that arise whenever the background intensity is larger than the foreground. Thus, when calculating log-ratios (M) from the background corrected intensities (in the case of two-colour experiments) using (1.2), many values are undefined and missing from further analysis. The proportion of missing values is appreciable (e.g. up to 14% in RITCHIE *et al.* (2007)), and complicates further analysis. Furthermore, background correction is also widely reported to increase the variability of observations, especially at low intensity when $\mathrm{X}_f \approx X_b$. This is often visualised on an MA-

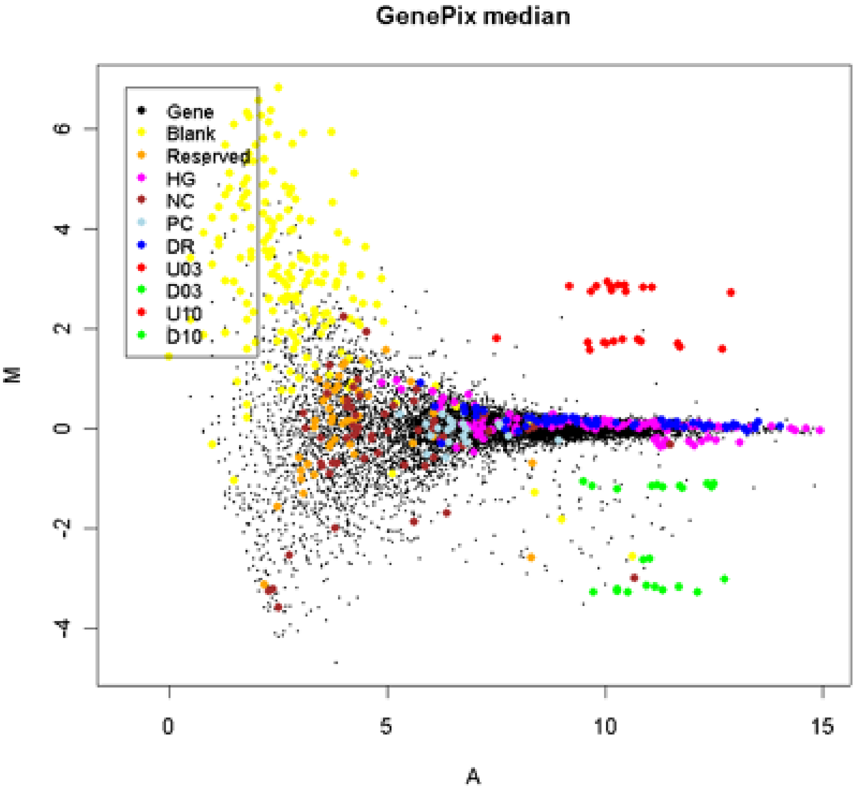


Figure 2.1: Illustrative example of the “fanning effect” for microarray data. This MA-plot compares two replicates of the same biological sample, and hence no differential expression ($M = 0$) should be expected. However, for A< 7, the M-values are seen to have a wide range of values. The coloured spots indicate control probes on the array. Figure courtesy of Gordon Smyth.

plot as a “fanning effect” and has been noted many times in the literature (Ritchie *et al.*, 2007; Yang *et al.*, 2001; Kooperberg *et al.*, 2002). See Figure 2.1 for an illustrative example of this effect.

Given these common problems, it is little wonder that there is no consensus about whether microarray data should be background corrected or not, and the decision is often made on personal preference (Scharpf *et al.*, 2006). Early work in this field demonstrated that background estimates based on the median intensity of pixels surrounding a feature were noisy and increased the standard deviation of the resulting log-ratios (Yang *et al.*, 2001). In fact, the authors recommended avoiding a local background correction, even though in the absence of background correction, the log-ratios produced by this method were biased. An alternative method of background estimation employs a non-linear filter which uses a structuring element to remove all features from the image, thereby generating an image which is effectively background. This morphological filtering approach (*morph*) was found to produce lower, less variable estimates of background in the same study and performed well in the detection of DE genes. However, this method is not commonly available unless implemented in the manufacturers' scanning software and cannot be easily used retrospectively.

The problems of increased variability and missing values have been tackled by developing alternative transformations to $\log_2$ that stabilise the variance across the whole intensity range. Examples include the VSN method of Huber *et al.* (2002) and that of Durbin *et al.* (2002), that can handle negative intensities and give similar results to a log transformation at high intensities. However, these methods assume that negative values have already occurred rather than avoiding them. Alternatively, more sophisticated model-based methods of adjusting for background, rather than just subtracting, have been proposed. These include the method of Kooperberg *et al.* (2002), which assumes the observed intensities have normally distributed foreground and background, and uses an empirical Bayes approach to obtain background adjusted intensities. A simpler approach is offered by Edwards (2002), which uses the standard subtraction for high intensities, whereas for low intensities (defined by a threshold) a smooth monotonic function is used to avoid negative values.

Finally, the *normexp* convolution model introduced by Ritchie *et al.* (2007) was motivated by observing the different distributions of background and foreground, and assuming that observed foreground for each feature (O) is composed of true signal S (that we would like to estimate) and an additive background B. In other words O = S + B, with S being exponentially distributed with mean α and B normally distributed with mean μ and standard deviation σ. The parameters α, μ and σ are estimated on a per-array basis and used to calibrate the observed foreground and background for each feature to obtain an estimate for the true signal.

Ritchie *et al.* (2007) also compared different background correction methods in terms of their variance and bias. The trade-off between these two measures was again found to be an important consideration, as methods which produced the least variable measurements also exhibited the most bias. As with other studies, not background correcting at all gave rise to the most biased but least variable data. However, methods that aim to reduce variance were found to be superior in terms of detecting DE genes in a control experiment. The standard method of background subtraction, as given by (1.1), was found to be much worse than other methods, leading the authors to conclude that this method should be avoided in favour of model-based approaches.

Although Affymetrix use a different approach for background correction, some of the issues encountered are similar. In particular, the MM values were found to have greater intensity than their corresponding PMs in around 1/3 of probes in a control experiment (Irizarry *et al.*, 2003b). Most methods for processing Affymetrix data do not make use of the MM probes, and the popular RMA method uses a convolution model similar to *normexp* to perform background correction (Irizarry *et al.*, 2003b).

2.1.2 Normalisation

The normalisation of microarray data is a crucial step as it aims to correct for biases caused by technical variation inherent in the technology (Smyth and Speed, 2003; Quackenbush, 2002). For two-colour microarrays, the log-ratios M for each gene are often corrected for the dye-bias effect. A simple approach is to correct each array independently and obtain normalised $M-$values of the form $M - c$, where c is a global constant derived from the mean or median of all log-ratios on the array. However, these methods assume that the observed dye-bias is the same for all probes. In practice, the amount of dye-bias is often found to be intensity-dependent, which is manifested as curvature on an MA-plot (see Figure 1.3). If we are willing to make the assumption that most probes should not be DE, (and thus should be located at $M = 0$), then the curve in the MA-plot can be estimated and normalised M-values are given by $M - c(A)$, where $c(A)$ represents the predicted value of the curve for a given A-value. This is the principle behind the popular method of loess normalisation (Yang *et al.*, 2002a). Variations on this method include fitting the curve through just the genes we expect not to change (e.g. any control genes on the array) rather than all genes, or to fit a separate curve to genes within different print-tip groups on the array (Smyth and Speed, 2003).

Normalisation of single-channel arrays aims to make the intensities observed on each array comparable. If the data have been log-transformed, then one can adjust every array separately by subtracting a constant value from all intensities on the array. Thus, we set all arrays to have the same baseline. A common choice of a constant is the median intensity over all arrays. Such methods are easy to implement, but they do not cope well in situations where non-linear variation is observed between arrays (Bolstad *et al.*, 2003). For completeness, it should be noted that if data have been transformed using VSN (Huber *et al.*, 2002), rather than $\log_2$ transformation, they have already been normalised as part of the VSN algorithm.

The loess method used to calibrate log-ratios may be adapted to single-channel data after they have been log-transformed. This cyclic loess approach (Bolstad *et al.*, 2003) uses pairwise combinations of arrays to estimate intensity-dependent effects. However, for large experiments this method can be time-consuming, as for a given array in an experiment with k arrays, we have to define transformations with $k - 1$ arrays.

Arguably the most popular approach is quantile normalisation (Bolstad *et al.*, 2003), which adjusts all arrays to have the same intensity distribution. It works by ranking the intensities on each array separately and then calculating the average across all arrays for each rank to form a "target distribution". The ranked intensities for a particular array are then re-assigned to the corresponding value on the target array, so the highest intensity gene on the array is assigned the average of all the highest intensity genes etc. The method was shown to perform favourably compared to other normalisation methods for Affymetrix data (Bolstad *et al.*, 2003). An obvious drawback of quantile normalisation is the assumption that all distributions should be the same. This may not be realised in practice, especially for experiments involving multiple biological sources that could have completely distributions. Therefore potentially interesting biological variation could be removed by forcing all arrays to have the same distribution.

An alternative to the "global normalisation" approaches of quantile and cyclic-loess are base-line methods which calibrate each array to a defined target distribution using a smooth curve. For instance, *qspline* normalisation (Workman *et al.*, 2002) defines a target distribution to be the average quantiles across all arrays. In practice, the method is not too different from quantile normalisation. An alternative is to base the normalising curves on "rank-invariant" genes, whose expression levels do not change appreciably across the experiment (Li and Wong, 2001).

2.1.3 Detecting DE genes

One of the main goals of any gene expression study is to identify which genes show evidence for being DE. If we have measured n log-ratios for the gth gene over multiple arrays, we might be tempted to use the average log-ratio, $\overline{M}_g$, as our statistic to find genes with most evidence for differential expression. These log-ratios could be obtained from the two channels in a two-colour experiment, or different arrays in a single-channel experiment. However, the log-ratio turns out to be a poor choice for detecting differential expression, as large values of $\overline{M}$ could be driven by outliers in the n observations, especially if n is small. Therefore, the variability of each gene should be taken into consideration. Using s_g, the standard deviation of the log-ratios for the gth gene, we can calculate a t-statistic to test the null hypothesis of $M = 0$, or no differential expression:

$$t_g = \frac{\overline{M}_g}{s_g/\sqrt{n}}. \tag{2.1}$$

However, this statistic is not without its problems either (Lonnstedt and Speed, 2002; Cui and Churchill, 2003; Tusher *et al.*, 2001). Most notably, a large value of t_g could be caused by a small s_g, even though the average log-ratio is low. With such a large number of genes being tested on an array, such genes with low standard deviation can occur by chance and be incorrectly called as DE. Therefore, a method is required that does not rely solely on the gene-specific variances, which may be unreliable in experiments with few replicates. Proposed solutions "borrow information" from all the genes on the array to reduce the impact of genes with extreme variances. The resulting t-statistics use a modified value for the variance of gene g in the denominator. For instance, the "significance analysis of microarrays" method

(Tusher *et al.*, 2001) adds a small positive constant to the denominator of a gene-specific t-test. In this thesis, I will mainly use the linear modeling approach of Smyth (2004) to detect DE genes. Up to now, I have described testing each gene for differential expression between two samples using log-ratios. However, with a linear modeling approach it is possible to test many such comparisons simultaneously.

We define $\boldsymbol{y}_g = (y_{g_1}, ..., y_{g_J})$ as the response vector of expression values for gene g measured on J arrays. These can either be log-ratios in the case of a two-colour experiment, or the intensities of each array in a single-channel experiment. It is expected that these values have been appropriately background corrected and normalised. We then assume the following linear model:

$$\mathrm{E}[\boldsymbol{y}_{\mathrm{g}}] = \boldsymbol{X}\boldsymbol{\alpha}_{\mathrm{g}} \tag{2.2}$$

where X is a design matrix with J columns that provides a representation of the different samples hybridised to the J arrays and $\boldsymbol{\alpha}_g$ is a vector of coefficients to be estimated for each gene. Along with the estimated coefficients obtained by least-squares, the sample variance s_g^2 and residual degrees of freedom f_g are obtained. If this model is fitted to the log-ratios derived from a two-channel experiment, then the coefficients are the quantities that can be tested for differential expression. In the case of single-channel arrays, a contrast matrix ($\boldsymbol{C}$) allows the coefficients from the design matrix to be compared to give contrasts of interest $\boldsymbol{\beta}_g$. Usually these contrasts are analogous to the log-ratios obtained from a two-colour experiment,

$$\boldsymbol{\beta}_g = \boldsymbol{C}^T\boldsymbol{\alpha}_g. \tag{2.3}$$

An unscaled standard deviation for each contrast (u_{gk}) is calculated and may be used in conjunction with the s_g to calculate a t-statistic, as in (2.1). Alternatively, an empirical Bayes approach can be used to estimate new sample variances $\tilde{s_g}$ by using information from all the s_g. The moderated t-statistic for contrast k of gene g is then given by

$$\tilde{t}_{gk} = \frac{\hat{\beta}}{u_{gk}\tilde{s_g}}. \tag{2.4}$$

Along with the moderated t-statistic for each gene, an associated log-odds statistic is computed that gives that the posterior log-odds of a given gene being differentially expressed. However, these values are dependant upon an assumption about the overall number of genes expected to be DE and are more commonly used to rank genes according to their evidence for being DE.

The presence of poor quality replicate observations in the linear model can affect our ability to detect DE genes. One solution to this problem would be to remove these arrays from the analysis completely. However, it is possible to adapt this linear model approach to incorporate weights representing our confidence in the observations used in the linear model, instead of all observations having equal influence (Ritchie *et al.*, 2006). This is done by assuming the gene-specific variances are of the form:

$$\mathrm{var}(y_{gj}) = \sigma_g^2 / w_{gj} \tag{2.5}$$

where w_{gj} is a weighting factor for gene g on array j. The linear model to estimate coefficients is then fitted using weighted least squares and contrasts are calculated as above.

2.1.4 Filtering

Microarrays provide many thousands of gene measurements simultaneously, and therefore the problem of multiple-testing becomes great. That is, with every statistical test performed, there is an associated probability that we will falsely choose a gene as being significantly DE in the study. Moreover, not all genes will be expressed in a particular tissue (Su *et al.*, 2002). Therefore, applying filtering methods to microarray data is advocated in order to remove genes that we do not believe to be informative and are likely false positives. This is especially important if using an ordinary t-statistic with few replicates, where genes with low expression level and extremely low standard deviation might be called DE. However, it is recommended that the filtering criteria is statistically independent of the test statistic to be used.

However, choosing a filtering method is difficult and often derives from adhoc criteria based on expression level and variability. For example, Scholtens and Heydebreck (2005) restrict their analysis of a published acute lymphoblastic leukaemia (the ALL dataset) to probes that have intensity greater than 100 units in at least 25% of samples and also have an interquartile range (IQR) of at least 0.5 on the $\log_2$ scale. Obviously these cut-offs will remove different numbers of genes in different experiments and may not always be appropriate. The code used to perform these, and other so-called "non-specific" filters, are available in the genefilter Bioconductor package (GENTLEMAN *et al.*, 2008). Alternative methods of filtering that use "Present / Absent" calls or between-array variability have been shown to be effective in reducing false positives for Affymetrix arrays (MCCLINTICK and EDENBERG, 2006; CALZA *et al.*, 2007).

2.1.5 Probe Annotation

The annotation of a microarray refers to the mapping of the probe sequences to the genome being studied. As the size of probes used for microarrays are small in comparison to the size of the sequences of genes they are intended to target, the design of probes is a crucial step in the analysis. Also, the speed at which knowledge about the human genome is being gathered means that probe sequences created for microarrays may not reflect the current knowledge of the genome. Problems that might arise include the probe matching more than one gene, which can lead to non-specific hybridisation, or not matching to the intended gene at all.

For Affymetrix data, it has been shown that the base composition of the 25-base probes can have an effect on the observed intensity and methods have been proposed to deal with this effect (WU and IRIZARRY, 2005). Other probe effects have also been described for two-colour microarrays (LYNCH *et al.*, 2007). Additionally, the reliability of Affymetrix probes has been called into question, with a large percentage of probes on an array sometimes not mapping to the intended transcript, which can lead to misleading conclusions in a differential expression study (DAI *et al.*, 2005; HARBIG *et al.*, 2005). By considering the latest version of the genome and re-assigning probes appropriately, the results of a differential expression analysis have been found to change by $30-50\%$ and result in more precise expression measurements (DAI *et al.*, 2005; GAUTIER *et al.*, 2004b; SANDBERG and LARSSON, 2007). The problems caused by poor probe annotation are seen regardless of the processing methods applied to the data. It is therefore of fundamental importance that the annotation of a microarray be routinely checked as part of an analysis.

2.2 Illumina BeadChip arrays

Illumina's whole-genome expression arrays use the BeadChip technology described on page 6, with each chip configured to interrogate 6 or 8 samples simultaneously (the Human6 and Human8 respectively) using either 12 or 8 strips on the chip surface. Each strip has around 22,000 or 24,000 bead types. In this chapter, I discuss the use of these chips for expression studies with the human genome. The Human8 chip is designed to have bead types that target genes from the Reference Sequence (RefSeq) database (Pruitt *et al.*, 2007). This public database funded by the National Center for Biotechnology (NCBI) aims to provide a collection of non-redundant sequences for different transcripts and proteins for an organism of interest. As all entries in the database are curated, it is widely recognised as a gold-standard for reliable genome annotation. The Human6 chip contains all the bead types used on the Human8 chip, plus additional content from other public databases that are not as well curated as RefSeq.

Human6 chips distributed prior to 2006 (Human6 Version 1) had the RefSeq and additional content on adjacent, physically separated, strips on the chip surface. Unless stated otherwise, the human data discussed in this thesis are from Version 1 chips. On newer versions of the chip (versions 2 and 3), the probe sequences were revised and the chip design altered so that all bead types can be found on both strips.

In addition to sequences that target genes, Illumina also add a number of control probes to each array. These are designed so that their behaviour can be predicted and used for QA or normalisation purposes. For example, a series of negative controls (around 1,500 on Human6 arrays) have been designed so that they have no target in the genome. When used in an expression experiment, they should not hybridise to the target sample. Ideally, these bead types should produce no signal and any intensity observed should be measuring background noise. Thus, they perform a similar role to the MM probes for Affymetrix, except the negative controls are not specific to particular bead types. A number of positive controls are also included that should produce signal on every array, regardless of the sample hybridised.

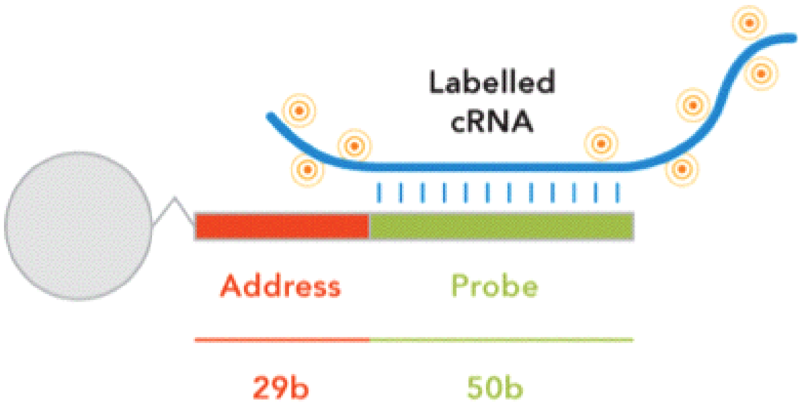


Figure 2.2: A cartoon representation of a bead in the direct hybridisation assay. Each bead has an address sequence attached for decoding purposes and a 50 base sequence specific to each bead type. The probe sequence is fluorescently labelled and designed to hybridise to a particular genomic sequence. Note that the diagram is not shown to scale and many thousands of sequences are attached to each bead. Figure courtesy of Illumina.

2.2.1 Direct hybridisation assay

The direct hybridisation assay is a single colour assay used in conjunction with Illumina's gene expression BeadChips. When these chips are supplied to the user, the decoding hybridisations described in Gunderson *et al.* (2004) have been performed to identify the beads. This includes a quality control step to ensure all arrays have at least 5 replicates of each bead type. As described on page 6, each array has a random configuration of beads, with each bead type (in this case, representing a particular gene of interest) having around 30 replicates. Figure 2.2 shows a cartoon diagram of a bead to be used for the direct hybridisation assay. Directly attached to the decoding sequence is a 50 base-pair sequence designed by Illumina to be complementary to a particular gene or control. As for conventional microarrays, mRNA from the target sample is converted into cRNA in a two-step process of reverse transcription and then in-vitro transcription (IVT). After the IVT reaction, which also includes an amplification of the transcripts, the sequences are labelled with biotin and allowed to hybridise onto the pre-prepared array. The chips are then stained with fluorescently labelled streptavidin, which forms strong bonds with biotin present on the array.

2.2.2 Illumina scanning and analysis software

The results of decoding are supplied to the user in the form of proprietary (i.e., cannot be viewed without specialist software) dmap files. After hybridisation and washing according the particular assay being used, BeadArrays are scanned using hardware controlled by the BeadScan software. This automatically handles the image processing of the raw images, including extraction of foreground and background intensities for each bead, and background correction. In order to extract intensities, the software also locates each bead on the array in a procedure known as registration (Gallinsky, 2003).

The following files are produced as standard output by BeadScan.

- Intensity data (.idat). These are proprietary files used by Illumina to store intensity data.
- Location information (.locs). Proprietary files giving the locations of all beads on an array.
- Image files (.jpg or .TIFF). The image produced by scanning in compressed or uncompressed format, respectively.
- XML files. Information about the scanning settings used for each array and the algorithm used to extract intensities.
- Metrics.txt file. Overall summary of the scanning quality of a chip. Quality control scores between 0 and 1 are given for each array to judge how well each array was registered and focused. The 5th and 95th quantiles of foreground intensity are also given.
- .txt file. Plain text file giving the identity, location and intensity of each bead on the array.

Note that the list of files produced, and their contents, have changed a few times since Illumina first started to produce arrays. In particular, the

.txt files have only recently been produced as standard - and even then it is only possible with modifications to the BeadScan software. We refer to the collection of TIFF images and text files as the *bead-level data* for an experiment. Each image has a corresponding text file, which gives the coordinates of each bead on the image, a background corrected intensity and a numeric code (ProbeID) identifying the bead type. In the case of two-channel arrays, separate text files are given for the red and green channels. However, a key point to note about bead-level data is that they can only be generated at the time of scanning. I have spent a large amount of time publicising the availability of bead-level data in both publications and presentations.

Illumina also supply the *BeadStudio* software to analyse their data. The role of BeadStudio is to read the idat and locs files for each array separately and create *bead-summary data*. The algorithm starts by taking the individual bead intensities (after background correction) and then excluding any beads that are more than 3 median absolute deviations (MADs) from the median of all replicate observations of the bead type. The remaining observations for a bead type are then used to calculate a mean expression level and corresponding standard error, both of which are presented on the unlogged scale.

2.3 Illumina analysis methods

I will now describe the algorithms employed by Illumina in the BeadScan and BeadStudio software. The methods employed by BeadScan, and creation of bead-summary data, are not under the control of the user and the analysis done in BeadStudio is done on a per-array basis with summarised data as described above. Furthermore, the image analysis and background correction methods to be described are applied to all types of Illumina data, whereas the methods I describe within BeadStudio are specific to the analysis of expression data. Although I have worked with other types of Illumina data, the analysis of such data will not be presented in this thesis.

2.3.1 Image processing and background correction

The foreground estimation algorithm used by Illumina is a two-step process described in more detail in Kuhn *et al.* (2004). In brief, these steps are:

i) All pixel intensities are altered using a sharpening transformation. The intensity of a particular pixel is made higher (lower) if its intensity is higher (lower) in comparison to the intensities of the pixels surrounding it.

ii) Foreground intensities are calculated as a weighted average of signals obtained using the four pixels nearest to each bead centre as a "virtual bead centre". Sharpened pixel intensities are used in the calculation and the value returned is unlogged.

Background intensities are estimated using an average of the five dimmest pixels (unsharpened intensities) within the 17×17 pixel area around each bead centre. Background corrected intensities are then calculated by subtracting the background estimate from the foreground, as in (1.1).

2.3.2 Normalisation

BeadStudio provides several normalisation options. All of these options use *background normalisation* (BGN) as a first step to set the intensities on each array to the same baseline. For a given array, this method involves subtracting the mean intensity of the negative controls from each mean expression value. This is intended to compensate for differences between arrays in both non-specific binding of dye and cross-hybridisation. After BGN, an additional normalisation can be carried out, the simplest being average normalisation (AN) which scales the intensities of each array separately so that the mean of each array is the same. Quantile normalisation is supported in later versions of BeadStudio (version 3 and above), whilst a variation of quantile normalisation using cubic splines can also be carried out (CSN). Finally, the rank invariant method (RIN) calibrates each array using the intensities of genes whose overall rank does not vary greatly in the experiment. It is also possible to analyse data in BeadStudio without applying any normalisation, although BGN is performed by default for many analyses.

2.3.3 Filtering

The bead-summary data for a given array include a detection p-value for each bead type. This is calculated using the relative rank (R) of a given type

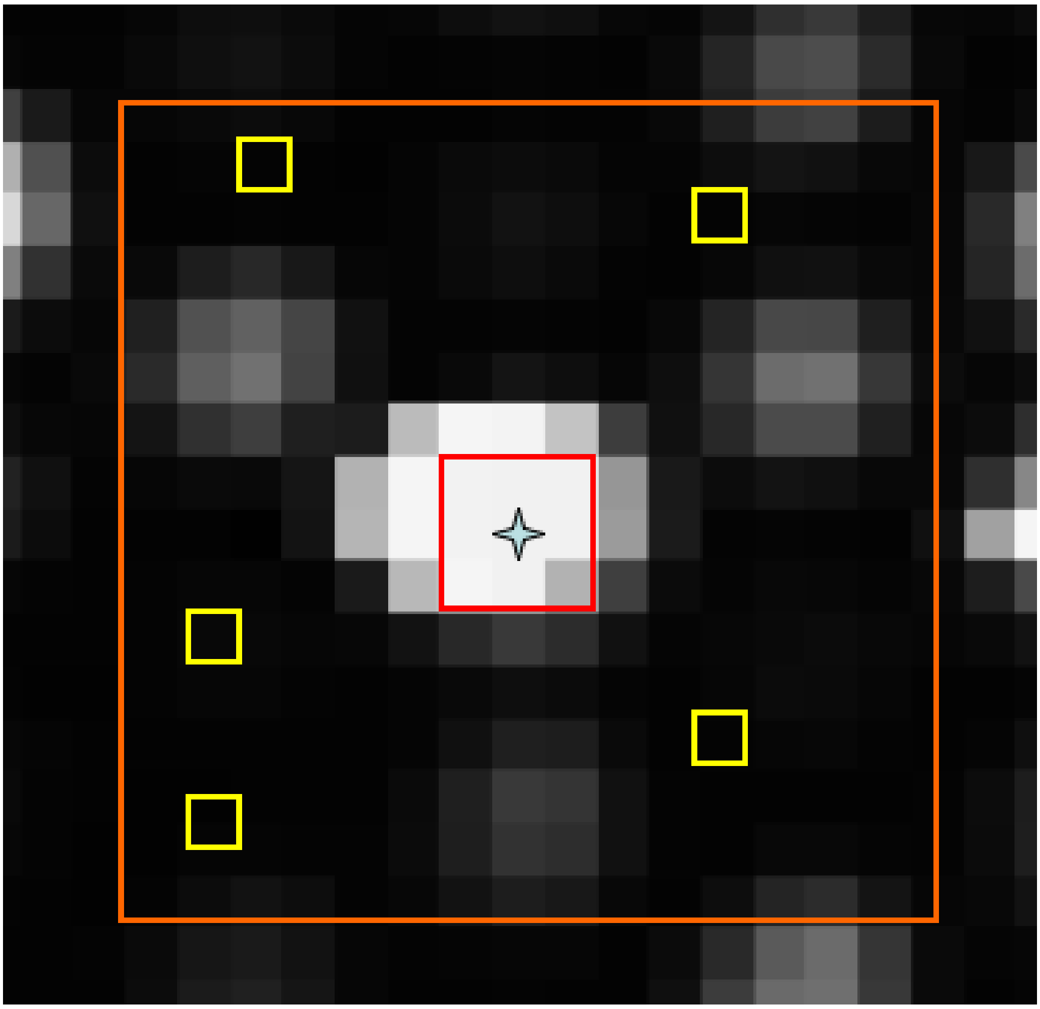


Figure 2.3: Diagram of the steps used for calculating the foreground and background of a bead. Using the coordinates determined during the decoding step, a centre for the bead is known, indicated by a cross in this figure. The intensities of the four closest pixels in a 3 × 3 square around the centre (red square) are then used in a weighted average to calculate the foreground. The five dimmest pixels within a 17 × 17 square around the bead centre (orange square) are averaged to give the background intensities. In this figure, the five dimmest pixels are indicated by the yellow squares. Figure courtesy of Dr. Matthew Ritchie.

against all N negative controls on an array. The detection p-value is then $1 - R/N$. Thus, a bead type that ranks higher than all the negative controls will be given a detection p-value of 0. The purpose of these detection values is to assist in excluding bead types unlikely to be expressed above background level, with a lower p-value indicating a lower probability that the intensity of the given bead type is due to non-specific hybridisation.

2.3.4 Detecting DE genes

BeadStudio provides a statistical test to determine DE genes between sample and treatments groups, with the user defining which arrays in the experiment belong to the relevant groups. For a particular bead type, the mean across the condition (cond) and reference (ref) arrays are calculated as I_{cond} and I_{ref} respectively, with variances S^2_{cond} and S^2_{ref}. These quantities are derived from the unlogged bead-type averages, although they may have been normalised by methods such as BGN.

The variances of negative controls across the two groups are denoted as $S^2_{\text{neg(cond)}}$ and $S^2_{\text{neg(ref)}}$ and used in the following regularised t-statistic:

$$t = \frac{|I_{cond} - I_{ref}|}{\sqrt{\frac{S^2_{\text{ref}}+S^2_{\text{neg(ref)}}}{N_{\text{ref}}} + \frac{S^2_{\text{cond}}+S^2_{\text{neg(cond)}}}{N_{\text{cond}}}}} \tag{2.6}$$

A p-value (p) is then calculated by assuming a two-sided standard normal for the test statistics of all bead types. "DiffScores" are then derived according to

$$\text{DiffScore} = 10\text{sgn}(I_{\text{cond}} - I_{\text{ref}}) \log_{10}(p) \tag{2.7}$$

where the sgn operator returns 1 if $I_{\text{cond}} > I_{\text{ref}}$, or -1 otherwise. Thus, DiffScores of 13 and 20 correspond to p-values of 0.05 and 0.01 respectively. The sign of the DiffScore also indicates whether a gene is up- or down-regulated.

2.3.5 Other analysis options supported by BeadStudio

Bead-summary data can either be analysed through BeadStudio or exported into tab-delimited text format, with one row per bead type ("Sample Probe Profile"), or one row per gene ("Sample Gene Profile"). In practice, each gene is usually represented by one bead type, so the contents of these files do not differ much. The bead-summary data for the control probes can also be exported on a per-bead-type basis ("Control Probe Profile") or by summarising all control types (e.g. the negative controls) into one set of observations ("Control Gene Profile"). The columns exported from BeadStudio can be defined by the user, but generally include average expression (AVG_Signal), standard error (BEAD_STDERR), number of beads on an array (Avg_NBEADS) and detection score (DETECTION_Pval). DiffScores, if they are available, can also be exported.

Analysis options within BeadStudio include common visualisation tools such as heatmaps and cluster dendrograms. More sophisticated options also allow expression levels to be viewed according to their position along the genome, potentially allowing genomic regions with aberrant expression to be identified.

2.4 Why Illumina arrays are attractive to researchers

Illumina expression arrays afford the opportunity to measure a very large number of transcripts at once, giving researchers a detailed picture of gene expression in their sample of interest. The ability to make 30 observations for a particular gene of interest is advantageous and it might reduce some of the measurement error inherent in microarrays. Since the replicates of a particular gene are spread across the array surface, the effects of spatial artefacts should be minimised. Moreover, the probe sequences are attached to beads rather than the array surface. Therefore, hybridisation should only take place only at a bead and all locations between beads should show no fluorescence. Hence we would expect the background levels to be low and consistent on the array surface.

A concern for traditional microarrays is the low number of available replicates, making reliable estimates of variance across samples problematic. Not only do BeadArrays offer more replicates of a particular gene on one array, but the ability to process more arrays in parallel should make it feasible and cost-effective to add more samples to an experiment and gain better estimates of gene-specific mean and variances. Furthermore, by being able to hybridise and scan multiple arrays simultaneously, we would hope to reduce the batch effects associated with running microarrays on different days or months, which complicate analysis.

2.5 Early uses of Illumina arrays in the literature

Confidence in the Illumina microarray technology can be measured in the number of publications using the technology, and particularly in high-profile studies. A notable project, the International Hapmap project (HapMap Consortium, 2003) describes patterns of common genetic variation in four different populations. A crucial part of the project was to identify SNPs in the genome and determine which genotypes were present at these SNPs. A large portion of this genotyping was done using two-colour Illumina arrays in the SAM format. Data from the project are available online and are a valuable resource for researchers investigating genetic variation in diseased or normal populations. Along with these data, the cell lines used can also be obtained to perform other studies.

One publication to make use of this resource is Stranger *et al.* (2005). In this paper, expression data for lymphoblastoid cell lines from 60 unrelated individuals of central European origin (commonly denoted as CEU) from the Hapmap project were generated using an early version of Illumina expression arrays. The arrays used were in the SAM format, with 1,433 bead types representing 630 genes. These genes were chosen as RefSeq genes lying within well-characterised regions of the genome. The intention of the project was to find associations between the expression levels of the genes and SNPs already genotyped in the same individuals. Each sample was hybridised to six separate arrays and distributed randomly among the SAMs used in the experiment. These experiments were processed using BeadStudio and exported as a sample probe profile with 1,433 rows and one column for each hybridisation. Normalisation was done with quantile normalisation before averaging the values for each individual. The 688 probes with the most variation were then taken forward for further analysis. A simple linear regression model was fitted to each probe intensity and to each SNP to see which SNPs are most correlated with gene expression level. Such SNPs may be regulatory element variants. Naturally, such a large number of tests (one per gene, per SNP) can lead to a lot of false positives if not treated carefully. Therefore, the authors used a number of methods to control the number of false positives or false discoveries, resulting in a modest number of genes with significant association.

The study of Barnes *et al.* (2005) aimed to answer the key question of whether Illumina expression data can be compared to data generated using Affymetrix platforms. Their experiment was a dilution design, where two distinct RNA samples are mixed in known amounts. The proportions of the two samples are expected to give a large number of DE genes, although the identities of these genes are not known in advance of the experiment. Mixtures of blood and placenta at percentages 100 to 0, 95 to 5, 75 to 25, 50 to 50, 25 to 75 and 0 to 100 were used on a pre-release version of the Human8 chip. BeadStudio was used to process the arrays, although the authors comment that BGN had a negative impact on data quality, and therefore elected to use quantile normalisation from within an existing Bioconductor package.

In a dilution design, the expression level of a given gene should correlate with the change in sample composition in a positive or negative direction. In Barnes *et al.* (2005) the proportion of genes that correlated with concentration was quite low on both platforms (35% and 33% for Illumina and Affymetrix respectively). However, correlation was more pronounced for probes with higher expression level. In other words, probes with low expression level tended to be consistently low, and measuring background noise rather than biological differences. Furthermore, the sequences provided by Illumina were compared to the genome using the BLAT tool (Kent, 2002). This revealed that out of the 24,114 probes supplied by Illumina, 19,924 matched known genes, with 2,978 probes unassigned. The correlation with the dilution series was found to be greater for probes targeting known genes, and was consistent for both Illumina and Affymetrix. Similarly, correlation of measurements of the same gene on both platforms was improved for genes with higher expression level and reliable annotation.

Citation	# Arrays	Filtering	Normalisation
Elvidge *et al.* (2006)	18	None	Quantile
MAQC Consortium (2006)	59	Cross-Platform	RIN
Golubkov *et al.* (2006)	6	Detection	Quantile
Greber *et al.* (2007)	18	None	RIN
Ramilo *et al.* (2007)	24	None	AN
Bykhovskaya *et al.* (2007)	12	Detection	AN
Stranger *et al.* (2007)	480	Variance	Quantile
Krig *et al.* (2007)	12	None	AN
Wang *et al.* (2007)	4	Detection	AN
Platts *et al.* (2007)	12	None	RIN
Deregibus *et al.* (2007)	6	None	Loess
Lenk *et al.* (2007)	15	Cross-Platform	CSN
Tesar *et al.* (2007)	6	Detection	RIN

Table 2.1: Summary of GEO datasets derived using Human6 chips. For each entry we list the citation, number of arrays used, type of filtering applied to the data, and normalisation method applied. All datasets had preliminary analysis done using BeadStudio.

The work of Stranger *et al.* (2005) and Barnes *et al.* (2005) details two of the earliest uses of Illumina technology. Although both used versions of the technology that were never commercially available, these results demonstrate the potential for Illumina to be used for high-throughput expression studies. Moreover, from an analysis point of view, they show the willingness of researchers to use analysis strategies other than those recommended by Illumina. Barnes *et al.* (2005) also showed that probe annotation should be accounted for in the analysis. I will explore this effect for commercial Illumina arrays in Chapter 6. The significance of Stranger *et al.* (2005) will become apparent in Chapters 3 and 4 when discussing the development of open-source software for Illumina and pre-processing issues for Illumina data.

2.5.1 Publicly available Human6 data

A standard requirement for publication is that the data supporting the analysis presented in a paper are made available. One such site that allows experimental data to be deposited is the Gene Expression Omnibus (GEO) (Barrett *et al.*, 2007). On 8th February 2008, I queried GEO for Illumina Human6 datasets and 22 were returned. Only 9 Human8 datasets were found, perhaps indicating a preference for the higher density arrays.

The datasets found in GEO were manually curated to exclude datasets that did not have a Pubmed reference listed. Some datasets were also found to relate to the same publication and therefore counted as the same dataset. A total of 13 datasets were then selected for further analysis. The characteristics of these datasets are summarised in Table 2.1, and a more detailed discussion of the results will be given in Chapter 6.

A number of commonalities were revealed when reviewing the use of Illumina technology in the literature. Firstly, BeadStudio was used to obtain bead-summary data and subsequent analyses were done with these quantities. Therefore, the common microarray tasks of image processing and background correction were already performed by internal Illumina methods and not accounted for in any of the GEO papers.

The analysis of many experiments included a filtering step to remove unexpressed probes from the analysis and reduce the amount of multiple testing. Commonly, this was done using the detection scores provided by Illumina (Golubkov *et al.*, 2006; Bykhovskaya *et al.*, 2007; Wang *et al.*, 2007; Tesar *et al.*, 2007), setting an arbitrary cut-off and requiring probes to exceed this cut-off on all arrays. Where the focus of the paper was to compare the results of different platforms (MAQC Consortium, 2006; Lenk *et al.*, 2007), a common list of transcripts was used in the analysis by comparing the list of transcripts available for Illumina to another platform.

A wide range of normalisation methods were applied to the data, possibly reflecting a lack of guidelines for the analysis of Illumina data. The majority of publications analysed data through BeadSudio using either AN, CSN or RIN. These methods also incorporate BGN, which some authors point out has the effect of producing negative intensities. These negative intensities could cause problems for further analysis, as noted previously (Barnes *et al.*, 2005). Some authors sought to avoid negative intensities by adding a small offset to the intensities on each array (Greber *et al.*, 2007; MAQC

Consortium, 2006).

Some papers used the detection scores as the outcome of the experiment (Wang *et al.*, 2007; Tesar *et al.*, 2007), whereas others went on to produce a list of DE genes between biological samples of interest (Golubkov *et al.*, 2006; Greber *et al.*, 2007; Krig *et al.*, 2007; Lenk *et al.*, 2007; Deregibus *et al.*, 2007; Elvidge *et al.*, 2006; Platts *et al.*, 2007; Greber *et al.*, 2007). In keeping with recent trends in microarray analysis (Allison *et al.*, 2006), many authors also sought relevant GO terms or pathways amongst the significant findings (Greber *et al.*, 2007; Krig *et al.*, 2007; Lenk *et al.*, 2007; Platts *et al.*, 2007).

One notable study using the Human6 platform was provided by MAQC Consortium (2006). This was a global collaboration aimed at assessing reproducibility of microarray results both between platforms and within the same platform. Ten microarray platforms were used, including six high-density platforms with over 30,000 unique probe sequences. Similar to Barnes *et al.* (2005), a dilution design was used with Universal Human Reference RNA (UHRR) and Universal Human Brain Reference RNA (UHBRR) at varying mixtures (100 to 0 and 75 to 25). These four samples were replicated 5 times and this set of 20 arrays was hybridised at three different locations. The project provides a public resource, all data derived being freely available and the samples used available for purchase.

To assist in comparing the performance of the different microarray platforms, a list of genes common to all platforms was drawn up by first matching all probes on every platform to the RefSeq database, and then seeing the overlap between all platforms. This list was condensed to 12,091 common genes by selecting only one probe for each gene from each platform. The reproducibility of each platform was assessed using the ratio of standard deviation to the mean of replicate observations, known as coefficient of variation (CV). The CV was calculated for all genes across all replicates within the same site, and then across all sites. Both Affymetrix and Illumina achieved the lowest median CV ($< 10\%$ within a site and $< 12\%$ across all sites, respectively) and therefore the most reproducible measurements. These CV calculations were made including only genes that were generally detected for each platform (genes that were detected in at least three of the five replicates). The definition of detection was unique for each platform, with genes requiring a detection p-value of less than 0.05 for Illumina. For Illumina, 85% of these calls were found to be in agreement, which was the highest percentage of the platforms compared in the study.

Finally, the platforms were judged on their ability to detect differential expression. For this, a simple t-test was performed, on a per-site basis, between all replicates of 100% UHRR and 100% UHBRR. Differentially expressed genes were then selected with a p-value cutoff of 0.001 and a two-fold change in intensity between the two samples. The composition of the resultant gene lists was then compared between platforms by calculating the percentage of genes in list X that were also present in list Y. This percentage was over 60% for all comparisons between the high-density arrays. For Illumina arrays processed at different sites, the percentage overlap was greater than 87%, and for Affymetrix agreement was on average 80%.

In summary, aside from providing a valuable research tool, the MAQC project showed that although being a relatively new technology, Illumina microarrays could produce high-quality data. Promotional literature produced by Illumina indicated that the cost of running Illumina arrays was substantially less expensive and required less biological material than its competitors. Therefore, it is not surprising that interest in using Illumina arrays is increasing.

2.6 Conclusions

The Illumina gene expression platform has many unique features that makes it appealing to genomics researchers planning high-throughput experiments. The relatively high number of observations for each gene and the random arrangement of beads should produce high-quality, reliable results. The volume and profile of publications using Illumina technology show that researchers are willing to trust the technology. A survey of publicly deposited data also showed the diversity of projects carried out on the Illumina Human6 platform. However, the analyses of all these projects were performed using Illumina's BeadStudio software.

BeadStudio has a number of restrictions that limit its use for bioinformatics research. The main restriction is the lack of choice over image processing. This has been found to have a dramatic effect on the analysis of other microarray platforms. Therefore, for the bioinformatician, it is difficult to know if these methods are appropriate for the data without in-depth exploratory analysis. Whilst Illumina data are believed to minimise the spatial effects seen on other arrays and produce robust measurements, there is no way to assess this with the summarised data output from BeadStudio. Given the random nature of the arrays, one would need both the individual bead locations and intensities to inspect this, and these are lost once the data are summarised.

There are also practical issues to consider regarding the software. For instance, BeadStudio only runs on Windows-based computers and is not available without a licence. To assist reproducible research it would be beneficial if analyses could be repeated on any computer without the need for specialist software. There are already many microarray software packages available and it would be useful if the analysis of Illumina data could interface with these. Illumina's primary focus is not to supply state-of-the-art bioinformatics methods, and therefore the tools implemented in their software may not reflect current trends in the field. An open-source framework should facilitate the implementation of new methods and tailor-made solutions, rather than relying on the company to update their software. I will discuss the development of open-source software for the analysis of Illumina arrays in Chapter 3.

Although two dilution experiments have been published comparing Illumina favourably with other platforms (Barnes *et al.*, 2005; MAQC Consortium, 2006), these publications do not look into the processing of Illumina data in detail. A caveat in such dilution experiments is that it is not known what genes are DE in advance of the experiment. Therefore, we cannot truly quantify the ability of the technology, and the processing methods used, to identify DE genes correctly without incurring too many false positives. The use of a more informative control experiment will be discussed in Chapter 5.

Chapter 3

beadarray: open-source software for Illumina bead-based microarrays

3.1 Introduction

Previously, I have shown that although Illumina arrays were well-received, little was known about their pre-processing steps. Furthermore, the software provided by Illumina did not accommodate some of the QA tools commonly applied to other microarray technologies. In this chapter, I describe the development of open-source software (beadarray) for the analysis of Illumina microarray data. This project was instigated in early 2005 to investigate the processing of Illumina data and to assist in the analysis of a pilot study conducted by the Wellcome Trust Sanger Institute, which later formed part of an association study (Stranger *et al.*, 2005). As part of this pilot study, we were able to gain advanced access to bead-level data.

After describing the Bioconductor project, the framework on which the software is based, I give the details of the beadarray software. A key feature of the package is the ability to read both bead-level and bead summary data. The data structures employed to store these two types of data are described, along with the QA and analysis options implemented in the package.

This chapter is based on the peer-reviewed work presented in Dunning *et al.* (2006a) and Dunning *et al.* (2007), along with documentation for the beadarray package.

3.2 The Bioconductor project

The Bioconductor project (Gentleman *et al.*, 2004) is an online repository of freely available genomics software. Bioconductor covers a wide range of tools for the analysis and visualisation of microarrays and related high-throughput technology. All software is primarily written using the `R` statistical programming language (Ihaka and Gentleman, 1996). This is a natural choice as `R` is well-known for its wide range of statistical and visualisation tools. The open-source nature of Bioconductor gives users full access to these statistical and visualisation tools and allows them to understand what is being done at each stage of the analysis, rather than relying on proprietary software that may not be fully explained. Therefore they are able to judge if the methods are appropriate to their data and to modify them to their own needs if required.

Bioconductor is made up of software packages, each package providing functionality to analyse a particular microarray technology, or implementing a new algorithm. Many publicly available datasets are also released through Bioconductor, along with the annotation for many popular microarray platforms. It is recommended that Bioconductor software is written to make use of common data structures. Not only does this promote re-usable code and interaction between packages, but it also reduces the learning curve for users. A key feature of the Bioconductor project is the interaction between developers and users. The mailing list encourages users to ask questions about the use of a particular package and also to report bugs and suggest improvements.

A software package included in Bioconductor consists of the following elements

- R code - Code to achieve the functionality of the package and also defining the data structures employed. These are primarily written in the `R` language, although functions that are more memory-intensive and time-consuming may be written in other languages such as `C` or `Fortran`, and then called from within the `R` code of the package.

- Documentation - Each separate function must have documentation giving adequate instructions to users, including descriptions of the input to the function and the output to be expected. Some example code segments should also be provided to demonstrate the use of the function.
- Example data - A small illustrative dataset exemplifying the use of the package.
- Package vignette - More detailed documentation of the package and methods used. This should give a step-by-step guide about how a typical analysis using the package may be performed. The Sweave tool (LEISCH, 2002) provides a convenient framework for creating such documents with embedded R code that may be reproduced by users.

Microarray technologies are being continually updated to have higher density and revised annotation. Therefore packages within Bioconductor must be constantly upgraded to meet the demands of data derived from these new technologies. Bioconductor operates a six month release cycle to ensure that all packages are up-to-date. Two versions of the Bioconductor project exist: the release version that has been rigorously checked for errors, and the developmental version where more cutting-edge code is available. Prior to release, all packages are checked to ensure that documentation can be found for all functions and example code runs without error. Failure to comply to either of these requirements results in the package being withdrawn from the release version.

3.3 Processing bead-level data using beadarray

I now describe the main features of the beadarray package and the object types and classes used to represent Illumina data in an efficient manner. One main goal of the software is to utilise tools from existing microarray technologies in the analysis of Illumina data. Therefore it was always our intention to implement the software in R and submit to the Bioconductor project. Figure 3.1 gives an overview of the package and the different entry-points for the user. Whilst we wanted to focus on the ability to use bead-level data, we also wanted to cater for users with only bead-summary data. However, as explained later in the chapter, downstream analysis can be performed in the same way regardless of whether the analysis started with bead-level or bead summary data.

3.3.1 Reading bead-level data into beadarray

Bead-level data can be read into memory using the `readIllumina` function. This function was designed to run in the same way as `ReadAffy` function within the affy package for Affymetrix data (Gautier *et al.*, 2004a), where the user does not need to set complicated parameters in order for the function to run. Such a function is therefore appealing to inexperienced `R` users.

By default, `readIllumina` will find all images and text files within the current `R` working directory and apply the image processing steps used by Illumina. Users are able to choose whether to use the sharpening procedure or choose a different window size to calculate the local background (rather than the default 17 × 17). Alternatively, the background corrected intensities can be taken directly from the text file to save time and memory.

Other parameters to `readIllumina` include the option to import a targets file that specifies the samples hybridised to each array. Such a file is commonly used when analysing two-colour arrays via limma and can contain other information such as the date of hybridisation, which could be useful for diagnostic purposes. The metrics file created by BeadScan may also be imported to give an indication of the scanning diagnostics assigned by Illumina.

The following code executed within an `R` session will read bead-level data from the current working directory

```
> BLData <- readIllumina(textType = ".txt", )
```

As a rough guideline, this function takes about a minute to read the data from a BeadChip on a PC with 2GB of RAM and a 3Ghz processor. Initial code to read bead-level data was developed for the low-density arrays from the Sanger pilot study and written entirely in `R`. However, this proved too slow when attempting to read BeadChip data. Therefore, the majority of

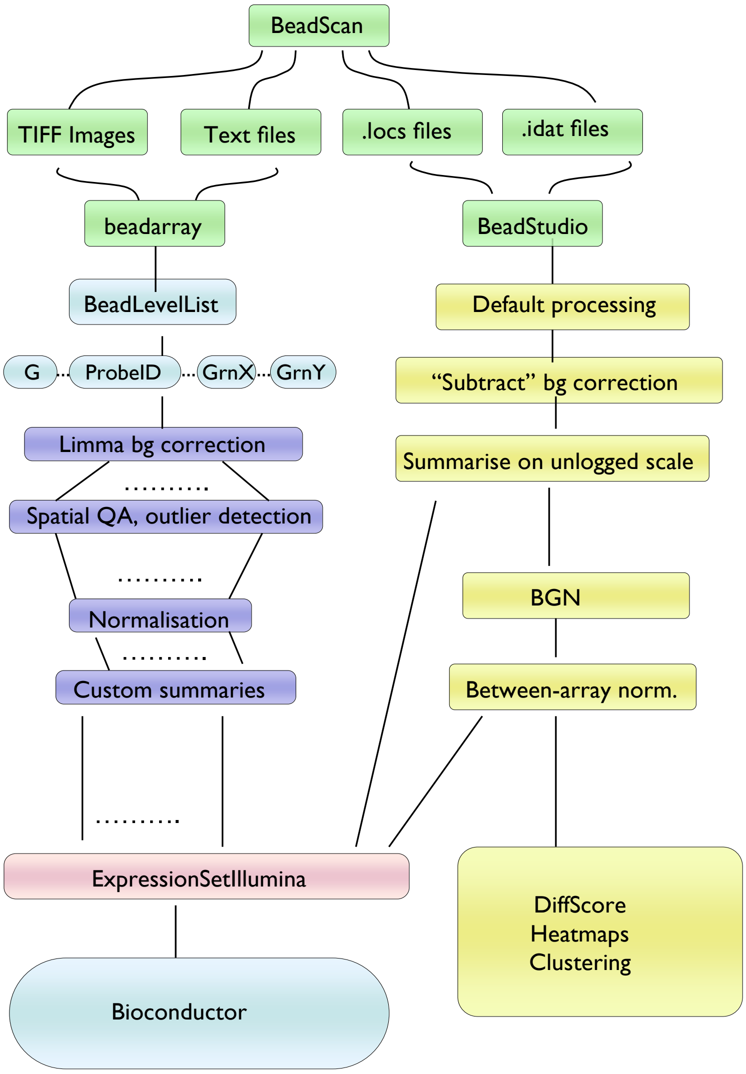


Figure 3.1: An overview of the beadarray software and the various tasks it can perform compared to BeadStudio. The software can be used to analyse either bead-level data or data exported from BeadStudio, although availability of bead-level data allows a more flexible analysis.

the `readIllumina` function was written in `C`.

3.3.2 Representation of bead-level data in beadarray

Once imported, the bead-level data are stored in an object of type *BeadLevelList* (called `BLData` in the example code above). In the current version of beadarray, this is stored as an `R` *environment* variable. Earlier versions of beadarray used a *list* variable to store bead-level data, similar to the *RG-List* object in limma. This data structure stores data for a microarray experiment with F features on K arrays by having a series of $F \times K$ matrices in a list. Typically, there will be a matrix of green and red intensities in the case of two-channel data. Many processing and QA options are then defined for this class.

Whilst we wanted to make full use of existing software tools, it quickly became apparent that the *RG-List* would not be suitable for Illumina data. The main problem is that we cannot always assume the same number of beads on each array. Whilst this was true for our preliminary bead-level data, once bead-level became widely available it was seen that the text files had differing numbers of beads. This is due to the scanning software removing beads that are not decoded by Illumina, or sometimes beads that are found to be outliers for their bead type. Therefore the number of features (F) varies and we cannot construct the required matrices directly from the data. One solution to the problem would be to pad-out the values in each column to ensure we have an equal number of observations. However, this is not very satisfactory as the differences in features between arrays may be on the order of tens of thousands.

The major difference between a *list* and *environment* is the way that subsetting is done. In a list structure, `BLData` would be divided into matrices of equal dimensions, each matrix representing a different set of information (e.g. foreground intensities) and having the same number of columns as arrays in the experiment. However, in an *environment*, we subset first by array, and then by the information stored for each array. This allows for a different number of features on each array.

The *list* structure in R can also be memory inefficient when operated on, usually creating many copies of itself. Such behaviour is obviously undesirable when dealing with large datasets. An *environment* does not create multiple copies of itself, but instead keeps modifying the same object. As a consequence, users are not allowed to modify the environment easily.

The *BeadLevelList* class has been developed to contain useful information for describing Illumina data at the bead level. This information is broken into different sections, or slots, including data for each bead (*beadData*), experimental information about the arrays (*arrayInfo*) and phenotypic information (*phenoData*). For convenience, the function `getArrayData` can be used to retrieve the data for a particular array, including foreground and background intensities, coordinates and ProbeIDs. Therefore, users of the package do not need to understand the internal representation of the *BeadLevelList* in order to extract the information they need.

The bead-level data for any Illumina assay are stored in the same text and image format. Therefore, the same call to the `readIllumina` function and *BeadLevelList* objects can be used. A slight exception is that two-colour Illumina arrays incorporate extra information for the red channel.

It should also be noted that for BeadChips where more than one strip represents the same sample (e.g. the Human6 chip), a separate image and text file is produced for every strip, and therefore separate entries for each strip are created in the *BeadLevelList* object for QA purposes.

3.3.3 Visualisation of bead-level data

I now describe the main diagnostic functions available within beadarray for plotting per-bead quantities of interest. These include boxplots (`boxplotBeads`), imageplots (`imageplot`) and density plots (`plotBeadDensities`). All of these functions make use of the `getArrayData` function to retrieve the quantities of interest, which are specified by using the `whatToPlot` argument. Options are `G`, `Gb` and `residG` (green residuals) for single channel data with the addition of `R`, `Rb`, `residR`, `M` (log-ratios) `residM` or `A` (average log-intensities) for two-colour data. The `qcBeadLevel` function can be used to generate these plots automatically for all arrays in the *BeadLevelList* object.

As described previously, the `imageplot` function can be used to identify spatial artefacts. Similar functionality can be found in the limma package, although it was not immediately applicable to Illumina data as it assumes equal spacing of features on the microarray. Because of the large number of beads on each array, `imageplot` maps a grid of size specified by the `nrow` and `ncol` arguments onto the array surface and averages the intensities of the beads within each section of the grid.

Being able to view the arrays in this manner is a clear advantage over viewing each image individually in high resolution. This kind of visualisation is not possible when using the summarised BeadStudio output, as the summary values are averaged over spatial positions. Imageplots in `R` are also more convenient than scrutinising the original tiffs, as multiple arrays can be visualised on the one page. Additionally, imageplots can be generated for the background intensities or residuals, which would not be possible by viewing the TIFF images directly. An added avantage is being able to associate the imageplots with the positions of beads that are outliers for their bead types. Firstly, outliers are identified using the `findAllOutliers` function, the result being a vector of identifiers for each outlier bead. The `plotBeadLocations` function will then plot the x and y coordinates of the relevant beads.

Such spatial artefacts would be a concern for conventional microarrays where a fixed position on the array is allocated to a particular gene. Thus, genes on a particular part of an array would be inaccurately measured. The outlier removal method used by Illumina is supposed to account for such regions of unusual intensity. Therefore, it is after summarised data have been created that we can assess the impact of possible problems observed at the bead level.

3.3.4 Creating bead-summary data

The `createBeadSummaryData` function can be used to summarise the values for each bead type. Outlier removal is performed by `findAllOutliers`, which excludes outliers as defined by Illumina (see Page 30) using a 3 MAD cut-off from the median of each bead type as the default. However, users are able to specifiy whether to perform a $\log_2$ transformation prior to excluding outliers, change the MAD cut-off, or use a trimmed mean or median of all bead intensities. For some two-colour arrays, the user can also choose to summarise other quantities of interest, such as red intensities or log-ratios. In the case of some BeadChip arrays where two items in the *BeadLevelList* correspond to the same array, we may collect together all replicates of each bead type that appear on both strips. This is done by specifying the argument `imagesPerArray` to be 2. Otherwise the function will treat each item in the *BeadLevelList* as a different entity to be summarised. The call to create bead-summary data is simply:

```
> BSData = createBeadSummaryData(BLData)
```

By default, we summarise the values for the green channel. In the case of two-colour data, one may wish to create summary values for the red and green channels separately, or summarise the log-ratios for each bead. This can be achieved by setting the `what` argument to `RG` or `M` respectively.

The default settings for `createBeadSummaryData` assume that the same bead types are to be found on each array in the experiment, which will be true in general. Alternatively, one might wish to summarise only the genes present on the array and not the control probes. This is possible with use of the `probes` argument, which is used to specify the identity of bead types to be summarised.

3.3.5 Proceeding with bead-summary analysis

The object type used to store bead-summary data depends upon the type of Illumina technology assay analysed. The default class for expression data, *ExpressionSetIllumina*, is an adaptation of the *ExpressionSet* class, written by the Bioconductor core development team to store and manipulate data from high-throughput genomics experiments. The *ExpressionSet* structure was devised for single-channel data, with an $n \times k$ matrix used to hold the expression values of n probes on k arrays. Obviously, this structure can only be used for Illumina data after the summarisation step. Within an *ExpressionSet*, the expression matrix can be accessed at any time by the convenient `exprs` function. Similarly, standard errors are found in an $n \times k$ matrix and accessed by the `se.exprs` function.

By basing *ExpressionSetIllumina* on *ExpressionSet*, we automatically inherit the ability to extract expression and standard errors using the same `exprs` and `se.exprs` functions. Therefore Bioconductor users who are new to Illumina data will be able to access data in a familiar way. We also chose to store the number of replicates for each bead type on each array in a `NoBeads` matrix with a `NoBeads` accessor function and detection scores in the `Detection` matrix with a `Detection` accessor function.

3.4 Analysing Gene expression bead-summary data

I will now discuss the analysis of bead-summary data using beadarray. The data in question could be the result of performing a QA on bead-level data as described above, or alternatively, beadarray is able to read the result of processing expression data using BeadStudio. At present, beadarray does not directly support the reading of summarised data from other Illumina assays. This is mainly because the majority of my research has involved the analysis of expression data, or exploratory analysis of bead-level data.

3.4.1 Reading BeadStudio output into beadarray

The format of the “Sample Probe Profile” (SPP) file exported from BeadStudio is already similar to that required by an *ExpressionSetIllumina* object as it has one row for each bead type. However, the columns in the SPP file are arranged with the expression values, standard errors and number of beads in adjacent columns for the same array. Therefore, the main challenge of reading BeadStudio output into beadarray is how to recognise the correct columns for each array and assigning to the correct part of an *ExpressionSetIllumina* object. A complication is that no standard format of the BeadStudio output exists and users are able to select as many columns as they like. Most column headings used by BeadStudio are generally the same between versions of the software (e.g. AVG_Signal for the expression values), but the column names for the standard errors have been known to change. We therefore assume the column headings from the latest version of BeadStudio (version 3 at the time of writing), but give users the chance to define alternative headings. Probably the most important column heading to specify denotes the column containing an identifier for each bead type. By default, this is assumed to be the column which contains unique numeric codes for each bead type.

In BeadStudio, it is also possible to export annotation information. However, we recommend that this information is not exported if the file is to be read into beadarray, as some of the special characters used in the annotation fields cause problems in R. Also, the inclusion of the annotation is unnecessary as it can be retrieved later on from other Bioconductor packages, such as illuminaHumanv1.

The function `readBeadSummaryData` is used to read exported BeadStudio data into beadarray. The minimum requirements for the function are the specification of a file name in the `dataFile` parameter, relating to the SPP file to be read. The complicated nature of BeadStudio output means that the list of parameters to this function can potentially be quite long and therefore full details will not be presented here (see the beadarray documentation for more information). Key points to note are the `columns` parameter, which allows the user to specify the names of the columns in the SPP file containing expression values, standard errors, the number of beads and detection scores. The `ProbeID` parameter also allows the column containing the unique identifiers for each bead type to be specified. This is a crucial step, as the *ExpressionSetIllumina* class does not allow repeated row names. Other parameters such as `skip`, `sep` and `quote` are important in specifying the format of the file. The default values of these are set to read BeadStudio version 3 output. Many common errors encountered during the execution of `readBeadSummaryData` can be solved by correctly setting these parameters, and wherever possible, `beadarray` will try to provide informative error messages. If problems using this function are reported to the Bioconductor mailing list, then the responses may be used to assist users with similar error messages.

Once the SPP file has been successfully read into memory by the `readBeadSummaryData` function and the contents have been verified, a valid *ExpressionSetIllumina* object is created. Essentially, this process involves matching the column names supplied by the user to columns in the SPP file and then creating a separate matrix for the expression values, standard errors, number of beads and detection values. The column and row names of these matrices are then set to the names of the arrays being read (determined from the SPP file) and the ProbeID values respectively. These matrices are then stored in the *assayData* slot of a newly created *ExpressionSetIllumina* object. Slots such as *assayData* are accessed using the "at" operator, with the $ operator then required to access the individual matrices. However, accessor functions such as `exprs` make this process convenient as the user does not need to know the details of how the class is implemented. Hence, the following two lines of code produce the same result.

```
e = BSData@assayData$exprs
e = exprs(BSData)
```

If quality control information has also been exported from BeadStudio, the name of this file can be supplied as the `qcFile` parameter. As with the SPP file, the columns exported from BeadStudio can be specified by the user, and therefore parameters can be set to specify the contents of this file. If imported, the quality control information is stored in a separate slot (QC) to the data imported from the SPP file and accessed using the `QCInfo` function.

Additional information about the samples can be imported through the `sampleSheet` parameter. This is a text file, usually created by Excel, that allows users to specify what samples were hybridised to each array and any grouping of the samples. This information is stored in the *phenoData* slot of *ExpressionSetIllumina*, which is a standard feature of an *ExpressionSet* and can be accessed using the `pData` function.

3.4.2 Visualisation of bead-summary data

beadarray provides a way of displaying MA (see page 10) and scatter (XY) plots for a set of arrays. We call this a *MAXY plot* with the MA plots for the arrays in the upper right and XY plots in the lower left.

Boxplots cannot be generated directly from an *ExpressionSetIllumina* object. However, they can be produced by only a few lines of additional code. First, one would need to extract the data to be plotted using one of the accessor functions (e.g. `exprs`). This returns a matrix which must be converted into a data frame, using `data.frame`, before plotting with the `boxplot` function.

Plots of the quality control information, if available, can also be useful for diagnostic purposes. The automatic generation of such plots is not currently supported, but can be easily generated by the user. The `QCInfo` function returns a matrix with rows for each control probe and columns for each array. Therefore it is straightforward to plot the response of a particular control across all arrays.

Other plotting tools available in `Bioconductor`, such as cluster diagrams and heatmaps, generally require an expression matrix and therefore can easily be applied to Illumina data in the *ExpressionSetIllumina* format.

3.4.3 Further analysis of bead-summary data

Representation of bead-summary data using the *ExpressionSet* allows for existing methods within Bioconductor to be applied to Illumina data. For instance, the possible normalisation methods are extended beyond the methods given in BeadStudio. The `normaliseIllumina` function within beadarray can be used for normalisation by taking an *ExpressionSetIllumina* object and returning a copy of the object with modified expression values. Options supported by `normaliseIllumina` include quantile, qspline, and rank invariant, which are called directly from other packages such as affy. Alternatively, the expression matrix can be normalised by any other methods existing in Bioconductor.

The linear modelling approach described in Chapter 2 can be applied to Illumina data via limma. In particular, one needs the expression matrix returned by `exprs` and a design matrix defining the assignment of samples in the experiment. The construction of an appropriate design matrix can be assisted by information stored in the *phenoData* slot. The function `lmFit` can then be used to estimate coefficients as in (2.2). Once a model has been fitted, the analysis would proceed exactly as the examples given for single-channel data in the limma user guide.

After performing a differential expression test, it is often useful to relate the results back to prior knowledge of the probes on the array to ascertain if the results are biologically meaningful. As for most microarrays, annotation for popular Illumina chips can be obtained through Bioconductor (e.g. illuminaHumanV1 for Human6 version 1 arrays). These annotation packages provide a series of environments, each environment providing a mapping from the identifiers on the array to a particular genomic property. The `mget` function can be used along with a set of keys to be looked up (unique identifiers for each probe) and the environment name. For instance, the environment `illuminaHumanV1CHR` maps probes on the Human6 chip to a chromosome number and is used as follows.

```
ids = rownames(exprs(BSData))
chrs= mget(ids, illuminaHumanV1CHR)
```

The resulting vector gives the chromosome that each probe on the Human6 chip resides on. The consistent naming conventions for environment packages mean that repeating the same command for the MouseV1 chip requires the use of the `illuminaMouseV1CHR` environment with a list of appropriate identifiers. Access to this annotation information enables interaction with packages such as GOstats (Falcon and Gentleman, 2007) in order to find enriched GO terms or pathways among the results of a differential expression analysis.

3.5 Conclusions

BeadArray technology will become increasingly popular and I anticipate that beadarray will become an important tool in the analysis of Illumina data. The main benefit of beadarray is its flexibility. The package offers a variety of image processing and background correction methods, rather than the default methods used by Illumina. Having access to the bead-level data provides scope for users to develop their own analysis methods, or to interact with methods not supported by BeadStudio. Example usage of the package is given in Chapter 4.

The uniformity of bead-level data means that beadarray is able to read the output from any Illumina experiment. I have been able to process the results of Illumina expression, SNP, CNV and methylation experiments on both SAMs and BeadChips using the same commands for bead-level processing and QA. The format of bead-level data seems to be more stable than bead-summary data. With each new version of BeadStudio, the column names used within the software are changed slightly and it is problematic to support all possible options within beadarray without requiring the user to know some information about which software version was used.

I also find bead-level data more convenient as I perform most of my analysis on Linux, which is unable to run the BeadStudio software. With beadarray, a simple script can be used to read raw data, produce diagnostic plots and create summarised data. Therefore, the package is amenable for use in core facilities producing large numbers of arrays where processing data using BeadStudio may not be feasible and reproducible research is required. The `R` language also offers the opportunity for parallel processing and using more than 3Gb of RAM (which is the limit of current Windows machines), which should be beneficial for large datasets. I have not yet investigated the prospect of running beadarray in parallel, although it is clear that functions such as `readIllumina` and `createBeadSummaryData` could be termed "embarrassingly parallel". This means that they involve many identical operations that are performed independently, when in fact they could be performed simultaneously.

Other analysis options for Illumina data have arisen since the creation of the beadarray package. The BeadExplorer package is available in Bioconductor and gives a graphical user interface for users not familiar with R to read the output of BeadStudio and analyse the data through other existing packages such as limma. This package does not appear to be actively maintained. The IlluminaGUI project (Eggle and Schultze, 2007) offers a similar interface to other Bioconductor packages, although it is web-based rather than hosted in Bioconductor itself. Another Bioconductor package, beadarraySNP has been developed, although it is only for the output of Illumina SNP assays.

Finally, the lumi Bioconductor package (Du *et al.*, 2008) has also become a popular choice. Users are able to import bead-summary data with a simple command and the processing steps applied to the data are recorded as a history. The package also includes a number of unique features such as a novel annotation method for microarrays, nuID (Du *et al.*, 2007), and a transformation method, VST, based on the popular VSN method applied to microarrays (Lin *et al.*, 2008).

Despite these recent additions, beadarray has an important role to play in the analysis of Illumina data. The packages listed above all have advantages, but only deal with the summarised output of one technology (i.e., expression or SNP). On the other hand, beadarray is able to analyse the raw data from any experiment due to the uniformity of the bead-level data. The ability to perform detailed diagnostics and flexible analyses should be very appealing to bioinformatics researchers. The package is frequently discussed on the Bioconductor list and use of the package has been reported in institutes such as National Institute for Health, Harvard Medical School, Virginia Bioinformatics Institue, Walter and Eliza Hall Institute (Melbourne), University of Illinois, The Netherlands Cancer Institute, Leiden University, Turku Centre for Biotechnology (Finland), Australian Genome Research, University of Manchester, The European Bioinformatics Institute, as well as many users at the University of Cambridge.

Chapter 4

Investigation into the pre-processing of Illumina data

4.1 Introduction

In this chapter, I describe the low-level properties and processing of an experiment for which bead-level data were made available. The chapter is presented in the form of a worked example and goes through the steps one might perform when analysing Illumina data. The functionality of beadarray is also demonstrated.

As previously described, Stranger *et al.* (2005) studied the expression levels of 630 genes in 60 individuals from the Hapmap project. The published data comprised of five SAMs with each of the 60 individuals replicated 4 to 6 times. By special arrangement with Illumina, we were able to gain access to bead-level data before this type of data was available to the wider community. We also had access to a larger dataset than described in Stranger *et al.* (2005). This consisted of 15 SAMs profiling all 270 HapMap individuals, each individual replicated 4 to 6 times. This enabled us to evaluate the technology and develop beadarray with a view to handling large-scale experiments.

In this chapter I use a subset of 10 arrays from Stranger *et al.* (2005) to describe how the unique features of bead-level data allow a more detailed QA, as opposed to the summarised data used in most studies (see Chapter 2). I will also show how importing the data into R allows methods from other microarray technologies to be applied. This dataset is intended for teaching purposes and was presented as a tutorial at the BioC07 conference in Seattle. Hence, it shall be referred to as the BioC07 dataset. In the second part of the chapter, I show how the `beadarray` package can be used to process all 15 SAMs (referred to as the HapMap dataset) without the need for BeadStudio. This chapter is based on Dunning *et al.* (2006b) and user guides distributed with the `beadarray` package. It will also demonstrate the importance of the processing steps introduced in Chapter 2 before these are investigated in more detail in Chapter 5.

The `R` code to reproduce the figures in this chapter is provided as an Appendix.

4.2 Investigating the BioC07 dataset

The BioC07 dataset consists of 10 arrays with five replicates of two individuals, which we will refer to as samples A and B for convenience. For each array, there is a 6Mb `TIFF` image and a 1.6Mb `csv` file, the `csv` file containing 49,777 rows of data. Note that these `csv` files were generated using an early version of BeadScan and may not reflect the current output from the scanner. In particular, all beads are reported including those that failed the decoding process (these are assigned a ProbeID 0) and beads that are outliers for their bead type (these are sometimes removed by BeadScan). These bead-level data can be read using the `readIllumina` function, which creates a *BeadLevelList* object as described in Chapter 3. I now explore the bead-level data for the BioC07 dataset.

4.2.1 Image Processing

Firstly, I use the `boxplotBeads` function to plot various per-bead quantities for all 10 arrays. The raw foreground intensities are shown in Figure 4.1A. These are the raw values obtained by using the image processing steps described by Illumina and the boxes are colour-coded according to sample type (A or B). The design of the experiment meant that replicates of the same sample were randomly located on different SAMs. Therefore, some differences in underlying intensity between the different replicates could be expected. However, the raw foreground intensities show good consistency, with a median of around 10 and 25th and 75th quantiles of around 9.8 and 10.3 for

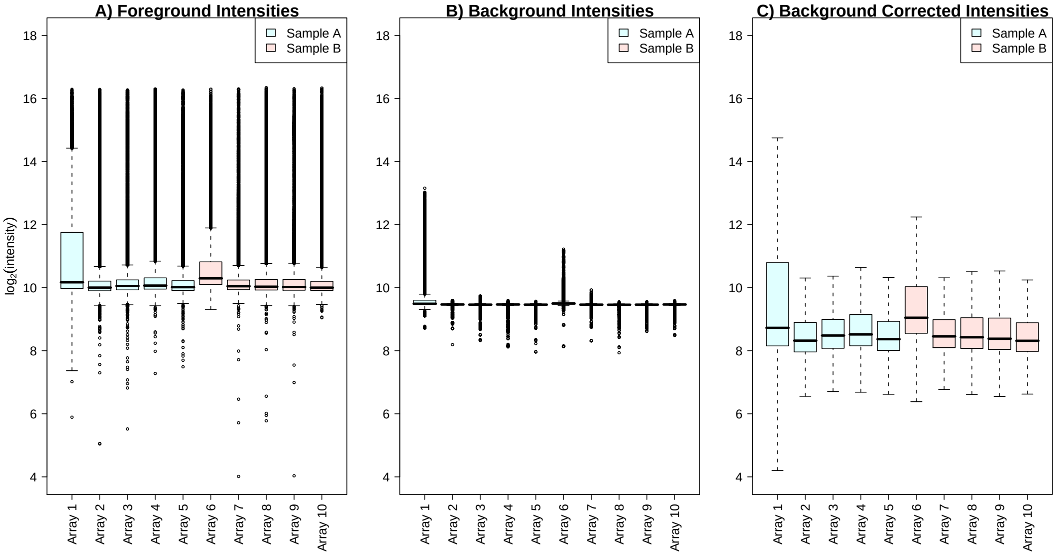


Figure 4.1: Foreground (A), background (B) and background corrected (C) intensities of all beads in the BioC07 dataset. Arrays are coloured separately for Sample A (blue) and Sample B (red). The foreground intensities are seen to be consistent across the dataset, with the exception of arrays 1 and 6 which are generally have higher intensity. The background intensity has extremely low variability on all arrays.

all arrays, except Array 1 and Array 6. Even though Array 1 has the same median value as the others, there are many more beads with high intensity. For all arrays, the intensities are skewed towards lower values. The spread of background intensities is extremely low (Figure 4.1B), and the majority of beads have background intensities around 9.5 on the $\log_2$ scale. With the exception of arrays 1 and 6, the other arrays do not have background values that exceed 10.

Background adjustment was applied to these 10 arrays separately using the `backgroundCorrect` function in `beadarray` with the default options. The default background correction method is the simple subtraction given by (1.1) and thus mimics the background correction performed by BeadScan. After this adjustment, the spread of the data is seen to increase further with Array 1 and Array 6 standing out even more as possible outliers (Figure 4.1C).

In other words, the differences in median levels between arrays are seen to be increased. Comparing these background corrected levels to the example two-colour experiment in Figure 1.3, we see a much tighter range in Figure 4.1. In fact, the array that most resembles the arrays we have seen before is Array 1, whilst the other arrays in the BioC07 dataset are found to have a very low range of intensity values.

Given that the background level lies just above the lowest values of the foreground, we might be concerned that background correction could cause negative values to appear, which would be removed from the analysis after $\log_2$ transformation. This is major concern for two-colour arrays and a motivation for more complicated background correction approaches than the simple subtraction applied to the data. However, a simple calculation reveals the largest number of negative intensities on any of these arrays is only 105 after background correction, even for Array 1.

4.2.2 Spatial Plots

The imageplots of the background adjusted intensities of the BioC07 data are shown in Figure 2, with yellow and red indicating low and high intensity regions respectively. These were generated using the `imageplot` function in beadarray and imageplots of the residuals for each bead had similar results. The colours on each plot are calibrated in such a way that the same shade always corresponds to the same intensity. A saturation intensity threshold is set for each array, with all intensities outside these limits given the same colour. For these plots, the saturation levels have been set to 6 and 16 respectively. Ideally, we would like to see a random distribution of colours and no tendency for any part of an array to have higher or lower intensity than any other. Our expectations are generally met for the dataset with the exception of arrays 1, 3 and 6. On Array 1, the left side of the array is consistently higher than the rest of the array. Moreover, as the imageplots for the other arrays are dominated by yellow or orange, these high intensity values are higher than most beads seen in the dataset. This agrees with the observation made from Figure 4.1 that this array has a much higher percentage of higher intensity beads. By eye, spatial artefacts are seen for Array 3 (bottom-left of the array) and Array 6 (bottom-right of the array).

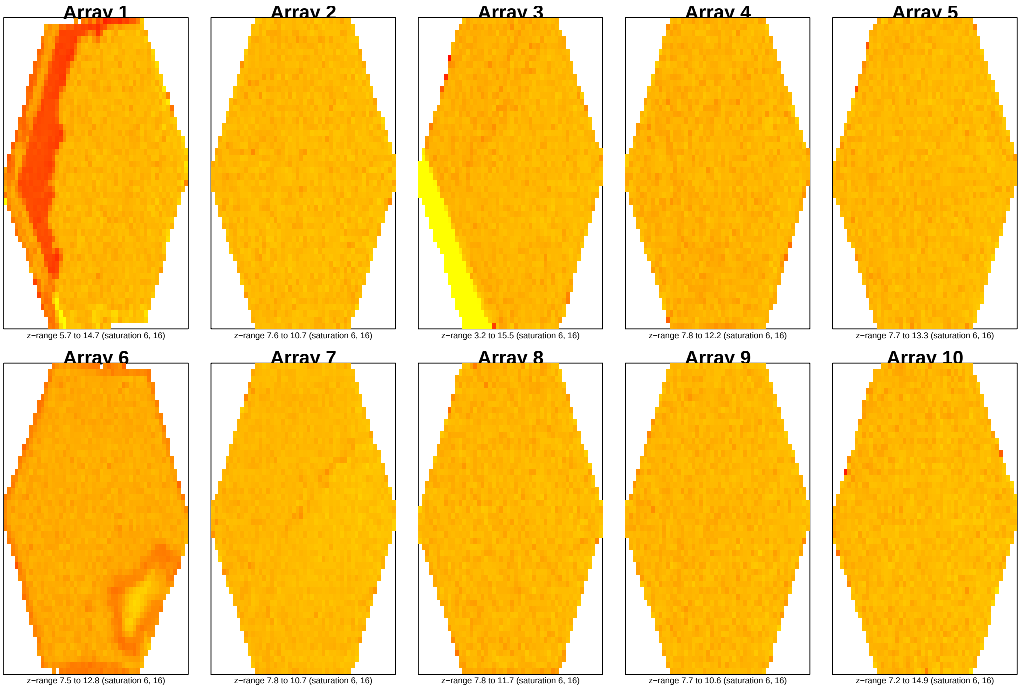


Figure 4.2: Imageplots of foreground intensity for the BioC07 dataset, with replicates of sample A in the top row, and sample B in the bottom row. Red and yellow denote high and low intensity regions respectively. Clear spatial artefacts can be seen for arrays 1, 3 and 6.

Whilst such spatial artefacts are relatively commonplace for other microarray technologies (see Figure 1.3), they have not previously been seen for Illumina data as it is assumed that the random arrangement of beads and robust summary mechanism will reduce the effect of such artefacts. I now use some of the plotting facilities from within beadarray to see how the outlier removal approach used by Illumina deals with these artefacts.

4.2.3 Outlier Detection

The `findAllOutliers` function was used on each of the 10 arrays separately. The result of the function is a vector of numeric values, each of which indexes a particular bead on a given array. For the BioC07 dataset, the number of outliers expressed as a percentage of beads on the array are 27.55, 5.44, 6.00, 5.53, 5.19, 12.02, 4.88, 5.43, 5.48 and 4.07 for arrays 1 through 10 respectively. In other words, two of the arrays identified from Figure 4.1 as having different distributions are found to have the most outliers, whereas the other arrays have around 5% outliers. This suggests that a crude cut-off could be derived to identify poor quality arrays based on a higher than expected percentage of outliers.

After identifying the outlier beads using `findAllOutliers`, we can proceed to find the location of these beads on the array. We would hope that the summary method used by Illumina would exclude any beads lying within spatial artefacts from the analysis. Figure 4.3 shows the location of the outlier beads on arrays 1, 3 and 6 plotted using the `plotBeadLocations` function. These plots make the spatial artefacts on arrays 1 and 6 even more obvious. We can also see that Array 1 has a dense concentration of outliers over the entire array. However, it is initially quite puzzling why we do not see a spatial artefact in the bottom left corner for Array 3. If anything, a spatial artefact is suggested in the top of the array. The reason for the apparent discrepancy between Figures 4.2 and 4.3 for Array 3 could be the choice of scales used to create the plots. Figure 4.2 was generated after applying a $\log_2$ transformation to the data to compress them into a convenient scale for visualisation. Without this, the images would have been completely dominated by low intensity beads. However, the Illumina method for removing outliers was used for Figure 4.3, and this does not perform a $\log_2$ transformation to the data. Therefore the outliers we see by eye after a $\log_2$ transformation may not necessarily be the same beads picked as outliers from unlogged data. This will be explored in more detail later on in this chapter. For completeness, the TIFF images for arrays 1, 3 and 6 are shown in Figure C.1 in Appendix C. This figure was generated by adjusting the contrast and colour-balance of the original TIFF images to enhance our ability to identify the spatial artefacts. Obviously such adjustments are impractical for large-scale experiments.

Another factor to consider is the location of beads that could not be decoded by Illumina. On Array 3 a large area in the bottom left corner is seen to coincide with undecoded beads (data not shown). In the bead-level data for the BioC07 dataset, these beads are present but have a ProbeID of 0. In beadarray these beads are not used in bead-summary analysis as it does not make sense to average over them, therefore they would not be highlighted as outliers.

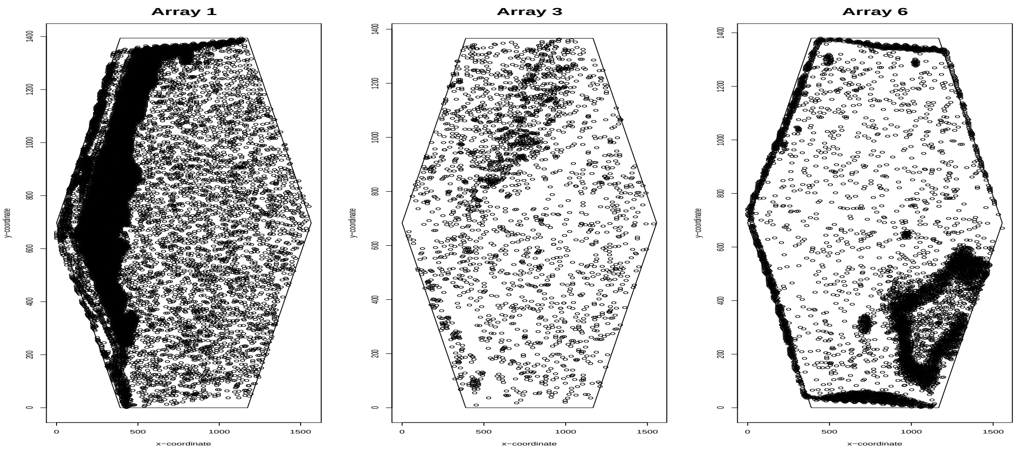


Figure 4.3: The locations of beads that are found to be outliers on arrays 1, 3 and 6, which were seen to have spatial artefacts in Figure 4.2. With the exception of Array 3, the locations of outliers correspond well with the spatial artefacts seen by eye. The outlier removal method implemented by Illumina was used, which excludes beads using a cut-off of 3 MADs from the median of the unlogged bead intensities for each bead type.

We will now proceed to analyse the bead-summary data for this example by running the function `createBeadSummaryData` on the background corrected bead-level data. This would have been the starting point for analysis if the raw data had been processed using BeadStudio.

4.2.4 Analysis of summarised BioC07 data

Figure 4.4 shows the summarised expression values and number of observations for all bead types on the 10 arrays. Both these matrices can be easily retrieved from the *ExpressionSetIllumina* object, as described in Chapter 3. Note that these plots have a much lower density as we now have 1,471 observations in each box, rather than the 49,777 in Figure 4.1. If we were looking at these data for the first time (for example, if the data had been processed using BeadStudio) we might think that the data do not need much normalisation. It is interesting that Array 1 does not look like such as extreme an outlier as it did in Figure 4.1. The median level of this array is slightly higher, which is not surprising by itself as these arrays were all hybridised and processed on different dates. Similarly, Array 6 also has higher median

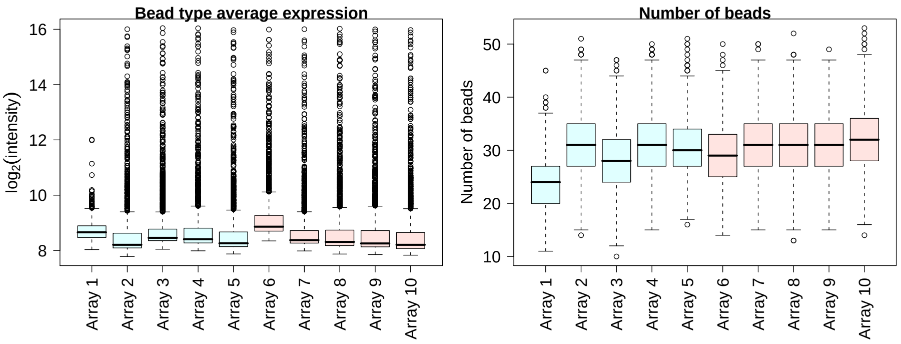


Figure 4.4: An overview of the BioC07 dataset after summarising the bead-level data. The summarised expression levels for all arrays are seen to be in good agreement, although arrays 1 and 6 have slightly higher medians. Array 1 also has fewer bead types with extreme high intensities. The number of beads after outlier removal are also shown for all arrays in the BioC07 dataset. The average number of beads for a bead type in a given array is generally around 30, although arrays 1, 3 and 6 have lower numbers of replicates. No bead type on any array has fewer than 10 replicates.

intensity. The IQRs of all arrays are roughly the same, although Array 1 now has a lower number of extreme high intensities.

This reduction in extreme high values for Array 1 can also be seen in a boxplot of the number of observations (see Figure 4.4). Whilst most arrays have around 30 observations for each bead type, Array 1 has around 24 observations on average, which is a noticeable decrease. We also observe that despite the severe artefacts on some arrays, no bead type in the dataset was left with fewer than 10 replicates after outlier removal. We now look at how comparisons between arrays might be affected by such spatial artefacts.

In Figure 4.5, MA-plots are shown for selected pairwise comparisons of Sample A and Sample B without having applied any normalisation. For the plots involving Array 1, many points are found away from the $M = 0$ line, indicating genes that have greater intensity in Array 1 than the other array, or vice-versa. It is unlikely that any normalisation approach would be able to fix this problem, as most methods attempt to correct for curvature seen in the MA-plots. At the same time, MA-plots of other pairwise com-

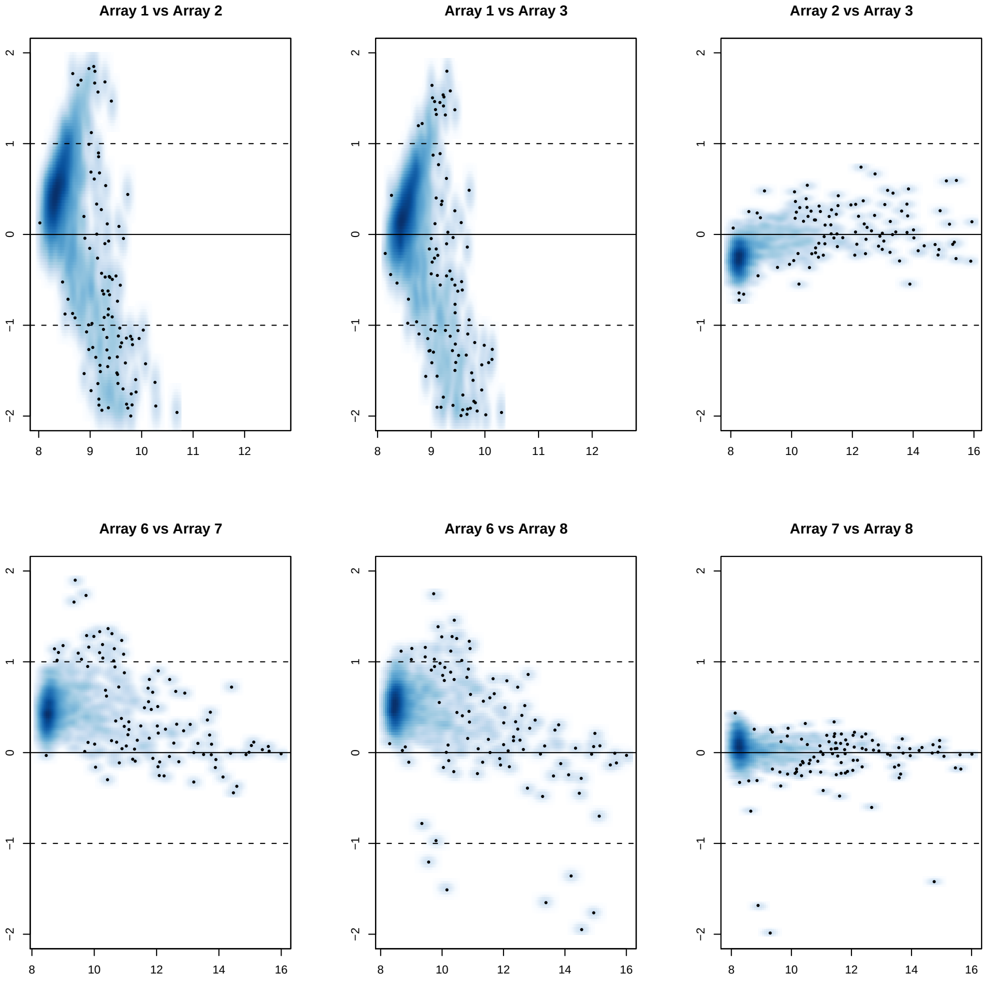


Figure 4.5: MA plots constructed using selected replicates of Sample A (top row) and replicates of Sample B (bottom row). Comparing Array 1 to arrays that have the same biological sample hybridised yields many genes with log-ratios away from 0 in a non-linear fashion. This trend is not seen for other comparisons of Sample A. The log-ratios generated using Array 6 are more variable than other comparisons of this sample and also systematically greater than 0. However there are many normalisation schemes that might correct for this.

parisons of this sample do not appear to show differential expression. Given our previous knowledge from the bead-level data, it would be a reasonable assumption that the spatial artefact on this array is having a dramatic effect on the observed intensities. Correlation coefficients of Array 1 with the other replicates of sample A are 0.039, 0.035 and 0.039 respectively, giving further indication that this array is unreliable.

The replicates of Sample B are very consistent. Despite Array 6 having a spatial artefact, it appears to agree well with other replicates, having correlations between 0.77 and 0.79. In Figure 4.5 when comparing Array 6 to Arrays 7 and 8, the M-values are shifted above 0, which is concordant with our observation that this array has generally higher intensity. Also, the M-values involving Array 6 have a slightly larger range than those for other replicates of Sample B. Such trends should be removed by methods which scale each array to have the same average intensity or overall distribution.

4.2.5 A simple differential expression analysis

In order to demonstrate the benefit of removing problematic arrays from the analysis, I consider an example of finding DE genes between the two sample types A and B, which was not the intention of Stranger *et al.* (2005), but is a common goal for microarray analysis. Using limma, the linear model given by (2.3) was fitted to the quantile normalised expression values, with the design matrix $\boldsymbol{X}$ being the 5×2 matrix,

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

Thus, the two coefficients for each gene ($\boldsymbol{\alpha}_g$) estimate the expression level of that gene in samples A and B respectively. The contrast between sam-

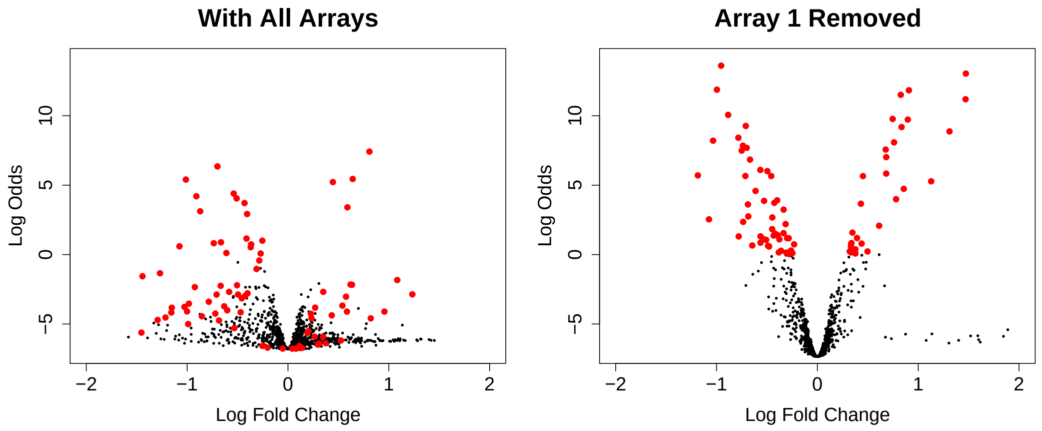


Figure 4.6: Comparison of the volcano plots produced for a differential expression analysis involving all arrays (left) and with Array 1 removed (right). The y-axis shows a measure of evidence for differential expression (log-odds) and the x-axis shows the estimated coefficients from the linear model. Red dots indicate genes with positive log-odds (roughly corresponding to greater than 50% chance of being DE) in the analysis that excludes Array 1. Removal of Array 1 from the analysis is seen to improve our ability to detect DE genes.

ples A and B was estimated using the contrast matrix $\boldsymbol{C} = (1, -1)$ and followed by empirical Bayes shrinkage of variances (the `eBayes` function) to give moderated t-statistics and log-odds for each gene. Separate analyses were performed before and after removing Array 1 from the analysis (by modifying $\boldsymbol{X}$ accordingly). Figure 4.6 shows the volcano plots for the analysis with and without Array 1. This is a common plotting tool as it displays the evidence for differential expression (log-odds) on the y-axis and estimated log fold-change on the x-axis. Ideally, we should see any DE genes having both high log-odds and fold change. It should be noted that we have no prior knowledge about how many DE genes to expect in this analysis.

The base of the plot is quite wide, indicating that while many genes that have high fold change, they do not have much evidence for being DE. Removing Array 1 produces a much more desirable picture, with many genes having higher log-odds than before. Red dots indicate genes with log-odds scores > 0 (greater than 50% chance of being DE under the assumptions of the model) in the analysis that excluded Array 1. By removing Array 1 from the analysis, we have removed noise from the experiment and increased our ability to detect DE genes in this simple example.

Rather than removing Array 1 completely, we could also consider downweighting it in the analysis. Such an approach is described in Ritchie *et al.* (2006). Computing array weights (using the `arrayWeights` function in limma) for our 10 arrays gives values of (0.02, 1.93, 1.23, 1.68, 1.64, 0.72 1.09, 2.21, 1.82 and 2.53), suggesting that Array 1 should be given little influence in the analysis.

4.3 Observations from the HapMap dataset

I now consider the bead-level data for the HapMap dataset, consisting of 15 SAMs and 270 individuals. Such large-scale experiments are made affordable by the high-thoughput nature of Illumina arrays. The `readIllumina` function has a `path` argument allowing multiple directories to be specified. Thus, the files in these directories can be read with the same function call and stored in a *BeadLevelList* object, allowing the same diagnostic plots to be accessed as for the smaller dataset already described in this chapter.

Figure 4.7 shows the percentage of outliers detected on a random sample of 200 arrays using Illumina's method, and after a $\log_2$ transformation. This percentage is broken down into beads that were removed due to being below the median, and those removed for being above the median. The total number of outliers is seen to be less than 5% for most arrays, with the majority of these outliers being above the median. By removing the outliers on the $\log_2$ scale, we get a more symmetric distribution of outliers above and below the median, as the $\log_2$ transformation reduces the spread of the data. Hence, the percentage of outliers is roughly the same in both directions. Across the entire set of over 1,400 arrays, the number of observations of a given bead type on an array was around 30 with 25th and 75th quantiles of 27 of 35 respectively. Furthermore, the lowest number of replicates on any array was 8. These quantities were similar regardless of whether outliers were removed on the $\log_2$ or raw scale.

Figure 4.8 gives an impression of how the overall intensity level of an array changes with the date of hybridisation. The median intensity of each

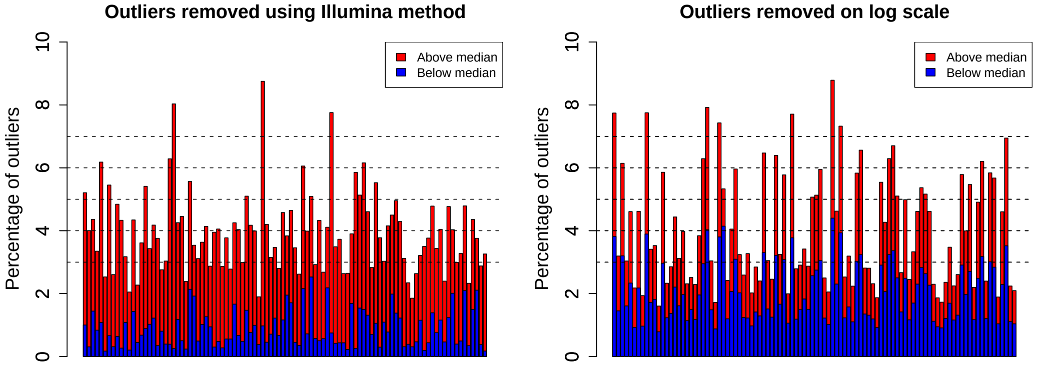


Figure 4.7: The percentage of outliers for a random selection of 200 arrays from the HapMap dataset, with outliers below and above the median shown in blue and red respectively. On the left, outliers were removed on the unlogged scale using a 3 MAD cut-off, and on the right a $\log_2$ transformation was applied prior to removing outliers. Without applying a $\log_2$ transformation, we find many more outliers above the median than below, whereas we get a more even distribution of outliers after a $\log_2$ transformation.

array was calculated and then plotted according to which SAM the array belonged to (i.e., each box has 96 observations within it). Clear differences can be seen between SAMs. For instance, arrays on SAM 1269941 have higher median values than arrays from other SAMs. The median levels on this SAM are about 1 unit on the $\log_2$ scale higher, or double on the original scale. This was the first SAM to be hybridised and some time had elapsed before hybridising the other SAMs. Therefore it is possible that some of the scanning conditions had changed before running the other SAMs. The SAM with the consistently lowest median is 1318811, with genes on this SAM measured to be about half the intensity of other SAMs (1 unit on the $\log_2$ scale). Clearly this needs to be accounted for in an analysis of gene expression levels and when trying to combine replicates of the same sample hybridised on different SAMs. Interestingly, the background estimates for this SAM were not noticeably lower. In fact, the background estimates across all arrays showed remarkable consistency (data not shown).

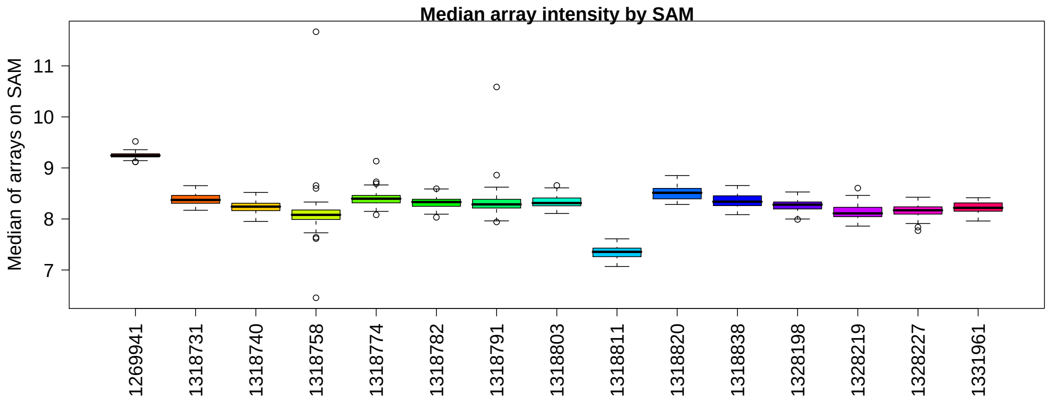


Figure 4.8: An overview of the overall array intensities for the HapMap dataset. The median intensities were calculated for each of the 1400 arrays and plotted according to which SAM (indicated by a 7 digit number) the arrays belong to. Clear differences are seen between the chips.

4.4 Discussion

The beadarray software described in Chapter 3 was used in conjunction with a specially obtained dataset to explore some of the general characteristics of Illumina data for the first time. All other publications using Illumina data take these steps of image analysis, background correction and summarisation as given.

The extra information given by bead-level data appeared to be beneficial in a small example with two samples replicated five times each. Using beadarray to make imageplots we were able to see significant spatial artefacts on three of these arrays (arrays 1, 3, and 6). The main design features of Illumina arrays, namely the randomisation of beads on an array and robust summarisation, are expected to cope with such artefacts. Indeed, the location of beads that would be called as outliers by Illumina are generally consistent with the regions seen by eye. The same artefacts were visible on the original TIFF images after some image manipulation. Performing the same manipulations on a large dataset would be time-consuming, whereas the `imageplot` function can provide the same information and does not require the storage of each TIFF image. In the BioC07 dataset, the two arrays with the most obvious spatial artefacts were are found to have higher numbers of outliers in total. This suggests that the total number of outliers on an array (easily calculated by beadarray) could serve as a proxy for the quality of an array. When looking at a much larger dataset, we found that the number of outliers on an array was generally around 5%, and that observing more than 10%, such as seen twice in our small example of ten arrays, is extremely uncommon. The calculation of outliers is only possible with bead-level data, although the number of observations reported as part of the bead-summary data could also be used for QA purposes.

Array 3 from the BioC07 dataset was seen to have an obvious region of lower intensity beads. However, not all the beads inside this region were called as outliers by Illumina. This is partly due to Illumina using an unlogged scale to call outliers and a symmetrical MAD cut-off. Also many beads in this region could not be decoded by Illumina and were not used in the outlier calculations. It would be interesting to investigate if undecoded beads are often associated with areas of low intensity.

Due to the distribution of intensities on the unlogged scale being skewed towards lower values, Illumina's outlier calling method is more likely to chose beads higher than the median as being outliers. As expected, when looking at the number of outliers across all 1,400 arrays, more outliers were called above the median than below. An implication of this result is that by including more beads with lower intensity, the bead type summary values could be underestimated. Applying a $\log_2$ transformation prior to outlier removal was seen to remove roughly the same amount of outliers above and below the median. Thus the choice of scale used to create outliers has an impact on the beads excluded from the analysis and the variability of the resulting summary values. If further downstream analysis is planned on the $\log_2$ scale, which is usually the case for microarray analysis, then it might be advisable to remove the outliers on this scale too.

We attempted a differential expression analysis between the two sample types using all 10 arrays in the linear model. It was found that the intensities on Array 1 (with 27.55% outliers removed) were dramatically different to other replicates of the same sample, whereas Array 6 (with 12.55% outliers) was more comparable to other replicates. Array 1 had to be excluded in order to find a greater number of DE genes between the two samples. It could be that we are able to tolerate a certain percentage of outliers on an array before the results become compromised. This will be investigated further in Chapter 5.

In this experiment, each biological sample of interest was replicated five times, which is quite a large number of replicates for a gene expression study. We therefore had the flexibility to remove one of the replicates before the analysis. However, for some genotyping studies it is common to hybridise each sample only once. Additionally, large-scale expression studies that involve many hundreds (or thousands) of samples may not have the ability to perform replicate observations. Therefore, it is essential that outlier arrays can be detected so that the conclusions of the experiment are valid.

Despite the varying numbers of outliers found on the arrays and possibility of spatial artefacts, the lowest number of replicates for any bead type in the HapMap dataset was 8. Even Array 1 from the BioC07 dataset still had an average of over 20 replicates for each bead type despite the spatial artefact. This is still a reasonable number of replicates compared to other technologies. However, it should be noted that lower numbers of replicates will occur by chance on the higher density BeadChips.

The results of estimating foreground and background were investigated for the BioC07 dataset. It was found that the foreground values themselves had a very narrow range of values that are skewed towards lower intensities. The background estimates were remarkably similar both within and between arrays. This seems to be intentional since Illumina pick so few pixels from a relatively large area to estimate the background. Such low background estimates, such as calculated using the *morph* method, have previously been found to be beneficial for gene expression studies (see page 18). Therefore, there is a hope that Illumina's method might work well. Unfortunately, with this dataset we are unable to quantify the effect that increased variability has on the analysis of Illumina data. Usually there is a trade-off between variance and bias, whereby some variance can be tolerated as long as the bias is small, or vice-versa. We cannot judge this using this dataset as no truth is known about the probes used in the experiment.

In this chapter I have demonstrated how *beadarray* can be used for QA and low-level analysis for large-scale experiments and have presented some findings about the impact of the image processing steps used by Illumina. It should be noted that this experiment used a version of Illumina expression arrays that was never made commercially available. Even so, it should serve to demonstrate that Illumina technology is not infallible and that Illumina data should be treated with the same careful QA principles as other microarray data. In particular, such large-scale experiments need careful planning and analysis to account for systematic trends that may arise when running many samples over an extended period of time. Although the results obtained within a batch may be similar, comparing batches may not be straightforward. The possibility of outlier arrays also needs to be accounted for. This is entirely consistent with experiences from other microarray technologies.

Chapter 5

Analysis of an Illumina spike-in experiment

5.1 Introduction

In previous chapters, I have introduced the Illumina BeadArray technology that uses randomly organised arrays of beads. As described in Chapter 2, analysis of Illumina data is routinely carried out using BeadStudio. Whilst this software provides an intuitive graphical user interface, there is no control over image processing and the details of the algorithms used by the software are not easily visible to the user. Such processing steps are known to be critical for other microarrays. In Chapter 3, I introduced the beadarray software package that allows a thorough investigation into the pre-processing of Illumina data to be performed.

A recent high-profile study found that data from Illumina expression arrays had good reproducibility and agreement with Affymetrix data (MAQC Consortium, 2006). A similar conclusion was reached by an earlier study (Barnes *et al.*, 2005). However, in comparison to Affymetrix, which is an established technology, there is a lack of in-depth literature on the low-level analysis of Illumina data. Information about how to obtain bead-level data has only recently been released and these data cannot be generated retrospectively. Therefore, no publications have taken the processing of bead-level data into account apart from our own preliminary investigations (Dunning *et al.*, 2006b) and the work presented in Chapter 4. Although the two dilution studies for Illumina (Barnes *et al.*, 2005; MAQC Consortium, 2006) made their data publicly available, these data were the summarised output from BeadStudio. Therefore, there is no publicly available dataset for which the bead-level data may be obtained and for which there is some expectation about the results. Such datasets are available for Affymetrix and have allowed researchers to understand more about the technology and to develop and evaluate analysis methods (Cope *et al.*, 2004).

The focus of (Barnes *et al.*, 2005; MAQC Consortium, 2006) was to compare the results of Illumina to other platforms, rather than to discuss the optimal processing of Illumina data. To make the comparisons more realistic, the current best-practice guidelines were used to process each microarray technology. In the case of Affymetrix, these methods have been developed and refined over a number of years, perhaps giving Affymetrix an unfair advantage over Illumina in such comparisons. It is possible that better understanding of Illumina data could give a more accurate comparison to other technologies.

This chapter is divided into two parts and describes a specially designed experiment (a *spike-in* experiment) performed on Illumina arrays. The first part of the chapter describes how the the spike-in experiment was used to investigate the background correction, summarisation, and normalisation of Illumina data, and was published in Dunning *et al.* (2008a). I also demonstrate how to use bead-level data to derive improved measures of differential expression and how probe annotation also has an effect on the observed intensities.

Around the same time as the publication of Dunning *et al.* (2008a), a new transformation method for Illumina data was published (Lin *et al.*, 2008). The validation of this method was done using Barnes *et al.* (2005), although the authors suggest that a spike-in experiment would have be useful in further validating their method. Thus, the second part of this chapter describes how I used the spike-in experiment to validate this new method (Dunning *et al.*, 2008b).

The contributions of other authors to the work presented in this chapter are clearly stated.

5.2 Control experiments for microarrays

There is a wide range of statistical tools available for the analysis of microarray data and knowing which methods work best is difficult without an effective means of comparison. It is common practice when developing a new analysis technique, or when evaluating different microarray technologies, to use a dataset where there is some expectation about the results. Such validation methods are crucial in the development of new algorithms (Allison *et al.*, 2006).

Dilution experiments such as Barnes *et al.* (2005) and MAQC Consortium (2006) are easy to perform, and if the samples being mixed are chosen appropriately, the analysis can yield many DE genes. One disadvantage of this approach is that we cannot be certain of which genes are expected to be DE. However, the ability of probes to respond to the change in concentration of the samples can be used as a criterion to judge the precision of different methods or technologies (Holloway *et al.*, 2006).

Alternatively, a spike-in experiment may be used whereby particular genes are added at known concentrations on each array. The genes chosen are usually artificial, or not found in the genome of the organism under investigation. The concentration at which the genes are added may vary between arrays, which allows the change in expression level between arrays to be predicted. Different methods may then be assessed to judge how well they recover the predicted change in expression, usually as a measure of bias and variability. Other than the spiked genes, the concentration of all other genes remain the same. Therefore, a measure of the number of false positives may be obtained by seeing how many non-spiked genes are called as DE.

Affymetrix technology has benefitted greatly from the use of a publicly available dataset to development new methods. One such spike-in experiment was performed by Affymetrix themselves, with the full data available online for other researchers. The design of the experiment is described in detail in Irizarry *et al.* (2003a). To summarise, 14 distinct spike genes were added to each array at concentrations 0, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512 and 1024pM, with the concentration of a particular gene on an array assigned by a "Latin square" system. This ensures that each spike gene on a given array is added at a different concentration. This dataset was used to develop the RMA method to summarise probe level intensities, which offered significant improvements over the default method implemented by Affymetrix (MAS5). The main difference is that RMA produces more precise expression measurements, at the cost of increased bias. This method is now widely used, although subsequent improvements have been proposed to account for sequence-specific intensity differences (Wu and Irizarry, 2005). The ability of such spike-in experiments to compare methods is exploited by the affycomp (Cope *et al.*, 2004) package, which provides a number of graphical and statistical benchmarks for the dataset. Users are able to submit to a server the results of processing the spike data using their own algorithm. A series of metrics are then calculated and used to compare to other method submissions. One potential disadvantage of affycomp is that the purpose of the package is to compare summarisation and normalization methods. Therfore, for simplicity, it uses fold-changes to call DE genes, rather than some of the more sophisticated methods that are used in practice. This point has recently been addressed by the creation of a AffyDEComp Bioconductor package that compares different methods of detecting DE genes (Pearson, 2008).

Another drawback of the original Affymetrix spike-in experiments is the small number of spiked probes. The "Golden Spike" experiment (Choe *et al.*, 2005) was an effort to create a control dataset with many expected DE probes (1,331) and also included many probes (2,535) spiked-in at equal concentrations. Two conditions were used in the experiment (control and sample) and replicated three times. However, there have been several criticisms in the literature regarding the analysis of these data and the experimental design (Dabney and Storey, 2006; Irizarry *et al.*, 2006).

In summary, there is plenty of evidence of the utility of control experiments in the literature, and in particular for Affymetrix arrays. Although some publications from Illumina described the use of spike-in experiments (Chudin *et al.*, 2006; Kuhn *et al.*, 2004), data from these publications were never made available. I now describe the spike-in experiment analysed in this chapter, which has now been made publicly available to accompany the publication of Dunning *et al.* (2008a).

5.3 The Illumina spike-in experiment

The Illumina spike-in experiment consists of eight customised Mouse-6 version 1 BeadChips hybridised with a complex mouse background. In addition to the ∼48,000 bead types included as standard, the bead pool for these chips was modified to include 33 bead types chosen to target 9 different bacterial and viral genes absent from the Mouse genome. These 33 bead types are referred to as *spikes* in this chapter and the remaining bead types on the array are referred to as *non-spikes*. Each array also had a number of standard Illumina controls, including 1,616 negative controls. Each BeadChip comprises six arrays and each array is made up of two strips on the chip surface. Similar to the Human6 chips described previously, Strip 1 interrogates targets from the curated MEEBO database (`http://www.microarray.org/sfgf/meebo.do`), and Strip 2 contains targets from other sources.

The spikes were added at concentrations of 1000, 300, 100, 30, 10 and 3 pM on the six arrays from the first four BeadChips. A further four chips were hybridised with spikes at concentrations of 1, 0.3, 0.1, 0.03, 0.01 and 0 pM. Unlike the Affymetrix spike-in experiment, the spikes on a given array in the Illumina spike-in were all added at the same concentration. Each concentration was allocated to the same position on all replicate BeadChips. For example, 1000pM was always array 1 on a chip and 300pM was array 2 and so on. Thus, when comparing the observed intensities of any given spike between array 1 and array 2 we would expect to see a fold-change of 3.33 on the original scale or $\log_2(3.33) = 1.74$ on the $\log_2$ scale.

The spike-in experiment was designed and scanned by Illumina. Raw data from the experiment, which includes the TIFF image and text file for each strip, are available online along with annotation information and supplementary materials (see Dunning *et al.* (2008a) for details).

5.4 Topics of investigation and methods

In this section I describe how the spike-in experiment was used to investigate various issues in the processing of Illumina data.

5.4.1 Image analysis and background correction

The function readIllumina was used to obtain the foreground (X_f) of each bead and corresponding background estimate (X_b). Background correction was then performed by the following methods:

- No adjustment - Use the estimated foreground ($X = X_f$) in the analysis (assume $X_b = 0$).
- Subtract - The estimated background is subtracted from the foreground for each bead ($X = X_f - X_b$).
- Normexp - A normal-exponential convolution model was fitted to the background subtracted signal ($X = X_f - X_b$) to adjust the intensities from each strip separately. In order for the code to run more efficiently, we took the background corrected intensities directly from the bead-level data (the option `useImages=FALSE` in `readIllumina`), and then performed *normexp* on these values using the implementation within limma.

5.4.2 Summarisation

Most analyses in this chapter used the background adjusted (see above), $\log_2$ transformed data from replicate beads on a given array and summarised these values using Illumina's default method. To look at how robust this method is relative to other summarisation methods (mean, trimmed mean removing 10% of highest and lowest intensities or median), we measured the bias for each method from simulated data, where varying numbers of outliers were added (from 0% to 40% in increments of 5%). The true values were assumed to be the means calculated from the original data. Data from a good quality BeadChip from this experiment were replaced at random by intensities at the saturation level (2^{16}).

By varying the number of outliers, we can roughly assess the break-down point of Illumina's summary method. Data for each simulation were summarised on the original and $\log_2$ scale. Bias was computed by subtracting the summary values obtained from the simulated BeadChip from the the summary values obtained from the original data for each probe on each array. Per array, per probe variances were also calculated within each simulated dataset. This simulation study was carried out by Dr. Matthew Ritchie.

5.4.3 Normalisation

The $\log_2$ summarised data were quantile normalised as in Barnes *et al.* (2005). This approach is reasonable given that the majority of genes do not change between arrays, and hence the distribution of intensities on different arrays should be the same. Background normalisation (BGN) (see page 31) was carried out on the non-normalised, background subtracted data by subtracting the average value of the negative controls on each array output by BeadStudio, from the summarised intensities of the non-control probes. M- and A-values were calculated to allow comparison with quantile normalised results.

5.4.4 Linear models and contrasts

The limma Bioconductor package was used in order to assess differential expression. Bead-level data were created using different background correction methods and then summarised to give an expression matrix of $g = 1, \ldots, \sim 48,000$ rows and $k = 1, \ldots, 48$ columns, each row being a bead type and each column an array (12 spike concentrations, each of which was replicated four times). The linear model $E[\boldsymbol{y}_g] = \boldsymbol{X}\boldsymbol{\alpha}_g$ was used where $\boldsymbol{y}_g^T$ is the gth row of quantile normalised expression matrix, $\boldsymbol{X}$ is a 48×12 design matrix defined to denote the concentration of spikes on each array and $\boldsymbol{\alpha}_g$ is a vector of coefficients to be estimated for each probe at the 12 different concentrations. The function `lmFit` in limma was used to fit this model. The contrasts of interest are given by $\boldsymbol{\beta}_g = \boldsymbol{C}^T\boldsymbol{\alpha}_g$ is a contrasts matrix created to make all pairwise comparisons between concentrations (e.g. 1000pM vs 300pM, 300pM vs 100pM etc). After empirical Bayes variance shrinkage, the moderated-t statistics and log-odds scores for each contrast were analysed separately to assess the performance of different background correction methods.

A second series of linear models was fitted to take the variability of bead types into account. We now assume that $\text{var}(y_{gk}) = \sigma_g^2/w_{gk}$ where w_{gk} is a weighting factor for bead type g on array k. Weights $w_{gk} = 1/s_{gk}^2$, where s_{gk}^2

is the sample variance calculated using the standard error and the number of observations of bead type g on array k, were used. Using inverse variances as weights gives less influence to observations with higher variability in the linear model. The coefficients, $\boldsymbol{\alpha}_g$, were estimated using weighted least squares and contrasts, $\boldsymbol{\beta}_g$, were calculated as before. This weighted approach will be referred to as a *weighted log*$_2$ *analysis* and can be done by setting the `weights` argument in `lmFit`. The variances required for this analysis are retrieved using the `getVariances` function in beadarray.

5.4.5 Annotation

The sequences for all probes used in the experiment were acquired after correspondence with Illumina. This included the sequences for the spikes and control probes, which are not usually part of the Illumina annotation files. Probe sequences were BLASTed and BLATed against the corresponding mouse genome and transcriptome, which included UCSC Genome Browser (Kuhn *et al.*, 2007), RefSeq, and GenBank transcripts. The subsequent annotation and probe classification were performed with a Perl script, comprising BioPerl modules (Stajich *et al.*, 2002), and relied on transcriptomic annotation tables downloaded from the UCSC Genome Browser. The script to perform this reannotation was written by Dr. Nuno Barbosa-Morais (Barbosa-Morais *et al.*, 2008). The resulting table supplements the annotation information supplied by Illumina by giving details of where each probe sequence was found to map in the genome and the matching transcripts. Using the reannotation information, we were able to assign each probe sequence to various broad categories according to the quality of the match. I now describe each of these categories.

- Intronic - The probe matches to an intronic region of a gene.
- Intergenic - The probe matches to a region without genes.
- Unreliable - The probe perfectly matched a known transcript, but that transcript could not be aligned to the genome.
- Mismatch - The probe did not perfectly match a transcript, but it is likely (based on BLAST criteria) that a transcript match can be found.

- No Match - Probe is not likely to match any region of the genome.
- Multiple Match - Entire probe sequence is likely to match to more than one genomic region.

The impact of these annotation assignments was assessed by boxplots of all the summarised intensities on an arbitrary array, grouped according to the category of the probe sequence.

The possible relationship between probe composition and observed intensity was investigated as follows. We defined A, C, G and T to be matrices of binary values with $j = 1, \ldots, \sim 48,000$ rows and $p = 1, \ldots, 50$ columns to represent the sequence of each probe, where $A_{jp} = 1$ if the sequence for probe j contained an "A" at position p, or 0 otherwise. The total number of As (a_j) in the sequence of the jth probe is simply $a_j = \sum_{p=1}^{50} a_{jp}$. The total number of Cs (c_j), Gs (g_j) and Ts (t_j) were defined in a similar fashion. The GC content for probe j was then defined as $g_j + c_j$.

We then plotted the normalised intensities of the Strip 1 probes on a given array in terms of their a_j, c_j, g_j, t_j and GC content. Similarly, for a particular contrast in the differential expression analysis, we ranked the same probes according to their log-odds scores and plotted probes with the same GC content together.

The linear model $E[\boldsymbol{y}_k] = \boldsymbol{A}\boldsymbol{\alpha}_k + \boldsymbol{C}\boldsymbol{\beta}_k + \boldsymbol{G}\boldsymbol{\gamma}_k$, was fitted to the intensities and variances of the kth array to estimate coefficients $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ representing the effect of having an A, C or G at each position, relative to having a T at that position.

5.5 Results

5.5.1 Bead-level issues

Figure 5.1 shows the raw foreground intensities and local background estimates for all beads on each strip from a typical BeadChip. We see that the signal on Strip 1 is generally higher and has a greater dynamic range than the signal from Strip 2. This might be expected since Strip 1 contains probes

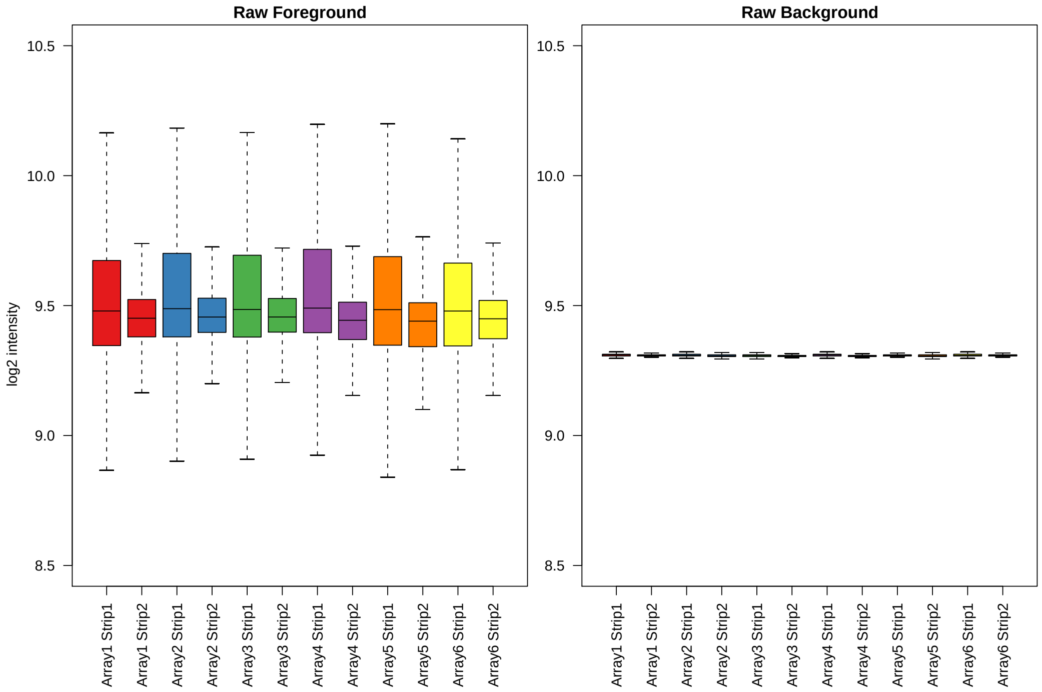


Figure 5.1: Raw foreground and background intensities for each strip on a typical BeadChip. Each BeadArray is made up of two strips (colour-coded) on the chip surface. The consistency of foreground and background signals between arrays is evident from this plot, as is the tendency for beads from Strip 1 to have higher intensities than those from Strip 2.

from a curated database, whereas Strip 2 contains probes targetting rarer transcripts. Concordant with observations from Chapter 4, the local background estimates show very low variability both within and between arrays, with a median of 634 on the original scale (9.3 on the $\log_2$ scale). The distribution of background signal is the same for all strips, despite differences in foreground signal.

Access to the bead-level data allowed us to identify a significant spatial artefact on one BeadChip in the experiment. An error in the scanning of the chip resulted in the the x coordinates of many beads on the left-hand side taking negative values. Consequently, the Illumina algorithm for image processing was unable to calculate foreground and background intensities for these beads and set their foreground intensities to zero. This problem affected between 4.4% and 7.3% of beads from each strip on this chip, with

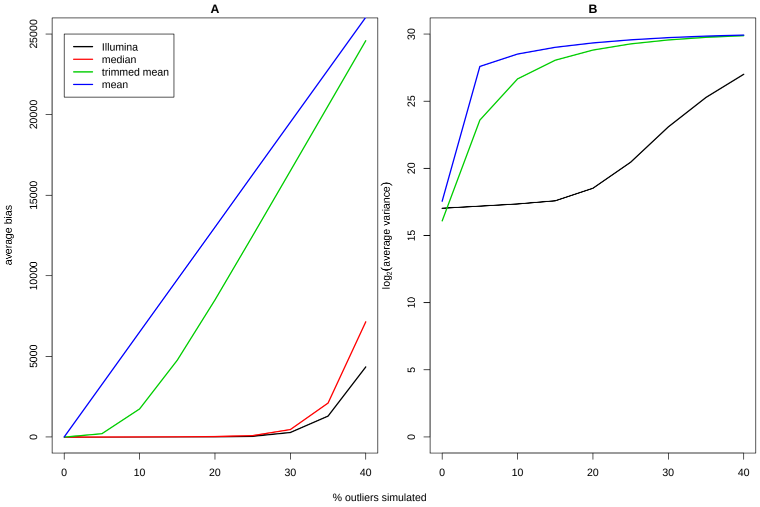


Figure 5.2: The average bias (A) and $\log_2$ variance (B) versus percentage of simulated outliers plotted for each summary method. In panel A, we see that Illumina's summary method can handle up to about 30% of saturated intensities before the bias starts to increase dramatically. The trimmed mean breaks down much earlier, at around 5%. The median is comparable to Illumina's method. Similar trends can be noticed in the variance (B).

the percentage of affected beads decreasing from the top to the bottom of the chip. These beads were subsequently removed as outliers by Illumina's summary algorithm. We note that after background correcting the bead-level data, the median percentage of beads with negative intensities on a strip was 0.32% with a 75th percentile of 0.56%.

Before proceeding with an analysis of summarised data, we investigated how robust Illumina's technology is to the spatial effects observed above. Arrays with varying numbers of saturated beads were simulated as described on page 80. These datasets aimed at assessing how many outliers could be tolerated by Illumina's default summary method compared to other methods (mean, trimmed mean and median).

Figure 5.2 shows the average bias and variance versus the percentage of outliers introduced. Illumina's summary method performs best overall, with the lowest bias (Figure 5.2A) and variance (Figure 5.2B). After around 20% of the bead intensities become saturated, the bias and variance start to increase. Using a trimmed mean that excludes 10% of the smallest and largest intensities, we see an increase in bias and variance after more than 5% of the beads are saturated. This is not surprising as the outliers are not simulated to be symmetric around the mean, which this method is suited to handle. The median offers similar robustness to Illumina's method. Similar results were obtained if the analysis was performed on the log scale, or if the data were censored at 0 rather than at 2^{16} (data not shown).

After applying the Illumina summary method, the median number of beads per bead type on an array was 36 with 25th and 75th percentiles of 31 and 41 respectively. The median number of beads per bead type per array removed as outliers was 1 with a 95th percentile of 4.

5.5.2 The effects of pre-processing on differential expression analysis

In Figure 5.3 we show the bead type means and variances calculated on the $\log_2$ scale, for the 33 spikes across all arrays in the experiment using three background correction methods (no background adjustment, background subtraction and *normexp*). These boxplots of non-normalised data are arranged according to the concentration of the spikes on the array. Different background correction methods are shown in different colours and data from each array are plotted in a separate boxplot. Given the design of the experiment, we would expect to see a decrease in observed intensity as the concentration of the spikes decreases.

Note that even though the data shown are not normalised, we can see that the replicates of the same array processed using the same method show low variability and only slight differences in the medians. The same trend can be seen for all background correction methods. A saturation effect can be seen between 300pM and 1000pM, as the increase in the concentration of the spikes is not reflected in a change in observed intensity. At 3pM, there is a clear difference between arrays with no background adjustment and arrays where a local background estimate has been subtracted. The linear relationship between spike concentration and observed intensity persists below 3pM for the background subtracted data, whereas without background ad-

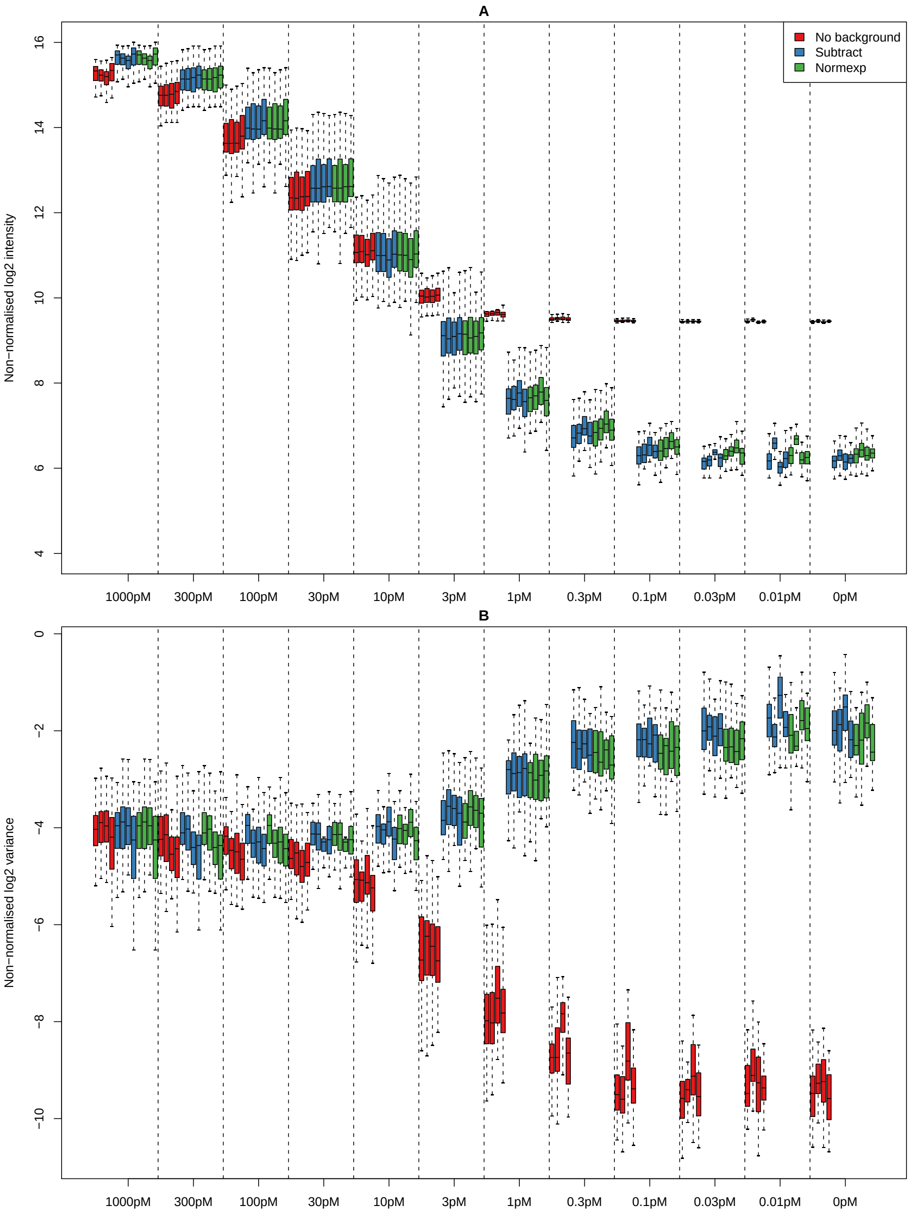


Figure 5.3: Boxplots of the means (A) and variances (B) for the 33 spikes on all arrays in the experiment after outlier removal. The boxplots are arranged in decreasing order of spike concentration, with different background correction methods labeled in different colours. The no background adjustment option shows dramatic attenuation in signal, which begins at a higher concentration than the other background correction options.

justment, an attenuation in signal is evident below 3pM.

The variances of each method are similar in the range 1000pM to 100pM. However, at 10pM we see a steep decrease in the variability for the non-adjusted data, whilst the background subtracted data show a slight increase. The rate of decrease in variability for the no background adjustment option is greater than the rate of increase in variability for the subtracted data. For concentrations of 0.1pM and below, the variance of the spikes does not decrease any further with decreasing concentration.

We now quantify how well the expected change in spike concentration is recovered by different background correction methods. In Figure 5.4 we show an MA-plot of the $\log_2$ transformed data from an array with spikes at 3pM and an array with spikes at 1pM. The data shown in Figure 5.4A were not background adjusted and we can see that the range of M-values is very low for all genes. The largest M-value we see is around 1.2 and the A-values are in the range 10 to 15. The observed log-ratios for the spikes are much lower than the expected value of 1.73. Figure 5.4B shows the same data after background subtraction. We see a wider range of M and A values compared to Figure 5.4A, and the log-ratios for the spikes are closer to the expected value on average. Notice that although these data have not been normalised, the non-spikes lie around M=0, indicating no differential expression.

In Figure 5.4C we show the same comparison for data that have been background subtracted and background normalised. This is equivalent to processing the raw data using BeadStudio's recommended settings. For visualisation purposes, and to compare with the other methods, we $\log_2$ transformed the background normalised data. The difference that this makes to the MA-plot is striking. We see a much increased range of M-values as A decreases. There are clearly a large number of genes that would be selected as DE if a simple cut-off approach were used, even though few genes are expected to change for this comparison. In addition, the log-ratios of the spikes are systematically over-estimated by this method. There are also a large number of bead types with negative intensities on each array after applying BGN, ranging from 11.2% to 49.4% (median 39.1%). These become missing values after a $\log_2$ transformation, which is undesirable in downstream analyses.

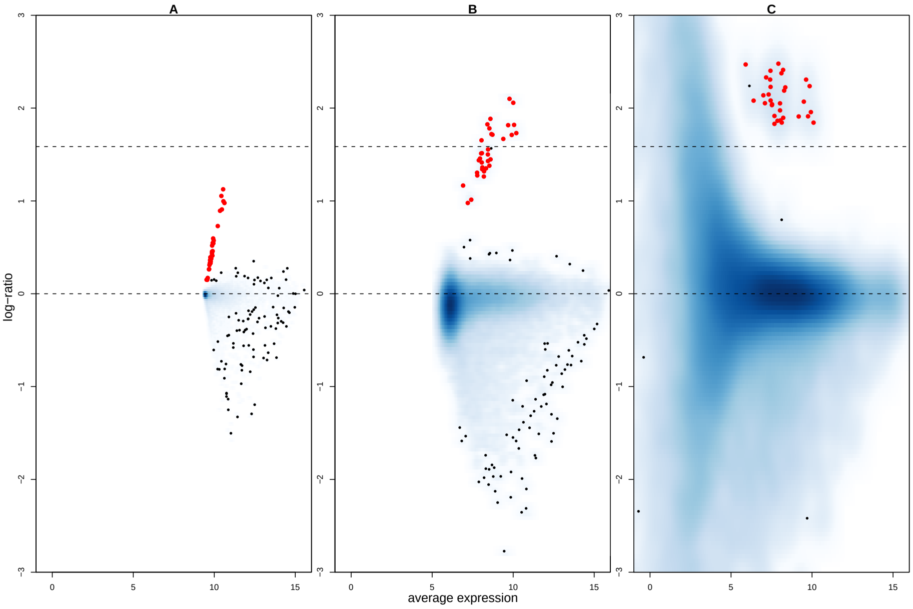


Figure 5.4: MA-plots comparing the bead-summary values for one array with spike concentration 3pM to an array with spike concentration 1pM. An increased density of points is indicated by darker shades of blue. Red points highlight the spike genes. The horizontal line at M=1.73 represents the intended log-ratio for the spike genes, and the line at M=0 is the desired level for the remaining non-spikes. Each panel shows the data processed using different background correction methods. Panel A shows the data with no background adjustment, while in panel B local background has been subtracted and in panel C the data have been background subtracted and background normalised. When the data are background subtracted, the range of M and A-values increases and the spike genes are closer to the true value than for the non-adjusted data. BGN (see page 31) produces the most variable M-values and over-estimates the M-values for the spikes.

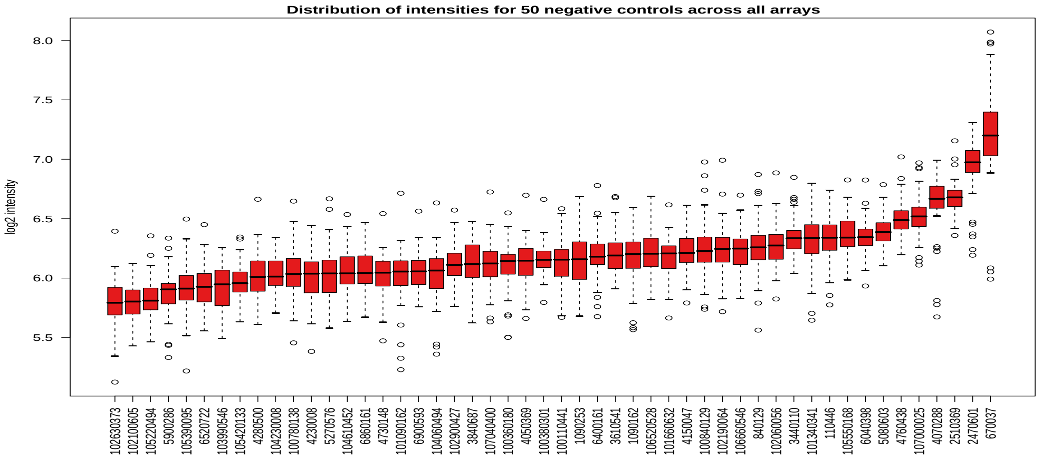


Figure 5.5: The distribution of background subtracted and summarised intensities for 50 negative controls across all arrays in the experiment, ordered by increasing median. Each control is a bead type with a random sequence attached that should not hybridise to any target in the genome. Despite this, some controls clearly appear to show consistently higher intensity than others.

I also looked at the intensities of the negative controls that are used in the creation of background normalised data. In Figure 5.5, for 50 negative controls picked at random, we plot the (background adjusted) averaged values for each control over all arrays. As expected, each control shows intensities at the lower end of the values observed on an array (Figure 5.1). However, we do see some variation in median intensity between the different probes, with a greater than two-fold difference in intensity measured between the bead type on the far left and far right of the plot.

The linear modelling approach of Smyth (2004) was used to detect DE genes. Figure 5.6A shows the log-odds scores for the contrast between 3pM and 1pM and for data processed using different background correction methods. Separate boxplots are shown for the spikes and non-spikes (solid and transparent colour respectively) and results are shown for a standard linear model and a weighted $\log_2$ analysis (see Page 81). Outliers for the non-spikes are indicated by crosses. Background subtraction is seen to increase the log-odds of being DE for the spikes. Moreover, a greater distinction between the log-odds of the spikes and non-spikes is seen after background subtraction.

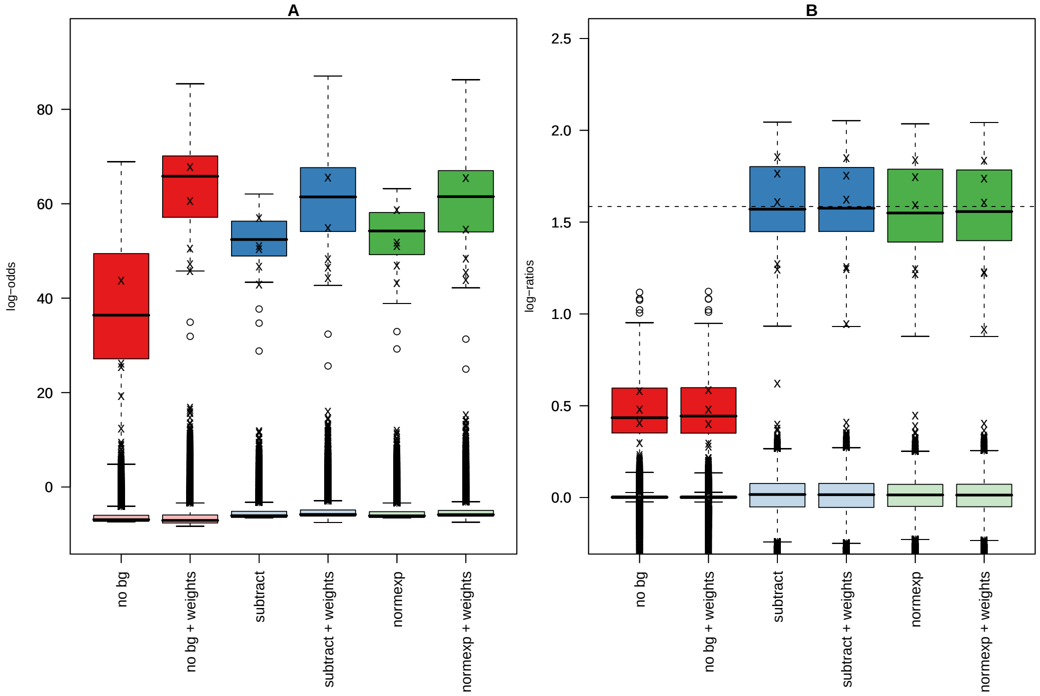


Figure 5.6: Boxplots of the log-odds scores (A) and log-ratios (B) obtained after fitting a linear model to all genes across all arrays in the spike experiment and making contrasts between 3pM and 1pM. A separate box is shown for each background correction method with a standard linear model and a weighted $\log_2$ analysis. Two separate boxplots are shown for each method and weighting scheme to indicate the log-odds scores for the spikes (bold colours) and non-spikes (transparent). The weighted $\log_2$ analysis improves the log-odds scores for the spikes without increasing the log-odds for the non-spikes, which represents an increase in power to detect true differential expression. In panel B, we show that the log-ratios for the spikes are under-estimated when the data are not background adjusted, whereas the background subtracted and *normexp* processed data recover values much closer to the true log fold-change (dashed line, M=1.73).

Five non-spikes are seen to have high log-odds in both linear model fits. These bead types were ranked amongst the spikes for all contrasts and had a similar expression profile to the spikes. Personal communication with Illumina revealed these probes are controls from the MEEBO database and not used in current Illumina chips. Generally, the spikes were the top ranked probes for each contrast, with very few false discoveries. The choice of background correction method was found to have little impact on the number of false discoveries (data not shown).

When using a weighted $\log_2$ analysis, we see an increase in the log-odds scores calculated for the spikes. At the same time, we do not see a substantial change for the non-spikes. The most dramatic increase in log-odds after a weighted $\log_2$ analysis is seen for the non-adjusted data. It is interesting to note that the log-odds of the different methods are more comparable under a weighted $\log_2$ analysis. We produced the same plot for all contrasts in the linear model (data not shown). The log-odds typically increased for all contrasts in the middle of the concentration range. However, for contrasts comparing 0.3pM to lower concentrations, we found little improvement, or sometimes a decrease in log-odds. Figure 5.6B shows the estimated coefficients for the comparison between 3pM and 1pM. For this contrast, we would expect the spikes to have a log-ratio of 1.73. For data processed without background adjustment, the highest log-ratio seen for the spikes is not much greater than 1. For the subtract and *normexp* methods, the log-ratios are centered around the expected value. For other contrasts (data not shown), the log-ratios were often underestimated by all methods, especially at high and low concentrations. Pairwise contrasts 30pM to 10pM, 10pM to 3pM and 3pM to 1pM accurately recovered the predicted log fold-changes. The non-adjusted data consistently produced the most biased values for all contrasts.

5.5.3 Probe properties and annotation considerations

We now repeat a similar analysis to Figure 5.3, but consider the behaviour of each spike separately. In Figure 5.7, we show the coefficients for each spike estimated by fitting the first linear model described on page 81 without weights to the background subtracted data. For clarity, each spike was labelled and coloured according to its target gene. A smoothed curve was

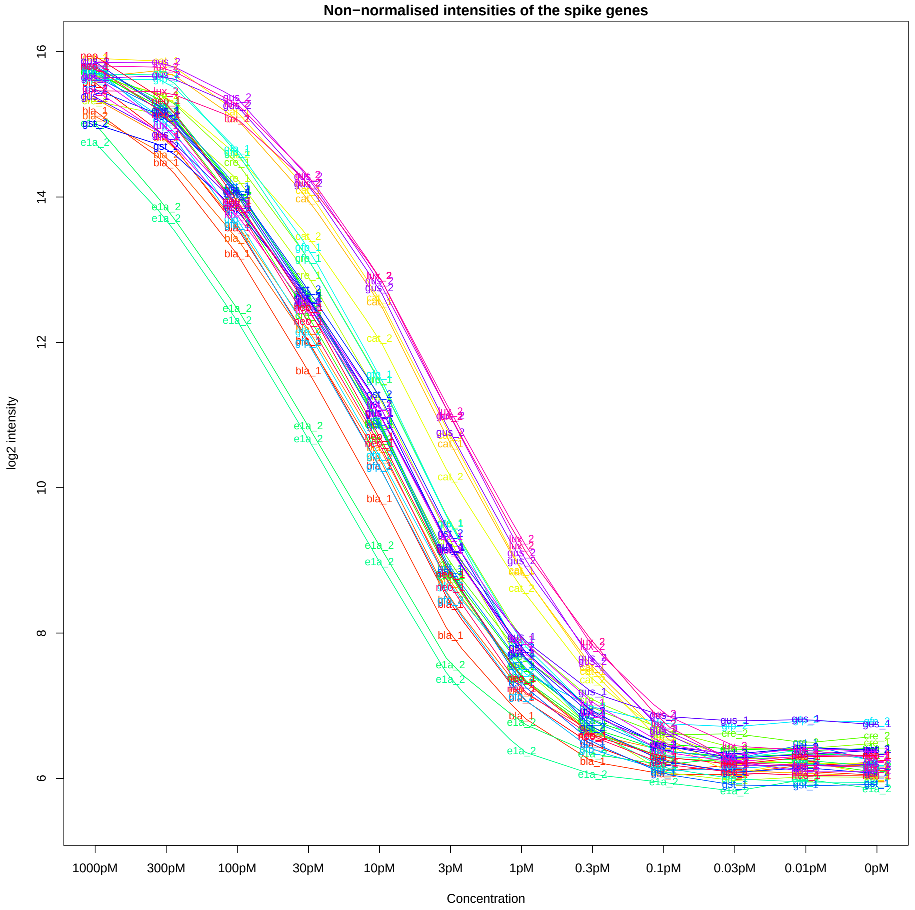


Figure 5.7: The $\log_2$ intensities for the 33 spikes on each array estimated using the linear model. Each spike is indicated by a different colour and line. Despite being added at the same concentration, consistent differences are seen between the spikes, for example, *ela_2* consistently has the lowest intensity.

	Strip 1	Strip 2
Total	23983	22022
Intronic	682	6094
Intergenic	78	2124
Unreliable	951	3031
Mismatch	7789	4178
No Match	173	1514
Multiple Match	849	817
Percentage of good probes	57%	22%

Table 5.1: Table showing the reannotation of the probes sequences for all non-spikes in the spike-in experiment. The 23,983 probes on Strip 1 and 22,022 probes on Strip 2 are divided into categories (see page 82) describing various annotation problems.

fitted to the coefficients for each spike. Note that bead types with the same target name (e.g. *ela_2*) have the same probe sequence attached, but are located on different strips.

We can clearly see different intensities for spikes at the same concentration. These differences are consistent across the concentration series. For example, *ela_2* always shows the lowest intensity at all concentrations, whereas *gus_2* and *lux_2* tend to have the highest intensity. This is consistent with a previous Illumina spike-in study that used the same spikes (Kuhn *et al.*, 2004). The intensity difference between the spikes is quite dramatic for some concentrations. For instance, at 30pM the highest intensity spikes are measured at 14, whereas the lowest intensities are at 11. It is also apparent that the spikes respond differently to the decrease in concentration. The *ela_2* spikes show a larger decrease between 1000pM and 300pM than the other spikes and the curve for these spikes flattens out at a higher concentration. Conversely, the spikes for *gus_2* are flatter for concentrations 1000pM to 100pM, but attenuate at lower concentrations than the other spikes. Some small differences can be seen for bead types having the same probe sequence, but hybridised to different strips.

After reannotation of all probe sequences, bead types were categorised according to where they map in the genome. The results are shown in Table 5.1

with the various categories explained on page 82. As expected, reannotation of the spikes and negative controls produced no genomic or transcriptomic matches. Some probe sequences among the non-spikes were found to match to intronic and intergenic regions. The percentages of bead types on Strip 2 with intronic and intergenic matches were 27.67% and 9.64%, respectively, compared to 2.84% and 0.32% on Strip 1. Any bead types that fall into these categories are potentially uninformative, as intronic or intergenic regions of the genome will be spliced out during transcription and therefore we would not expect any signal from these bead types. Bead types whose sequences match to more than one transcript could also complicate analysis as we might not be able to tell apart the contributions of the different transcripts to the observed signal. However, the impact of having a mismatch in the probe sequence, or mapping to an unreliable transcript, is less easy to predict.

Figure 5.8 shows the summarised intensities for all bead types on a representative Strip 1 array, grouped according to where the probe sequence for that bead type matched to. Bead types that had perfect matches to a known transcript are termed a "Good" match. As one might expect, bead types with intronic, intergenic or no matches are seen to have a lower median and IQR compared to bead types with good annotation. However, it is still possible to find intronic or intergenic matches with high intensity (say, more than 13 on the $\log_2$ scale). The same trends were observed on all arrays in the experiment.

In Figure 5.9 we see the normalised intensity of all non-spikes on Strip 1 of a particular array grouped according to how many A, C, T or G bases are found in the sequence attached to each bead type. Generally, we see that an increase in the number of As or Ts in the sequence is associated with a decrease in mean intensity, whereas an increase in the number of Cs or Gs results in an increase in mean intensity. Moreover, as the GC content increases, the variance of the bead types decreases. We also see that probes with either a G or C as the first base have a higher normalised intensity and lower variance relative to having a T at that position. We note that the distribution of GC content for the spikes was skewed towards higher GC content and showed little variation. Therefore, we did not have sufficient information to conclude a GC-related effect for these probes alone (data not shown). We could find no evidence for an effect of the GC content on the intensity of the

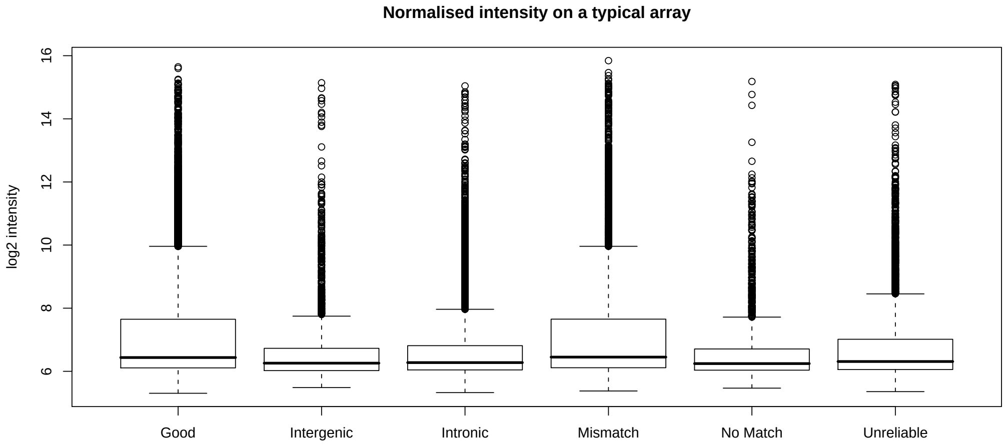


Figure 5.8: Boxplots of all bead types intensities on an arbitrary Strip 1 array grouped according to the reannotation of the probe sequence assigned to the bead type. Bead types with reliable annotation (the “Good” category) are seen to have higher intensity compared to others.

negative controls.

Ideally, we would like such probe effects to be removed when comparisons are made between arrays. In Figure 5.10 we show the ranking of the log-odds scores of the contrast shown in Figure 5.5 for all non-spikes on Strip 1. Clearly there is a preference for bead types with 18 to 21 GCs in the sequence to be higher in the list. On average, sequences with 19 GCs are 10,000 places higher in the list than sequences with 24 GCs. The “hump” seen in Figure 9 was evident for most contrasts in the linear model.

5.6 Discussion

In this chapter, I have described a dataset that can be used to perform a thorough investigation into the processing of Illumina data. Unlike previous data described in this thesis (e.g. Chapter 4), some degree of truth is known about a small subset of probes and how they should behave between arrays. I hope that the release of these data into the public domain will result in improved methodologies for Illumina data and encourage researchers to utilise

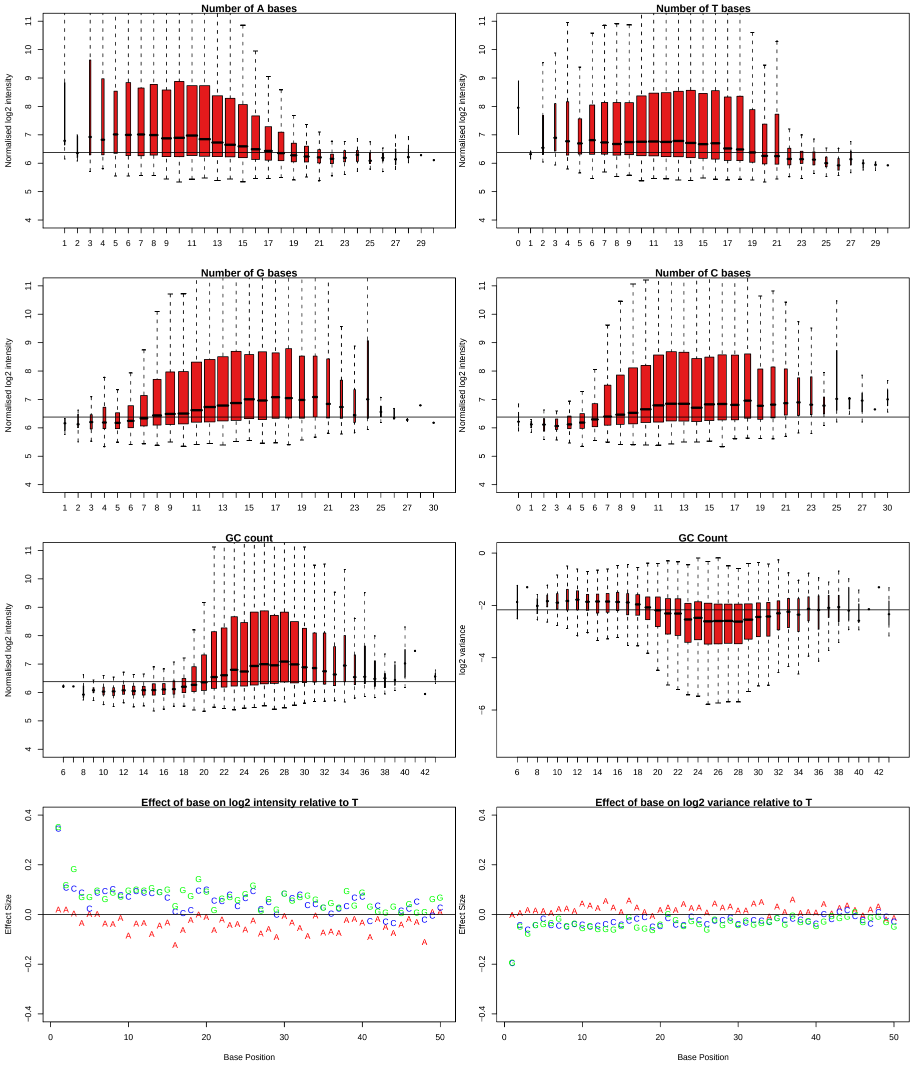


Figure 5.9: Normalised $\log_2$ intensities for all non-spikes on Strip 1 of a particular array in the experiment grouped according to the number of As, Ts, Gs or Cs in the sequence for the probe. The normalised $\log_2$ intensities and bead type variances are also shown in terms of GC content. The width of each box is proportional to the number of observations. Probes with higher GC content are shown to have higher intensity on average and a lower variance. Finally, estimated effect sizes are shown for each base position relative to having a T at that position. The normalised intensities are seen to be higher if a G or C is present at the first base in the sequence and have a lower variance. However, no other systematic trend is seen.

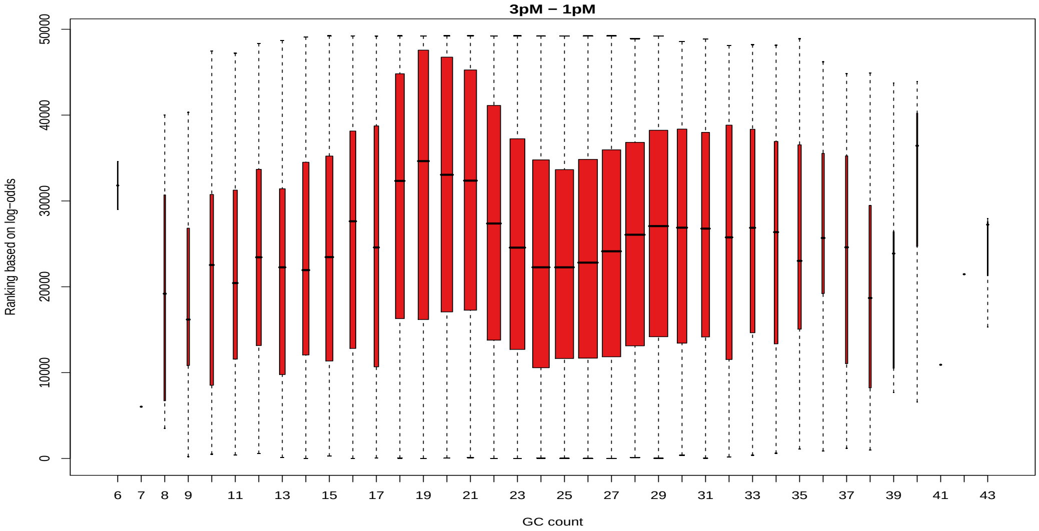


Figure 5.10: The log-odds ranking of all non-spikes on Strip 1 in the contrast between 3pM and 1pM aggregated according to the GC content of each probe. Probes with a GC content of 18-21 are generally ranked higher in the list. The width of each box is proportional to the number of observations.

the availability of bead-level data rather than the output of BeadStudio. I will now discuss some of the main points raised by this analysis of the dataset.

5.6.1 Data quality

The data produced using Illumina technology are widely reported to be of high quality. Naturally, we would still recommend careful QA of Illumina arrays and not to take high data quality for granted. Whilst initial QA using the raw data showed little variation between arrays, we were able to detect a consistent spatial effect on a particular BeadChip. However, we found that in this case, there was no impact on further analysis due to the random placement of beads and robust summary method used by Illumina. Although BeadStudio is capable of giving a good overview of an experiment, it may miss important artefacts on arrays, as spatial information is lost when the data are summarised.

We found that the two strips for each array show consistently different intensity distributions, with Strip 1 showing a wider range of expression values. It is important to determine if this difference arises due to annotation differences (i.e, curated probes on Strip 1, and other transcripts on Strip 2), or is a manufacturing issue with the chips. In the Version 2 whole genome BeadChips produced from 2007 onwards, the replicates of each bead type are spread between the two strips. Whilst the quality control steps used by Illumina ensure that any bead type is represented at least five times on an array, there is no guarantee that these replicates will be evenly distributed between the two strips. Clearly, the summary value could be affected by any differences in underlying intensity between the strips if a disproportionate number of replicates appear on one of the strips. The default options within the BeadStudio software combine the two strips for every array on a whole genome BeadChip. Therefore, any systematic difference between the strips would be hidden from the researcher if data are processed by BeadStudio, in which case analysing strips separately would be appropriate.

5.6.2 Local background estimation and subtraction

It is interesting to note the consistency of the estimated background for individual beads that is observed within and between arrays. As described in Chapter 4, the background estimation used by Illumina takes an average of the five dimmest pixels within a comparatively large area surrounding each bead. This gives a very low estimate for background that is related to the optical properties of the array surface rather than being specific to the sequence attached to each bead. In contrast, background estimation for two-colour arrays typically uses the mean or median value of pixels surrounding each feature, producing higher, more variable estimates. The approach Illumina uses is more akin to a morphological background estimation, and that has been shown to perform well for two-colour arrays (Ritchie *et al.*, 2007; Yang *et al.*, 2002b).

As suggested in Chapter 4, the predictability of the background signal could be used as a simple diagnostic to identify poor quality arrays on which the background level is considerably higher and more variable than usual. When analysing this experiment, we found that subtracting this low estimate of the background was beneficial for detecting DE genes. At low spike concentrations (around 1pM), the observed values for the spikes are close to the negative controls. Therefore, when comparing arrays with low spike concentrations that have not been background subtracted, the calculated log-ratios will be biased towards zero as the difference in spike concentration is obscured by the background noise. Background subtraction reduces this bias, although, as anticipated, we see an increase in variability after a $\log_2$ transformation of the subtracted data. The results of the simplest method of subtracting the background estimates are comparable to those of the model-based approach of *normexp*. This is due to the low percentage (less than 1%) of negative intensities produced using the subtract method, hence methods that avoid these negative values have little scope for improvement. This is encouraging for users without access to raw data who perform pre-processing using Illumina's default settings. Moreover, the current implementation of *normexp* in limma is too time-consuming to make it practical for the analysis of a large number of arrays.

5.6.3 Summarisation

In our simulations, Illumina's default summary method was able to handle around 30% of outlier beads before the estimates became noticeably biased. This provides a rough guideline on how much of an array can be corrupted before the analyst needs to worry about biases creeping into the estimates and inflating the variances. In addition, Illumina's method is better at accommodating asymmetric outliers than regular trimmed means. This is desirable, as these artefacts arise frequently in datasets we have analysed.

5.6.4 Normalisation

In this study we did not conduct a thorough investigation into normalisation methods. Although some degree of normalisation is always required, given the low variability of replicate observations for Illumina data it is important that the data are not "over-normalised", thus removing potentially interesting biological information. An important conclusion from the spike-in experiment is that the BGN recommended by Illumina is not appropriate for some analyses. This method is seen to introduce substantial variability into the data, particularly at low intensities, and also to increase the numbers of false positives. Another consequence of this normalisation is that low expression values become negative and cannot be $\log_2$ transformed. In the spike-in experiment, we found that around 40% of the data were missing on average per array. This is comparable to the situation described in early investigations into background correction and it is widely recognised that such missing values are problematic for analysis.

Illumina keep the bead-summary data on the unlogged scale and their model for differential expression takes the relationship between the mean and variance of each bead type into account (Chudin *et al.*, 2006). Differential expression analyses performed outside of BeadStudio usually require data that have been subjected to a $\log_2$, or similar, transformation to ensure the gene-wise variances are comparable. Therefore we recommend that only non-normalised data are exported from BeadStudio if they are to be analysed using established statistical methods. Otherwise, a small offset could be added to the intensities to ensure positivity of the background normalised data. However, optimal methods for deciding this offset require investigation. Another option would be to use *normexp* or other model-based approaches instead for this type of background adjustment.

5.6.5 Differential expression analysis

We find that the weighted $\log_2$ approach increases the evidence for differential expression for the spikes for each background correction method in most contrasts. At the same time, the log-odds scores for the non-spikes are not affected; this represents a gain in statistical power. The weighted $\log_2$ approach also produces more comparable log-odds between processing methods. Less precise observations arising from arrays with quality issues, or intensity-dependent trends in variablity introduced by the chosen pre-processing option, are down-weighted in the analysis. At very low concentrations (less than 1pM), this improvement was reversed, with differential expression statistics decreasing for the spikes. Although this would seem undesirable, it indicates that after considering the underlying variability of the observations, it is difficult to distinguish between very small changes in concentration, which is a limitation of any microarray technology. Having access to the bead-level data allows bead type variances to be calculated on the appropriate scale, and suitable outliers removed, so that they may be used in the linear model.

5.6.6 Annotation

We found the intensities of probes on an array to be related to base composition. In particular, probes with a higher GC content were seen to have a higher intensity, as were probes with a G or C at the first base. These effects were observed on normalised data from Strip 1 and persisted in the between-array comparisons. Inflated differential expression statistics were found for non-spikes with 17 to 21 GCs in their sequence. Crucially, we do not expect any of these bead types to show any differential expression.

A possible probe effect is also suggested by the intensity differences between spikes fixed to have the same concentration. Even the negative controls themselves show different intensity profiles across the experiment. Given these observations and previous work for Affymetrix arrays, it would seem that more sophisticated methods than BGN are needed to account for sequence-specific hybridisation effects.

Reannotation of the probe sequences provided by Illumina revealed that a large number of probes did not uniquely target their intended gene, or did not target the exonic region of the intended gene. Such problems are more prevalent on Strip 2, presumably because the transcripts targeted by this strip are less well understood and more challenging to design probes for. Nevertheless, many probes on Strip 1 were also found to map to intronic and intergenic regions, map to unreliable transcripts or have mismatches compared to the intended transcripts. The expression levels arising from probes with intronic and intergenic matches were generally low, although the exceptions where intergenic or intronic matches produce high expression levels require further investigation. Thus, in Chapter 6, I investigate the consequences of such annotation problems in more realistic experiments with a larger number of DE genes.

Finally, a fundamental design issue is raised by the inclusion of around 24,000 non-expressed probes on such whole-genome chips. Specifically, Illumina also offer the RefSeq content (roughly equivalent to Strip 1 of a Mouse6 or Human6 chip) as a separate product, with 8 samples interrogated on the same chip. Therefore we can obtain measurements for the reliable genes in a greater number of samples. If we were taking a naive approach of looking for the genes with the highest expression, then it is easy to imagine that the results will be comparable regardless of the choice of Human6 or Human8 chip. This issue will also be explored in Chapter 6 by looking at previously described experiments using the Human6 chip.

5.6.7 Application to other Illumina technologies

In this chapter, I describe the advantages of analysing a gene expression experiment using bead-level data. I anticipate that the analysis of other Illumina assays (e.g. GoldenGate, Infinium, DASL) can benefit from using bead-level data. For instance, recent genotyping methods for Affymetrix technology successfully use the full raw data and therefore having access to the bead-level data is likely to be useful in developing similar methods for Illumina. If log-ratios are required for genotyping, the situation is similar to expression data where the values output by BeadStudio are not on the desired scale. With the bead-level data, it is possible to obtain log-ratios for every bead and then calculate an average and variance for each bead type to be used in existing methods.

5.7 Validating a variance-stabilising transformation

In this section I describe how the spike-in experiment was used to validate an independently developed transformation method for Illumina data (Dunning *et al.*, 2008b). The data from microarray experiments generally require transformation in order to facilitate simple analyses such as the confident fitting of basic linear models. Variance-stabilising transformations are applied to microarray data in order to remove the mean-variance relationship in intensities. A $\log_2$ transformation is the simplest variance-stabilising transformation commonly applied to microarray data. Other more sophisticated variance-stabilising approaches have been developed, such as the VSN method (Huber *et al.*, 2002) and that of Durbin *et al.* (2002).

The VST method was introduced in Lin *et al.* (2008) as an adaptation of the VSN methodology for Illumina data, exploiting the abundance of replicate beads on each array. The authors show that VST outperforms $\log_2$

transformation, based on the results of Barnes *et al.* (2005). However, the authors commented on the (then) lack of a publicly available spike-in experiment, a dataset that would have provided an ideal test for their method.

The motivation for VST is given by assuming an error model that incorporates additive and multiplication errors, which give the variance of a measured intensity u, denoted by $v(u)$, as

$$v(u) = (c_1 u + c_2)^2 + c_3, \tag{5.1}$$

thus highlighting an undesirable property of microarrays that the variance of observations increases with the mean. The goal of variance-stabilising transformations in general is to define a function h that removes this dependancy. As shown in Lin *et al.* (2008), a suitable transformation is given by

$$h(y) = \frac{\text{arcsinh}(c_2/\sqrt{c_3} + c_1 y \sqrt{c_3})}{c_1} \tag{5.2}$$

for constants c_1, c_2 and c_3. Central to the VST method is the fact that the relatively large number of replicates available on Illumina arrays allow the estimation of v and u and therefore the constants c_1, c_2, c_3. The general procedure for VST first estimates c_3 by determining which bead types are not significantly expressed above background level (using the detection scores) and using the variance of these bead types as the estimate of the background noise, or c_3. Rearrangement of (5.1) then allows c_1 and c_2 to be estimated using a linear fit of the bead type means and variances (see Figure 5.11). The transformed values are then calculated according to (5.2).

In this section, we apply VST to data from the spike-in experiment. This offers further validation of the VST method, not only because the estimation of differential expression can be objectively assessed, but because the BeadArray technology used is different. The mixture data used in Barnes *et al.* (2005) is from a Human8 BeadChip with some 22,000 probes rather than the 48,000 used in the spike experiment. The Human8 chip is roughly equivalent to using only Strip 1 of a Human6 chip and we have already seen that the Mouse6 chip includes many low intensity probes that are only found on Strip 2. Therefore a similar effect should be expected for Human6 chips. Since the 48,000 probe Human6 is more widely used (see Chapter 2), it is important to confirm that VST can be applied to these higher density arrays

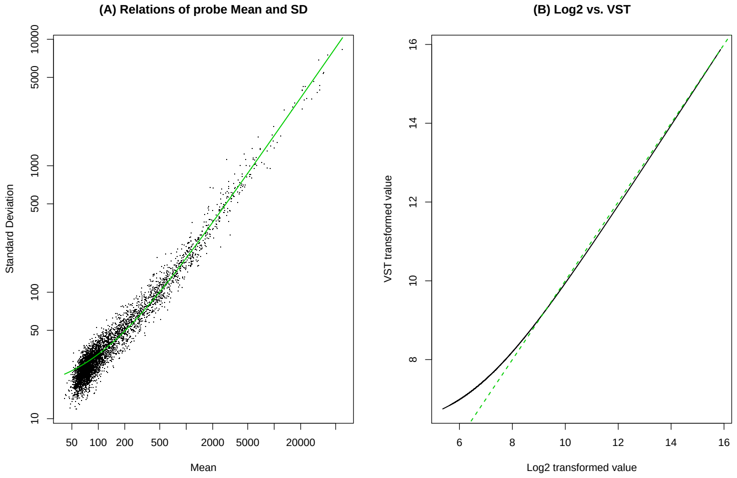


Figure 5.11: Demonstrating the VST transformation for an array in the spike-in experiment. A) The undesirable relationship between bead type means and standard deviations is shown. The linear fit shown in green is used to estimate the parameters c_1 and c_2. B) Comparison of VST and $\log_2$ transformed values for this array with the green line representing VST = $\log_2$. Figure created using the lumi package.

with no impairment due to the different distribution of intensities. Additionally, we will investigate whether VST can reduce the problem of missing values encountered when applying a standard $\log_2$ transformation after BGN. A key feature of VST is calculating the offset that must be added to each array to avoid negative intensities.

5.7.1 Methods

The bead-level data for the spike-in experiment were read by beadarray using the default background subtraction method. These bead intensities were then summarised using a 3 MAD cut-off to remove outliers. The data were summarised and transformed (VST or a $\log_2$) as appropriate, and the arrays were then quantile normalised. The bead-level data were also processed using both background subtraction and BGN after summarisation. The lumi software package (Du *et al.*, 2008) was then used to apply either a VST or a modified $\log_2$ transformation that avoids negative values.

The linear model described on Page 81 and subsequent analyses were used to find DE genes between arrays with different spike concentrations. We obtained log-odds scores quantifying the evidence for differential expression for both the spike and non-spike probes. The 12 spike concentrations allow for construction of 6 independent contrasts. We considered two sets: one where neighbouring concentrations are compared to provide the greatest challenge for differentiation (1000pM vs 300pM, 100pM vs 30pM etc.) and one where a range of effect sizes would be observed by contrasting pairs symmetric about the middle concentrations (1000pM vs 0pM, 300pM vs 0.01pM etc.). Finally, a series of smaller models were fitted, where only the 8 (of the 48) arrays featuring in the contrast of interest (4 arrays for each concentration) were considered.

5.7.2 Applying VST to the spike-in experiment

Figure 5.12 shows MA-plots comparing arrays with spikes at 3pM and 1pM. When BGN is not used, VST reduces the range of observed log-ratios for the probes we expect not to change. In the absence of BGN, both the $\log_2$ transformation and VST separate the spikes well from the non-spikes, but the log-fold changes achieved from the $\log_2$ transformation exhibit less bias.

Applying the transformations after BGN, we see that the MA-plot for VST is little changed. By contrast, the combination of BGN and $\log_2$ transformation is to be avoided, with much-reduced ability to separate out the spikes from the non-spikes by considering the $\log_2$ ratio, as we have previously noted.

Three linear models were fitted to the entire spike-in experiment: one using VST, one using a $\log_2$ transformation, and one using the weighted $\log_2$ analysis. For two of the linear models at a time, Figure 5.13 displays the differences in log-odds calculated for six contrasts. VST is seen to lead to a more powerful test than a standard $\log_2$ transformation, producing higher log-odds values for the spikes (Figure 5.13a/5.13c). At the same time, values for the non-spikes were not appreciably altered (data not shown). The difference between VST and $\log_2$ is seen to decrease as the spike concentrations get closer together (Figure 5.13c).

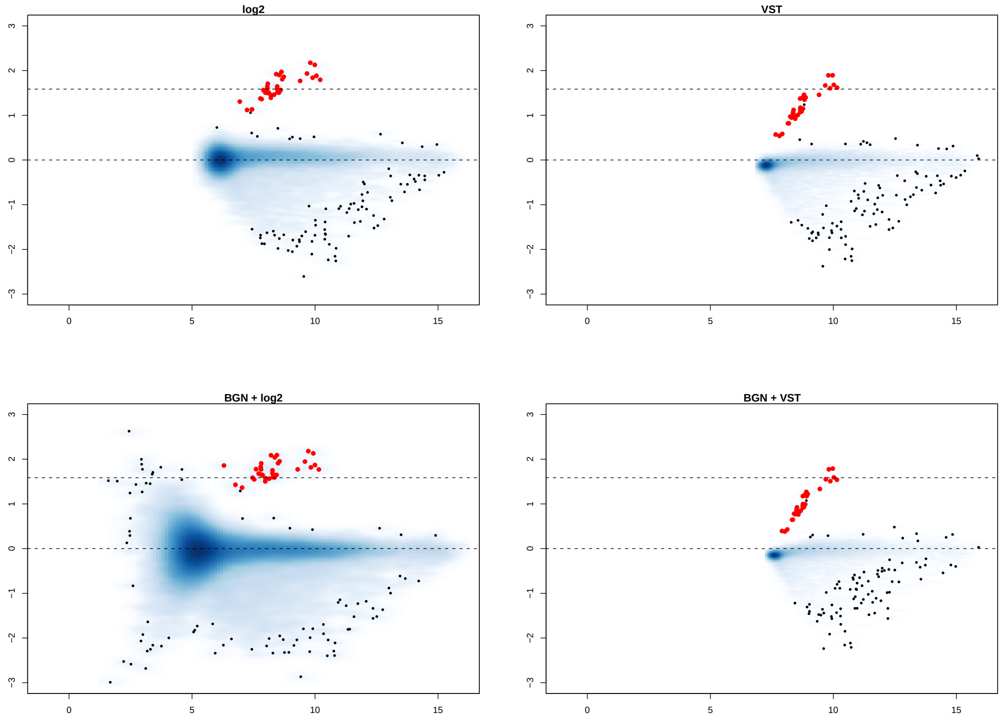


Figure 5.12: Here, we show the MA-plots for an array with spikes at concentration 3pM against spikes at concentration 1pM. In the top row, the arrays were transformed with a $\log_2$ transformation or VST. In the bottom row, the arrays were background normalised before transformation. In all plots, red dots mark the values for the spike probes and the dotted lines indicate the predicted log fold-change of spikes (1.73) and non-spikes (0) respectively.

When comparing VST to a weighted $\log_2$ analysis (Figure 5.13b/5.13d), VST is seen to be more powerful for detecting differential expression for large differences, but the weighted $\log_2$ analysis outperforms VST for finer comparisons (such as 100pM vs 30pM and 3pM vs 1pM).

When the models are fitted to only the arrays involved in the contrast of interest (Figure 5.14), the same broad trends are seen. The weighted $\log_2$ analysis, however, begins to show more sensitivity than VST even at quite extreme comparisons (e.g. 100pM vs 0.03pM).

5.7.3 Discussion

In agreement with Lin *et al.* (2008), we find that VST offers improvements over a standard $\log_2$ analysis. Thus, users with only the summarised output from BeadStudio will find this method beneficial. In particular, VST can cope with data that have been background normalised (BGN is implemented as the "subtract background" option in recent versions of BeadStudio).

Using a published spike-in experiment we are also able to show that VST offers greater ability to detect DE genes compared to a $\log_2$ transformation. This improvement was seen to diminish as the spike concentrations being compared become closer. At the same time, a weighted $\log_2$ analysis had more power than VST for finer concentration differences.

In our initial analysis of the spike-in experiment, we used all 48 arrays in the linear model. The size of such an experiment may not be typical for some researchers and therefore we repeated the analysis using fewer arrays. In this smaller experiment, VST was seen to have marginally improved log-odds over a regular $\log_2$ analysis. Under these conditions the weighted $\log_2$ analysis was seen to improve the detection of DE genes in most cases, especially when comparing arrays with similar spike concentrations. We note that a weighted $\log_2$ analysis is compromised without access to bead-level data. It would be beneficial if Illumina's software had the option to work with data on the $\log_2$ scale when creating summarised data.

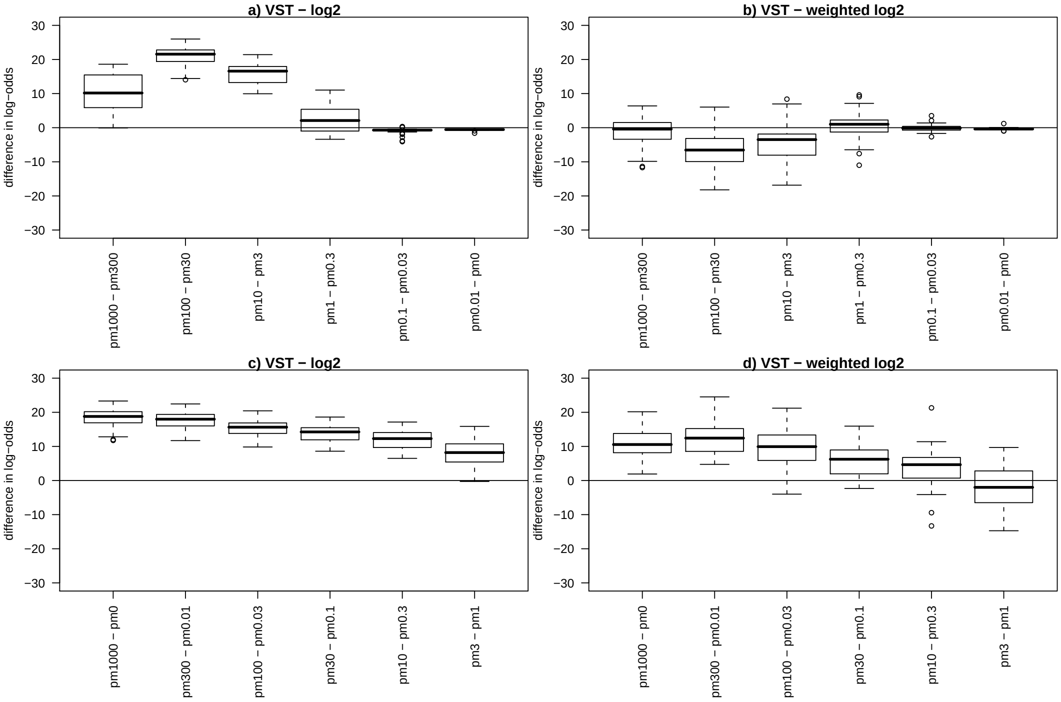


Figure 5.13: Comparison of spike log-odds obtained for a particular contrast in the linear model fitted to the entire spike-in experiment of 48 arrays. On the left we show the difference between the log-odds obtained after VST and the log-odds obtained after a $\log_2$ transformation. On the right, we show the difference between VST and a linear model incorporating $\log_2$ variances as weights. In the top panels, we show six independent contrasts with the closest spike concentrations. The bottom panel shows six independent contrasts from the same linear model, but chosen to provide a range in anticipated log-ratios (the finer differences being to the right of the panel). In all cases, a positive value indicates greater log-odds obtained (i.e., more evidence for differential expression) after VST.

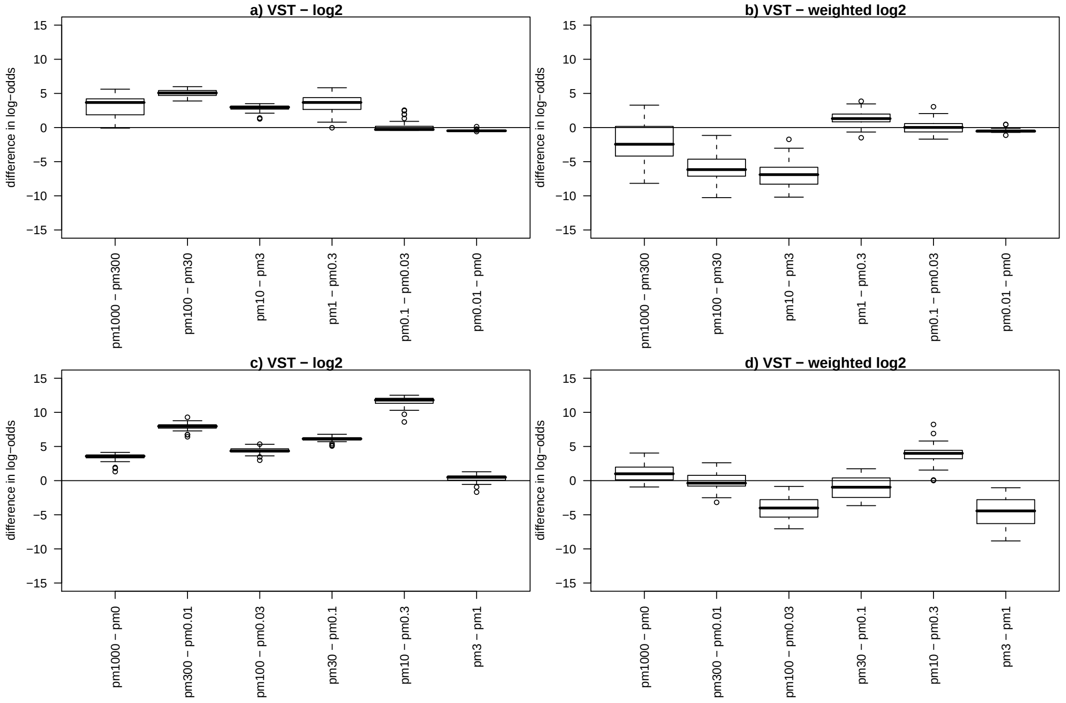


Figure 5.14: Comparison of spike log-odds obtained for a particular contrast in the linear model fitted to the 8 arrays involved in that contrast. On the left we show the difference between the log-odds obtained after VST and the log-odds obtained after a $\log_2$ transformation. On the right, we show the difference between VST and a linear model incorporating $\log_2$ variances as weights. In the top panels, we show six independent contrasts with the closest spike concentrations. The bottom panel shows six independent contrasts chosen to provide a range in anticipated log-ratios (the finer differences being to the right of the panel). In all cases, a positive value indicates greater log-odds obtained (i.e., more evidence for differential expression) after VST.

In summary, we have shown that the VST method does indeed perform well, and can be applied to the popular 48,000 probe BeadArrays. However, there are still benefits to having access to the raw data.

Chapter 6

Optimising the analysis of Illumina data by using prior knowledge

6.1 Introduction

In the previous chapter, various improvements to the analysis of Illumina data were described that take advantage of the availability of bead-level data. However, I recognise that many microarray facilities might not have the resources to analyse bead-level data. For a simple comparison of two samples types, where a swift conclusion is required, users may prefer to work with bead-summary data. Therefore it is important to understand how our experience of analysing the spike-in and other experiments can benefit these users.

There are many sources of error in a microarray experiment and different challenges are faced at each stage of the analysis. However, arguably the most fundamental issue is that of probe annotation. For without detailed knowledge of where the sequences designed for the array map to, we cannot hope to gain biologically meaningful conclusions. Several efforts to reannotate Affymetrix probe sequences have been presented (Harbig *et al.*, 2005; Dai *et al.*, 2005; Gautier *et al.*, 2004b) and it is generally believed that such redefinitions drastically improve the reliability of a differential expression analysis study (Harbig *et al.*, 2005; Dai *et al.*, 2005; Gautier *et al.*, 2004b; Sandberg and Larsson, 2007). Given the unique design of Affymetrix arrays, one has to combine the different intensities measured for the probes within the probe set for each gene. Clearly the summarised value for a particular gene could be severely altered if not all the probes in the probe set map to the correct location, or map to multiple locations. Thus, reannotation of Affymetrix probes centres around reorganising the existing probe sets to more accurately reflect the targets of the individual probe sequences. Alternative annotations based around genes, exons or transcripts (Dai *et al.*, 2005) can also be constructed. To my knowledge, similar reannotations have not been done for other technologies.

Reannotation of the probe sequences used in the Illumina spike-in experiment revealed many probes matching to intronic or intergenic regions. The spike-in experiment was not the ideal scenario to judge if probe annotation can affect the results of a differential expression analysis, as all non-spikes were essentially constant throughout the experiment and all spike probes had reliable annotation. In a more realistic experiment, no matter what combination of background correction and normalisation methods are applied, we would not expect to recover the true expression level of a gene whose probe matches to an intronic or intergenic region, as such probes may not accurately measure the transcription of the gene. Whilst measurements made in intergenic and intronic regions could be potentially interesting, the Human6 chip is intended to be a gene expression platform and for studies interested in measuring expression at all points along the genome, or alternative splicing, there are much more suitable platforms such as tiling or exon arrays. The reannotation of Illumina data poses a different problem from Affymetrix, as the replicated observations for the same bead type all have the same probe sequence attached. For an Affymetrix probe set, if one probe is defective then there are still multiple probes that can potentially be used to interrogate the gene. However, if an Illumina probe is defective, then all measurements for that bead type are compromised. Moreover, each gene is usually represented by only one bead type.

In this chapter, I focus on the MAQC dataset, which was specifically designed to have many DE genes, and other publicly available datasets that use the Human6 chip. A typical microarray analysis will often include a filtering step to remove probes that are uninformative. Such filtering techniques are applied after considering expression level and variability, and we have already seen in Chapter 5 that probes with intronic or intergenic matches tend to have lower signal. Therefore, I investigate whether some commonly applied filtering techniques eliminate these misleading probes from the analysis. I also show the effect that filtering has on the number of significant findings arising from a differential expression analysis, by looking at the MAQC dataset and other Human6 experiments as examples.

6.2 Data and Methods

6.2.1 The MAQC dataset

This chapter presents an analysis of the Illumina portion of the MAQC project. As described previously, the MAQC dataset consists of four samples (A, B, C and D), each sample being a mixture of Universal Human Reference RNA (UHRR) and Universal Human Brain Reference RNA (UHBRR). The mixtures were defined to be 100% UHRR and 0% UBRR (A), 0% UHRR and 100% UHBRR (B) , 75% UHRR and 25% UHBRR (C) and 25% UHRR and 75% UHBRR (D). Each sample was replicated five times and the entire experiment repeated independently in three different locations, giving a total of 60 arrays. However, only 59 of the Illumina samples passed the QC standards set by the MAQC. The analysis presented in the original MAQC paper was aimed at comparisons between microarray platforms and quantifying the agreement between the various platforms. For between-platform comparisons, the probes on each platform had to be filtered to obtain a list of common genes interrogated by all platforms. Thus, there was no investigation involving the entire set of probes for each platform. For this chapter, non-normalised bead-summary data were downloaded for the Illumina arrays, which included the summarised expression values for each bead type, standard errors, detection scores and number of beads. All 48,000 probes were included in the analysis, although no control information was available.

6.2.2 The GEO dataset

Datasets from Human6 Version 1 chips (GPL2507) performed in a wide variety of experimental conditions and tissue types (e.g. breast, blood, artery, stem cell, sperm) were taken from the Gene Expression Omnibus (GEO)

(Barrett *et al.*, 2007). These datasets have been previously summarised in Table 2.1. Although this is not the most up-to-date version of the Illumina chip available, it provides the greatest amount of publicly available data. The GEOquery Bioconductor package (Davis, 2008) was extremely useful in reading the GEO data into R for further analysis, as each dataset is converted into an *ExpressionSet* structure, including detailed metadata recording the samples hybridised to the arrays and processing methods used. The *ExpressionSet* representations of each GEO entry were collated into a list object, and this is referred to as the *GEO dataset*. The expression values for each GEO entry were also joined together into a large matrix with 684 columns and used to illustrate general properties of probes across a large collection of arrays from various sources.

6.2.3 Filtering Methods Used

A series of filtering approaches were applied to the entire MAQC dataset of all 59 arrays from the three locations.

- Detection - The function `detectionCall` in lumi was used to determine how many of the 59 arrays each gene was detected on. This uses the detection p-values provided by Illumina and a fixed p-value threshold, with values below this threshold deemed to be detected. Genes detected on at least one array passed the filter.

- Expression Level - The `kOverA` function from genefilter was applied to the non-normalised expression values to determine which genes had expression level higher than "A" on at least "k" of the 59 arrays. The value of k was set to 10 for varying values of A.

- IQR - The IQR of each probe was calculated from $\log_2$ normalised data and genes greater than a given cut-off passed the filter.

A range of different cut-offs were used for each of the methods, and the number of probes in each of the annotation categories (see page 82) retained by the filter were recorded, along with the number of Strip 1 and Strip 2 probes.

As the data in the GEO dataset were generated from a diverse set of tissue types and processed using different normalisation methods, direct comparisons between all arrays are not possible. Therefore, all intensities on each array were ranked separately to see how the relative expression level of particular probes changes on different arrays. The average rank for each bead type was then calculated and used to assess differences between the different annotation categories.

6.2.4 Use of annotation information

The Human6 Version 1 chip was reannotated according to Barbosa-Morais *et al.* (2008), resulting in a tab-delimited file. The file has one row for each bead type and a number of columns that can be used to judge the reliability of the probe sequence. Full details of the file contents are given in Barbosa-Morais *et al.* (2008).

This file can be easily read into `R` and incorporated into an analysis. For instance, one can use `grep` or `match` on the Target column to find the rows in the file that correspond to a given set of IDs. Similarly, the probes that have a particular property (such as having an intronic match) can be returned.

Online resources can also be used to retrieve information about the probe sequences used on Illumina arrays. In this chapter, I used the UCSC browser (Kuhn *et al.*, 2007) to manually align particular sequences to the latest version of the human genome.

6.2.5 Differential expression analysis

I performed a simple differential expression analysis for the MAQC dataset using limma. A gene-wise linear model was fitted to quantile normalised data from each location separately to estimate coefficients for each of the samples (A, B, C and D) for all 48,000 probes. All pairwise contrasts were then formed (six in total) and empirical Bayes shrinkage used to produce differential expression statistics. The same linear model was fitted to different subsets of the data.

	Strip 1	Strip 2
Total	26097	21198
Intronic	819	5890
Intergenic	838	8055
Unreliable	1256	3719
Mismatch	1663	629
No Match	444	964
Multiple Match	1024	470
Percentage of Good Probes	78%	9%

Table 6.1: Results of reannotation of the Illumina Human6 Version 1 platform. The 26,091 probes on Strip 1 and 21,198 probes on Strip 2 are divided into categories describing various annotation problems. “Good” denotes probes that had a complete genomic match to the exonic region of a known transcript.

- Fit 1 - Linear model fitted to all 48,000 probes simultaneously.
- Fit 2 - Linear model fitted only to probes that can be found on Strip 1. Around 26,000 probes were used in this analysis.
- Fit 3 - Linear model fitted only to probes with “Good” annotation. Around 20,000 probes were used in this analysis.

The `decideTests` function in limma was then used to find the number of significant findings for each contrast after multiple testing correction using Benjamini-Hochberg correction (Benjamini and Hochberg, 1995) and a false discovery rate (FDR) set at 0.05.

6.3 Results

6.3.1 General Observations

The reannotation of the Human6 Version 1 platform is shown in Table 6.1. A striking conclusion is the difference in overall reliability between the two strips. As for the spike-in experiment, we see far fewer probes with reliable annotation on Strip 2 than Strip 1. With the exception of mismatch probes, all other undesirable properties (intronic, intergenic, unreliable and no match) were more prevalent on Strip 2. The number of probes in each category are roughly the same as for the Mouse6 chip (see Page 94). However, the number of intergenic probes is far greater for the Human6 chip, especially on Strip 2. At the same time, the Human6 chip has far fewer mismatch probes compared to the Mouse6 chip.

Figure 6.1A shows the non-normalised Illumina MAQC data from one location after a $\log_2$ transformation. Note that one of the replicates of Sample C was removed during quality control by the MAQC, otherwise all samples are replicated five times. As these data were exported from BeadStudio, all 48,000 observations for each array are given by default. However, based on previous experience in analysing the Mouse6 spike-in experiment, we have good reason to suspect an intensity difference between the two strips on the Human6 chips. Therefore, the data for each array were also split up into Strip 1 and Strip 2 probes (by knowing which probes were located on each strip) to produce Figure 6.1B. Consecutive pairs of boxes represent the Strip 1 and Strip 2 probes for the same array. The difference between the two strips is obvious and seen for all arrays. For every array, the Strip 1 probes have higher median and higher IQR than Strip 2. For most arrays, the 75th percentile on Strip 2 is around the same as the 25th percentile for probes on Strip 1. This difference would not have been immediately obvious from the bead-summary output provided online.

6.3.2 Filtering

Figure 6.2 shows the number of probes retained under three different filtering methods for the MAQC dataset. A variety of different cut-offs were used for each method. For filtering based on expression level (labelled Detection and Expression), we see that Strip 1 probes are more readily retained than Strip 2. Requiring a detection p-value of at least 0.1 on any array retains around 91% of the Strip 1 probes, compared to around 63% of Strip 2 probes. As we make the detection cut-off more stringent fewer probes are retained, although more Strip 1 probes are retained than Strip 2 at each cut-off. This is in agreement with previous observations that Strip 2 probes generally have lower expression levels. With a cut-off of 0.05, commonly used in the literature, 86% of Strip 1 probes pass the filter compared to 49% of Strip 2 probes. In terms of the annotation categories, intronic and intergenic probes are also

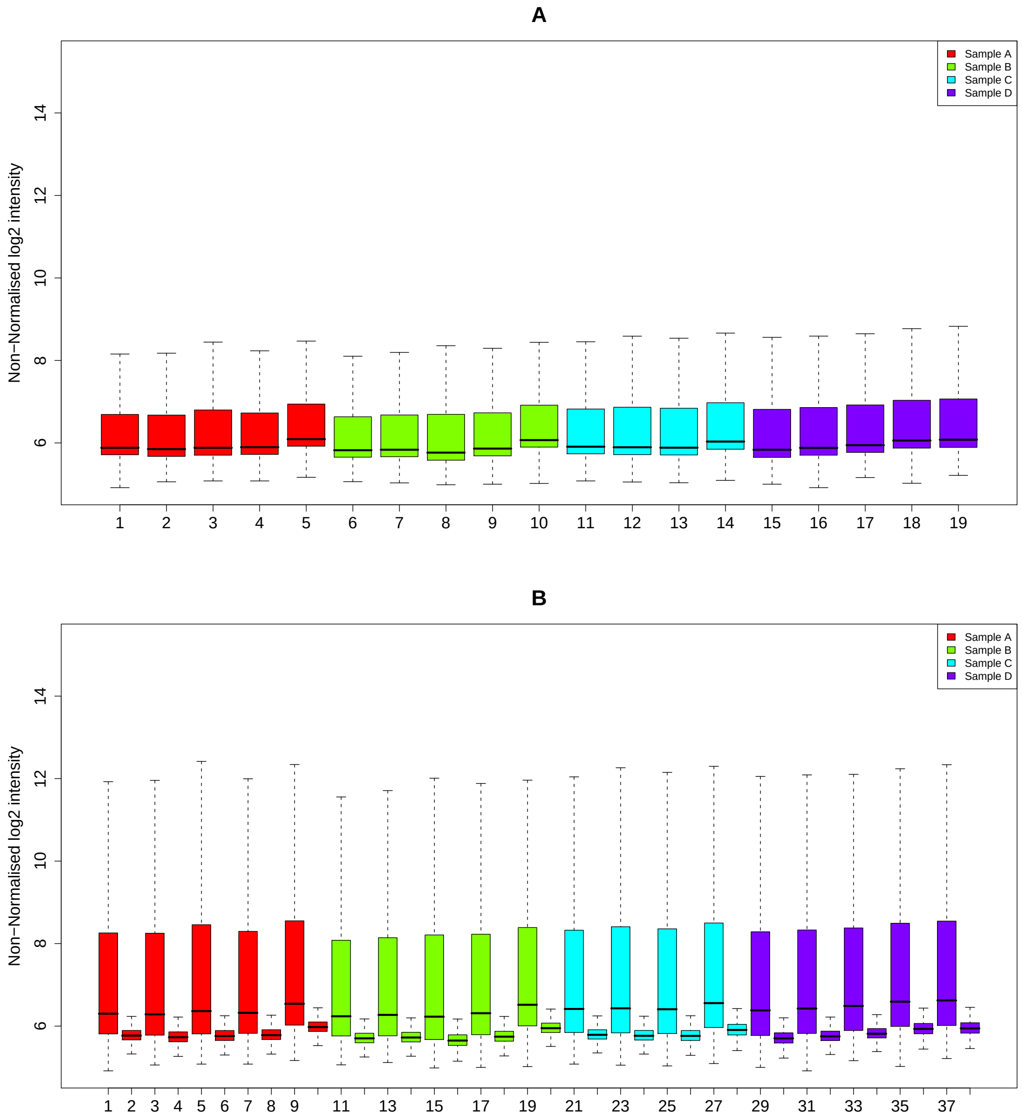


Figure 6.1: Two different views of non-normalised Illumina data from the MAQC study. In A, we show data from one location. One array was removed during quality assessment by the MAQC so in total there are 19 arrays with 48,000 observations each. In B, the same arrays are divided into the probes found on Strip 1 and Strip 2. Similar boxplots were seen for all three locations.

more often removed at a given cut-off, although this is confounded by the fact that most of these probes are located on Strip 2. On the other hand, probes with mismatches are less often removed by filtering, suggesting that hybridisation is possible even if the probe and target sequences do not match perfectly.

By filtering based on minimum expression level across all arrays, we are more likely to retain Strip 1 probes and remove Strip 2 probes. If the cut-off is suitably low (e.g. 60) then nearly all probes are retained. However, choosing a cut-off of 120 retains 66% of Strip 1 probes and only 19% of Strip 2 probes. Filtering based on IQR gives broadly similar results, with an IQR of around 0.2 being the point at which dramatic differences are seen in the numbers of Strip 1 and Strip 2 probes removed. A cut-off of 0.5 (suggested by other authors (Scholtens and Heydebreck, 2005)) will retain 35% of Strip 1 probes, but less than 10% of Strip 2 probes.

We conclude that simple filtering schemes with arbitrary cut-offs tend to remove Strip 2 probes and probes in intronic and intergenic regions more often than probes on Strip 1. Not only do probes on Strip 2 have lower expression on a given array, their expression levels also vary little between arrays. Ideally a filter would remove a reasonable percentage of uniformative probes, without removing too many useful probes. However, deriving a suitable cut-off is not trivial. Since some analyses of Human6 chips described in Chapter 2 used a filtering step akin to those used for Figure 6.2, we might reasonably question if the majority of the analyses in the literature are reporting results that came from Strip 1, and if Strip 2 added any useful information to the analysis.

For instance, in MAQC Consortium (2006) the first stage of the analysis was to find probes that were common to all platforms being studied. A filtered list of 12,091 probes was derived using the RefSeq database, so it is not surprising that 99% of the probes used in the analysis were found on the Human8 chip. In other words, the analysis did not use the vast majority of the additional content added to the Human6 chip, and the same analysis conclusions could have been reached using Human8 chips. Similar experiments that use Illumina technology as a validation of other results, or to compare with other platforms, also had to filter to a set of genes that would be almost

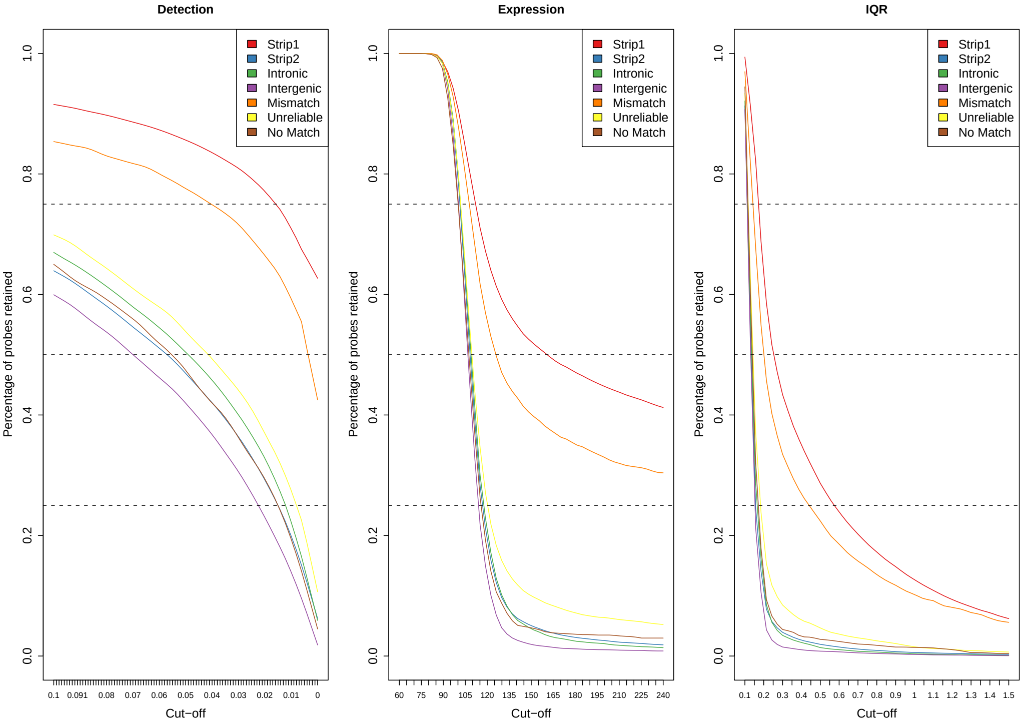


Figure 6.2: For different filtering methods (see Section 6.2.3) we show the percentage of genes belonging to Strip 1 and Strip 2 and different annotation categories retained by the filter.

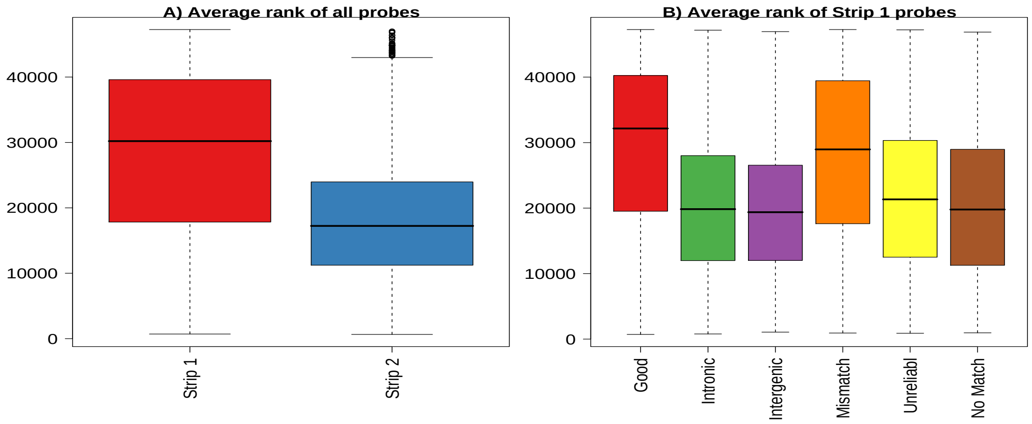


Figure 6.3: Average ranks of probes across all Illumina Human6 arrays in GEO. A) shows the average ranks of probes on Strip 1 versus probes on Strip 2. Probes on Strip 1 (roughly equivalent to the lower-density Human8 arrays) are found to be ranked higher than those on Strip 2. In B) the probes on Strip 1 are split into different annotation categories.

entirely probes found on the Human8 chip (Ramilo *et al.*, 2007; Lenk *et al.*, 2007).

As the arrays in the GEO dataset were analysed differently and sometimes on different scales, applying the same filtering methods as above is not possible. Instead, the probes on the same array were ranked and the average rank for each probe over all arrays was computed. Figure 6.3A shows the ranks in terms of the probes on the Human6 chip that are present on Strip 1 or Strip 2. A clear preference is seen for Strip 1 probes to be found at the top of the list.

Figure 6.3B shows how the ranks of genes vary between different categories of probes on Strip 1 (in order to account for the perceived difference in reliability between Strip 1 and Strip 2). The result is similar to Figure 5.8. As we would hope, probes with good annotation are seen to have a high rank (around 30,000 on average), whereas probes with annotation problems (e.g. matching to an intronic or intergenic region, or with no complete matches), with the exception of mismatch probes, have a much lower rank of around 20,000. We would expect such regions of the genome to not produce any signal in a gene expression study. However, there are a few examples of probes that fall into this category and are also highly ranked on all arrays.

A total of 132 bead types were found to have an average rank of higher than 46,800 (roughly equivalent to the 99th percentile of the ranks). Of these bead types, 36 (27%) were found to target ribosomal proteins, which function in protein biosynthesis and some of which have been previously found to be housekeeping or reference genes (de Jonge *et al.*, 2007; Thorrez *et al.*, 2008). Thus, these genes are often considered useful for normalisation. Indeed, 5 genes from this crudely derived list were also found in the table of top 15 candidate housekeeping genes presented in de Jonge *et al.* (2007), which included 13 ribosomal proteins. Another common feature of these 132 probes was that 32 had multiple matches to other regions of the genome.

Surprisingly, a handful of these 132 bead types with consistently high rank had intergenic (4), intronic (5) or no matches (2). Our reannotation of the probe sequences for these misannotated probes did not immediately explain the high rank. After a manual BLAT search using the UCSC genome browser, one intergenic probe and one intronic probe were found to map to regions with SINE repeated DNA. Such regions of DNA are found throughout the genome and hence the search returned many matches. The two bead types that were reannotated as having "no match" did, in fact, map to multiple locations, although the sequence mapped to splice junctions. These two probes with no matches were intended to target transcripts related to ribosomal proteins.

In Golubkov *et al.* (2006), data were processed using BeadStudio, although analysed through Bioconductor and GeneSpring. The first step of the analysis was to remove unexpressed probes using the detection p-values. For a probe to pass the filter (i.e. be included in further analysis) it had to be significant at the 0.1 level in at least one of the arrays (confirmed by personal communication with GEO submitter). After this filtering step, 21,892 probes out of the original 47,293 remained. Unfortunately, as detection scores were not available for this dataset, we were not able to reproduce this filtering. As an approximation, I ranked the intensities for each array and used the 21,892 probes with the highest average ranks for further analysis. Around 80% of our filtered list appeared on the Human8 chip. Also, around 74% of the 21,892 probes had good annotation. Thus, many probes with poor annotation, including intronic and intergenic matches were still present in the analysis despite filtering.

In Greber *et al.* (2007), no mention is made of filtering prior to testing for DE genes. However, the detection scores for each array are provided as supplementary material. We were therefore able to ascertain that if filtering had been used at the 0.05 level, 72% of the resulting list would have been probes present on the Human8 chip. This same list is also dominated by good probes (68%) but also with 20% intronic or intergenic probes and around 10% mismatch or unreliable probes.

One of the few papers to refer to the different annotation sources for the Human6 chip was Bykhovskaya *et al.* (2007). The analysis presented in that paper used a detection p-value filter of 0.05, which resulted in a list of 12,983 detected probes. Although no detection scores were provided we are able to estimate that 86% of this list is from the Human8 chip, with 81% of filtered probes having good annotation.

In summary, the intensity difference between Strip 1 and Strip 2 seems to be universal for Human6 Version 1 arrays, and by performing filtering we restrict the analysis to mostly Strip 1 probes. Also, from looking at probe behaviour over the large set of arrays in the GEO dataset, we saw that high probe signal is observed for many probes that we previously believed to have poor annotation. Certain probes also have consistently high signal, regardless of the sample type. Many analyses in the literature include misannotated probes that pass the first round of filtering. However, investigation of these anomalies revealed other causes of signal, such as the presence of repeated DNA, and matching multiple genomic locations, rather than just intended gene transcript. In the following results, it will be shown that such bead types can be reported in the results of a differential expression analysis if due care is not taken.

6.3.3 Differential expression

The MAQC dataset

For each pairwise contrast in the differential expression analysis described on Page 116, I ranked all 48,000 probes according to the evidence for differential expression (as measured by the log-odds score). Thus, the probe ranked 1 is the most likely to be DE and so on. Note that we are not trying to impose any cut-offs to decide which probes are truly DE at this stage.

Figure 6.4 represents how far down the ranked list of genes we have to go before finding a particular percentage of Strip 1 or Strip 2 probes. Separate curves are shown for each of the pairwise contrasts. For example, for each contrast in Fit 1, after looking at the 10,000 highest ranked probes, we have encountered around 35% of the Strip 1 probes and less than 5% of the Strip 2 probes. A common strategy for selecting interesting genes for further validation is to select the N genes with the strongest evidence of differential expression. Therefore, applying this strategy to the results of Fit 1, we would be more likely to pick probes from Strip 1, whereas Strip 2 probes very rarely appear amongst the top-ranked probes for a given contrast. For practical and financial reasons, the number of genes chosen for validation is restricted. In such a scenario, Strip 2 probes would be chosen rarely. On average, only 5 out of the top 250 ranked probes for each contrast were found to be from Strip 2. However, it should be noted that there are likely to be many thousands of DE genes in this analysis due to the samples chosen, with many genes found to be uniquely expressed in the brain.

Rather than relying on ad-hoc criteria to filter the data based on expression level or variability, Fit 2 used the same arrays, but only the probes from Strip 1. A different set of moderated-t statistics is obtained due to the variance smoothing using a different set of gene-specific standard errors. The prior degrees of freedom output from the model indicates the amount of information that has been "borrowed" in order to smooth the variances. The values of this parameter were 11.00 and 7.32 for Fit 1 and Fit 2 respectively, thus indicating a greater degree of smoothing performed in the first model.

Figure 6.4 shows the composition of the ranked gene lists from Fit 2 in terms of the annotation categories. As we expect, bead types with good an-

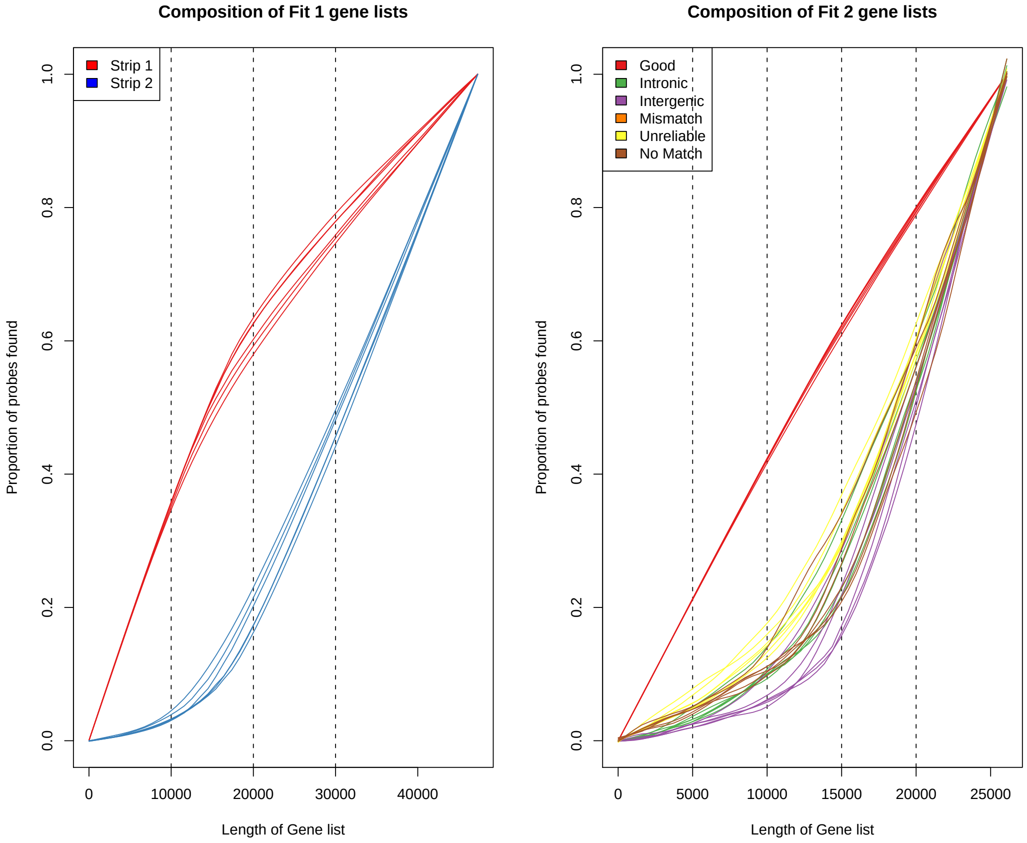


Figure 6.4: Representing the composition of ranked lists of genes obtained from the MAQC dataset after a differential expression analysis. Separate curves are shown for each pairwise contrast of samples. For the linear model fitted to all probes (Fit 1), we show the proportion of Strip 1 or Strip 2 probes encountered along the length of the gene list. For the linear model fitted to probes from Strip 1 only (Fit 2), we show the composition of the gene list in terms of the annotation categories.

notation occur towards the top of the list. On the other hand, genes mapping to intronic and intergenic regions, or having no match at all, are found to be ranked low in the list. Probes with mismatches still appear high in the list, although such mismatches should be treated with caution.

It is possible to find some examples of intergenic or intronic probes that appear to be DE in this analysis. For instance, the bead type labelled “GI_8922831-S” is supposed to target a gene with the symbol “FLJ11029”. This bead type has high evidence for being DE (log-odds scores > 35 in 5 of the 6 contrasts) and yet the reannotation for the bead type reveals that it maps to the intronic region of the gene. Manual BLAT search with the UCSC genome browser reveals that the primary match for the gene lies in a region of SINE repetitive DNA sequence and therefore has many matches throughout the genome. We should not conclude that the apparent differential expression for this bead type is due to biological variation.

Another interesting example is bead type “GI_5174761-S”, found on Strip 1, which is annotated by Illumina as targeting the gene MT3, and described as “Homo sapiens metallothionein 3 (growth inhibitory factor (neurotrophic))”. This bead type is ranked within the top 25 genes in contrasts A vs B, A vs C and A vs D in the analysis of data from two locations. Our reannotation flags the probe for this bead type as having no match in the genome, and manual BLAT of the probe sequence reveals that it matches a 53 base sequence in the MT3 gene. However, the normalised expression values shows that this probe is highly expressed in samples B, C and D (which contain some UHBRR sample), but not detected above background in Sample A. Moreover, checking the GeneCard for MT3 (www.genecards.org), we find it is commonly expressed in the brain, as the description of the bead type implies. Therefore, it could be that the signal measured by this probe is plausible and capturing a difference between biological samples, despite the gaps in the probe sequence.

Figure 6.5 shows the number of significant Strip 1 probes from Fit 1, compared to the number of significant probes in Fit 2. Bear in mind that Fit 1 was fitted to 47,289 probes compared to 26,091 probes in Fit 2. Despite the reduction in the number of probes used, Fit 2 achieves a great number of significant genes for every contrast. Each increase corresponds to genes that were not deemed to be significant in the original model.

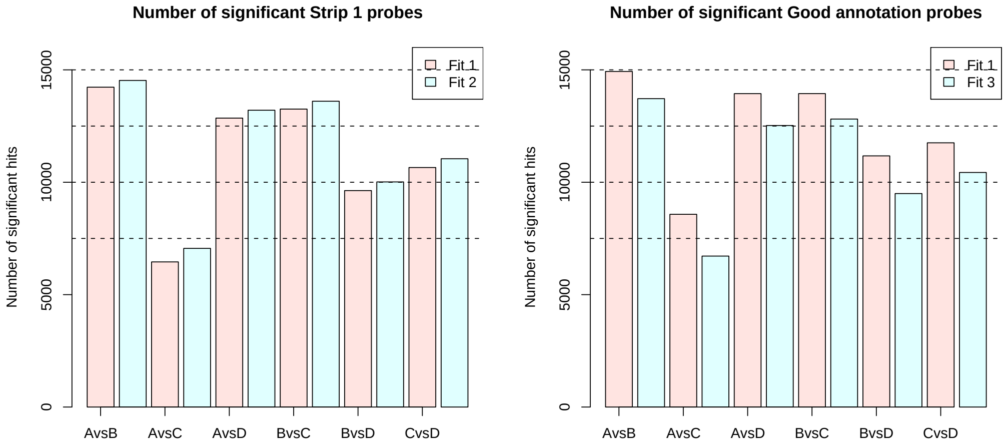


Figure 6.5: Improvements to the detection of DE genes by filtering. On the left, the number of significant findings (after multiple testing correction) for Strip 1 probes using Fit 1 and Fit 2 are shown. For all the contrasts, the number of significant probes is greater using Fit 2. On the right, the number of probes with good annotation is shown under Fit 1 and Fit 2. Fit 3 is also seen to increase the number of significant findings.

Removing all the Strip 2 probes from the analysis might seem a bit heavy-handed as we are ignoring any potentially useful probes on Strip 2. Therefore, Fit 3 was used to find DE genes among those with reliable annotation, which includes both Strip 1 and Strip 2 probes. For each of the six contrasts, a greater number of significant genes are found by fitting the model to just the probes with good annotation. This suggests that if we restrict our analysis to genes with reliable annotation we can achieve greater power to detect DE genes. By using only reliably annotated genes, we would also be more likely to obtain further biological information for the genes in the analysis, such as the names of pathways or gene ontologies.

The GEO dataset

A Welch t-test was performed in Golubkov *et al.* (2006) to determine differential expression in the contrast of interest with an additional requirement of 2-fold change between the two groups. This resulted in the 207 genes given in Supplementary Table 1 of Golubkov *et al.* (2006). Our attempt to repeat this analysis gave 205 genes, which shows that our method of filtering was comparable to the original approach. Of the 207 hits reported in Golubkov *et al.* (2006), only 7 findings were not present on the Human8 chip (Hs.28792-S, GI_27498491-S, Hs.196008-S, GI_30156248-S, Hs.445581-S, GI_27485722-S, Hs.370806-S). The four probes with prefix "Hs" are probes absent from the RefSeq database, whereas the other three are predicted genes ("XM"). If the authors had wished to use an analysis of GO terms, then these four "Hs" probes would not contribute to the analysis as tools such as GOstats (Falcon and Gentleman, 2007) generally require a RefSeq, or similar, identifier for each probe used in the analysis.

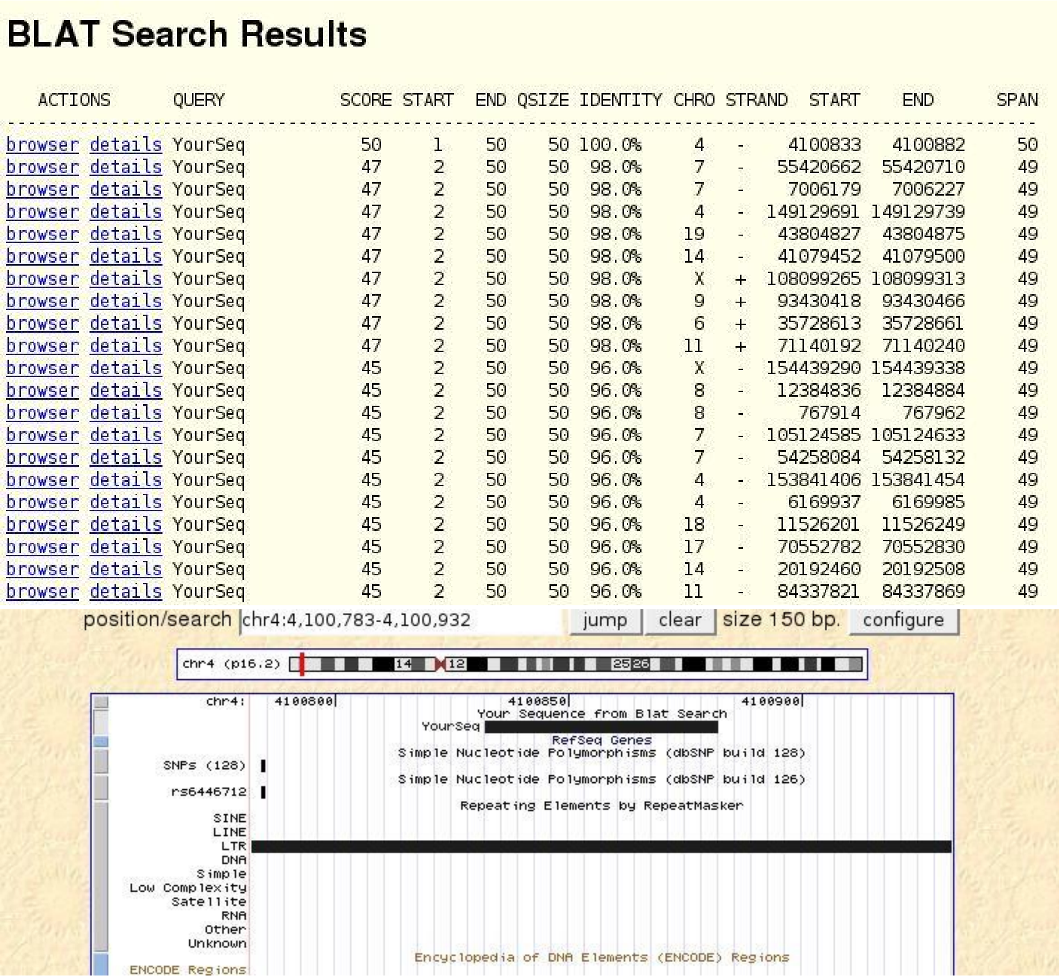
When ranking the 207 significant genes based on $\log_2$ fold-changes, five of the new findings from the Human8 chip showed $\log_2$ fold-changes around 0.43 to 0.48 and ranked lowly in the list (186 -195). The highest ranking for these 7 new findings is Hs.28792-S with a $\log_2$ fold-change of 5.16 between the two conditions. However, our reannotation shows this probe to have an unreliable transcript mapping (using the Comments column). It would be at the discretion of the researcher whether to follow up this finding or not.

Overall, the list of 207 reported genes comprised "good" probes, with the exception of 17 mismatch probes, 3 unreliable probes and 1 intergenic probe. The 3rd ranked gene in the list (GI_40316911-S) was found to be a mismatch probe based on the "Comments" column and was found to have an additional identical match on the same chromosome. The inclusion of an intergenic probe (GI_37549959) in the list of 207 DE genes is intriguing. This probe is intended to target a gene with symbol "LOC375459" on chromosome 5 with a description provided by Illumina of "Homo sapiens similar to Proteasome activator complex subunit 2 (Proteasome activator 28-beta subunit) (PA28beta) (PA28b) (Activator of multicatalytic protease subunit 2) (11S regulator complex beta subunit) (REG-beta)". After performing a manual BLAT search against the genome, we verify that this probe indeed targets an intergenic region. Additionally, it also shows a match of 47 bases to a sequence on chromosome 14 lying in the transcribed region of the gene PSME2 which is described as "Homo sapiens cDNA: FLJ22927 fis, clone KAT07022, highly similar to HUMPHPA28A Human mRNA for proteasome activator hPA28 subunit beta". Strikingly, the correlation of bead types GI_37549959 and GI_30410791 (targetting PSME2) in this dataset is 0.98. Furthermore, if we take a larger, unrelated, dataset such as Stranger *et al.* (2007) (with 480 arrays), we find a correlation between the two bead types to be 0.94. The two bead types seem to be related in function, and we might interpret the results of Golubkov *et al.* (2006) as the rarer transcript represented by GI_37549959 showing significant biological differences. However, this bead type is reporting the same transcript as GI_37549959, with the added uncertainty caused by the multiple matches of the probe sequence.

Six gene lists are provided as supplementary material to Greber *et al.* (2007), consisting of over- and under-expressed genes when each of three transcription factors were knocked down. For each transcription factor, the number of reported DE genes from the Human8 chip is 94% (Oct4), 93% (Nanog) and 95% (Sox2) and 515 probes were found to appear in each of the three lists of DE genes. Of these 515 probes, 5 were found to be intronic, 5 intergenic, 40 mismatches and 15 unreliable. The reannotation of the 5 intergenic probes did not give any further information on why these probes should be called DE. However, the manual BLAT of probe GI_42657060-S (shown in Figure 6.6) revealed a large number of partial matches of the probe sequence. Further investigation of the primary match for the probe shows that the probe does not lie within any RefSeq genes, but is found within an LTR region found many times throughout the genome. This explains the large number of partial matches.

Using the vast amount of Human6 data obtained from GEO, we can look at the performance of GI_42657060-S on a typical array. Figure 6.7 shows the rank of this probe on all arrays in the GEO dataset. There is a clear tendency for this probe to be among the highest intensity probes on a given array, in a similar way to the 132 highly ranked probes found previously. Therefore this probe would almost certainly be retained after filtering using expression or detection levels. However, due to the saturation effect seen (see Chapter 5), there is much uncertainty in measurements of genes at high intensity. We should therefore be cautious about declaring such probes to be DE. I will now give an example where reannotation can drastically change some of the conclusions of an experiment.

In Bykhovskaya *et al.* (2007), data were normalised using average nor-



BLAT Search Results

```
   ACTIONS      QUERY           SCORE START  END QSIZE IDENTITY CHRO STRAND  START     END      SPAN
---------------------------------------------------------------------------------------------------
browser details YourSeq           50     1    50    50 100.0%     4   -     4100833   4100882    50
browser details YourSeq           47     2    50    50  98.0%     7   -    55420662  55420710    49
browser details YourSeq           47     2    50    50  98.0%     7   -     7006179   7006227    49
browser details YourSeq           47     2    50    50  98.0%     4   -   149129691 149129739    49
browser details YourSeq           47     2    50    50  98.0%    19   -    43804827  43804875    49
browser details YourSeq           47     2    50    50  98.0%    14   -    41079452  41079500    49
browser details YourSeq           47     2    50    50  98.0%     X   +   108099265 108099313    49
browser details YourSeq           47     2    50    50  98.0%     9   +    93430418  93430466    49
browser details YourSeq           47     2    50    50  98.0%     6   +    35728613  35728661    49
browser details YourSeq           47     2    50    50  98.0%    11   +    71140192  71140240    49
browser details YourSeq           45     2    50    50  96.0%     X   -   154439290 154439338    49
browser details YourSeq           45     2    50    50  96.0%     8   -    12384836  12384884    49
browser details YourSeq           45     2    50    50  96.0%     8   -      767914    767962    49
browser details YourSeq           45     2    50    50  96.0%     7   -   105124585 105124633    49
browser details YourSeq           45     2    50    50  96.0%     7   -    54258084  54258132    49
browser details YourSeq           45     2    50    50  96.0%     4   -   153841406 153841454    49
browser details YourSeq           45     2    50    50  96.0%     4   -     6169937   6169985    49
browser details YourSeq           45     2    50    50  96.0%    18   -    11526201  11526249    49
browser details YourSeq           45     2    50    50  96.0%    17   -    70552782  70552830    49
browser details YourSeq           45     2    50    50  96.0%    14   -    20192460  20192508    49
browser details YourSeq           45     2    50    50  96.0%    11   -    84337821  84337869    49
```

position/search chr4:4,100,783-4,100,932 size 150 bp.

jump
clear
configure
chr4 (p16.2)
14
12
25
26
chr4:
4100800
4100850
4100900
Your Sequence from Blat Search
YourSeq
RefSeq Genes
Simple Nucleotide Polymorphisms (dbSNP build 128)
SNPs (128)
Simple Nucleotide Polymorphisms (dbSNP build 126)
rs6446712
Repeating Elements by RepeatMasker
SINE
LINE
LTR
DNA
Simple
Low Complexity
Satellite
RNA
Other
Unknown
Encyclopedia of DNA Elements (ENCODE) Regions
ENCODE Regions

Figure 6.6: Screenshots of the BLAT search for probe GL42657060-S. The top screen just some of the locations that the probe matched to. The bottom screen shows the genomic location of the top match. The probe sequence is seen to be outside any RefSeq genes and matching a region of the genome with a long-term repeat item.

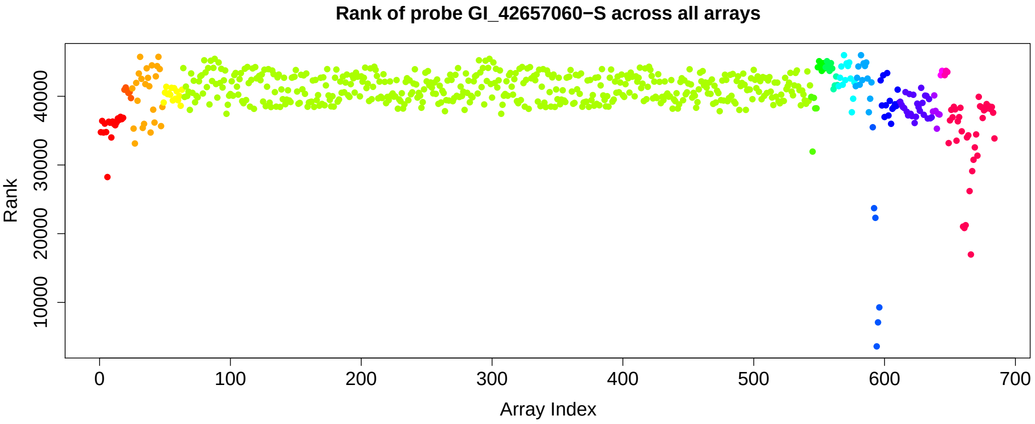


Figure 6.7: The rank of probe GI_42657060-S across all Illumina Human6 Version 1 arrays in GEO. It can be seen that regardless of sample or processing method, the probe is generally among the highest ranked probes on an array. The different colours represent the different datasets.

malisation (see page 31) and a list of 897 transcripts were identified as being significantly different after statistical testing using Illumina's DiffScore statistic. Of these 816 probes, 90% were present on the Human8 chip. Biological significance in the findings was assessed using the online DAVID resource (Dennis *et al.*, 2003) and relevant groups of genes were discovered, some of which are presented in the paper.

Table 3 of Bykhovskaya *et al.* (2007) lists 31 genes from the human ribosome KEGG pathway which were all found to be statistically significant when tested individually. Moreover, testing the pathway as a whole using the DAVID software gave a p-value of 2.8×10^{-19}, which is fairly convincing evidence that this pathway is biologically relevant for the biological condition under investigation. However, we have already seen that the genes that have this particular biological function tend to have very high rank on any array we look at. Also, 10 out of the 31 probes have an “OtherHit” and actually target other known transcripts.

In the case of the gene representing the protein S16 (accession number NM_001010), the median rank across all arrays is 47,236. This gene gave a p-value of 0.000414 in the statistical test, but it is easy to imagine that this is an artefact of data processing, especially if the mean-variance dependancy has not been accounted for appropriately in data processing or statistical testing. By using the function `normalize.invariantset` in affy, I calculated a set of rank-invariant genes for each array with respect to a target distribution containing the average intensity of each gene over all arrays. The results showed that 25 of the genes in Table 3 of Bykhovskaya *et al.* (2007) are considered part of the rank-invariant set of genes for every array. In other words their ranks are not significantly different from the distribution of an average array. It is likely that the perceived differential expression is due to the inability to measure accurately intensities at the high end of the intensity range, which are easily affected by batch effects. The normalisation used for this dataset would be unable to cope with such effects, since the calibration of each array by the mean value will have little effect at high intensities.

6.4 Discussion

After investigating a large collection of datasets derived using Illumina's Human6 chips, it seems that the issue of probe annotation is also important for more realistic experimental setups than the spike-in experiment. Firstly, not only are probes located on Strip 1 more often expressed, they are also more often called DE by a variety of analysis methods. Despite the technology providing measurements for around 48,000 transcripts, a filtering procedure will reduce this list by about 50% (depending on the samples used and filtering criteria). Moreover, the probes included in this reduced list are not represented equally by the two strips and the majority will be found on Strip 1. Carrying through to statistical testing, we again see that most significant findings are from Strip 1.

An important design issue is raised by this chapter: if the majority of significant findings are due to Strip 1, which is roughly equivalent to the Human8, should all experiments be run on the Human8 chip? The answer to this question largely depends on the expected outcomes of the microarray experiment. For studies involving multiple platforms, there generally needs to be a set of genes that are represented on all platforms. Although there might be a temptation to use "state-of-the-art" arrays with the most coverage, the extra genes included on these high-density arrays might not be present on all platforms in the study.

Human8 chips can be run at lower cost, with more samples in parallel and provide transcript measurements for curated, reliable content. Moreover, these transcripts can be more readily used for pathway or GO enrichment analysis. If a researcher has a small number of samples and wants to find what pathways are relevant to their study, then using these chips may be more cost-effective. The down-side to using the Human8 is that they potentially will not discover anything about the rarer transcripts that are provided on the Human6. One solution would be to use a Human6 chip to discover pathways in the analysis and then keep the data for the rare transcripts for reference in case they are required at a later date.

Reannotation of the Human6 chip also reveals many Strip 2 probes to be unreliable or mapping to regions not expected to produce signal. This might be expected as these probes generally represent less common transcripts, and therefore it is more challenging to design probes for them. If a Strip 2 probe were to appear to be statistically significant, one would have to refer to the reannotation to help judge if it is biologically relevant or not. Accommodating all 48,000 probes into the analysis may result in many false negatives and potentially interesting findings being missing. This is shown by the results of filtering on the MAQC dataset which already had many DE genes. Ideally, standard filtering techniques are designed to remove uninformative probes from the analysis. Having access to our reannotation information provides extra information that can be used to decide which genes are uninformative, rather than just relying on arbitrary cut-offs.

For Illumina data, the most common approach is to use the detection scores provided by Illumina, which concentrates on removing probes with low expression level across all samples being investigated. Whilst this should work in most cases, it is not difficult to find examples in the literature of probes mapping to intronic or intergenic regions with high expression level in any sample. Moreover, many housekeeping genes can be found that have high expression levels on all arrays. Such probes would not be removed by filtering, and in some cases can be selected as being DE due to inadequate data processing. The expression level of a gene should not be used as the sole criterion to judge whether to exclude genes from the analysis. A cut-off based on variability might be preferred in conjunction with annotation considerations. It seems apparent that intronic or intergenic probes are not likely to appear as significant in a differential expression analysis, and even if they do, the results are difficult to interpret. However, probes with mismatches compared to the genome could carry some useful information and are often found to be DE. Future versions of the reannotation script will report the base positions at which the mismatches occur and this could be used to inform the filtering, as we would expect mismatches in the middle of the sequence to affect the hybridisation to a greater degree.

Misannotated probes should be treated with care as they complicate our interpretation of the analysis. Single instances of misannotated probes occurring in a list of DE genes are easily dealt with if we are seeking a list of the top ranked N genes, since they can simply be removed. However, following up such findings may be a waste of resources and may have meant the exclusion of other genes which just missed the cut-off for being declared DE. Another problem occurs when we try to make inferences based on the list of DE genes, in particular when trying to find common properties such as pathways of functional annotation.

Unfortunately, it seems that the reported observation about ribosomal proteins in Bykhovskaya *et al.* (2007) is nothing more than a technical artefact. Looking at the behaviour of these genes over a large collection of arrays revealed them to have high rank regardless of the sample processing. Thus, the intensities of the genes are susceptible to batch effects and inadequate processing or transformation. A simple $\log_2$ transformation would have reduced the influence of these genes. Because a large collection of genes from the same pathway were considered to be DE when tested individually, the pathway as a whole was deemed to have high biological relevance. However, when the p-value for the pathway is calculated by DAVID (or similar tool) the calculation generally assumes a certain size of genes in the pathway. Care should be taken to make sure that these calculations are not affected by biases caused by many genes in the pathway having unreliable annotation. The results of smaller pathways could be dramatically affected, even if one gene was incorrectly called DE, which could falsely increase the evidence for the whole pathway to be DE. Further research is required into how annotation may be weighted in a GO analysis.

In summary, this chapter demonstrates the importance of probe annotation for Illumina arrays and shows how it can be incorporated into a standard differential expression analysis. The development of the script to perform the reannotation was refined in response to the results of this chapter. In particular the existence of repetitive elements will be accounted for in future versions, along with the position in the sequence that a mismatch occurs. The resulting annotation tables are freely available for researchers. In the future these will be incorporated into a Bioconductor annotation package, replacing the existing illuminaHumanV1 package, for ease of use. The beadarray package will also include functionality to incorporate annotation information into a standard analysis.

It should be noted that subsequent revisions of the annotation used on the Illumina Human6 chips have taken place, with versions 2 and 3 now available. Naturally, there would need to be a thorough investigation into the properties of these annotations. Such an investigation can be informed by the results from Version 1 and these platforms have already been reannotated using the script presented in (Barbosa-Morais *et al.*, 2008). However, data from these platforms are still quite scarce.

Conclusions

During the course of this thesis, I have described many important aspects in the analysis of Illumina microarray data. Investigations into new methodologies were initially hampered by the default analysis pipeline using bead-summary data as the starting point for analysis and the lack of a suitable dataset. The work presented in this thesis has made significant contributions in this regard.

The beadarray software allows analysis to commence with the raw data and puts every step under the control of the analyst. Therefore, the entire analysis can be documented and researchers are also able to judge the performance of each step and evaluate alternatives. Thus, a custom analysis can be produced without relying on the summarised data produced by Illumina, which may not be appropriate for all use-cases. Improved QA is also possible with bead-level data using methods arising from other microarray technologies. For example, imageplots can identify significant spatial artefacts, which were not previously reported for Illumina data as the facility did not exist to visualise them systematically. These and other bead-level diagnostics can be generated for any Illumina assay due to the raw data being identical for all assays. One ongoing project is to develop better diagnostic plots for Illumina data that can be generated automatically for large scale experiments. Guidelines should also be derived on how users can identify "bad" arrays without having to manually check all the diagnostic plots.

The publication of an Illumina spike-in dataset should encourage the development of new methods, especially as the bead-level data are available. In this thesis, the spike-in data were used to investigate background correction, summarisation and, to some extent, normalisation. Background correction applied at the bead-level was found to be adequate, but accounting for non-specific hybridisation using the negative control probes ("background normalisation") was found to be sub-optimal is its current form, especially if combined with a $\log_2$ transformation. An independently-developed transformation for Illumina data was found to alleviate some of these problems in the context of a gene expression study. However, similar performance was found by using the variances obtained from bead-level data to weight observations accordingly. Further investigation is required into the best way of correcting for effects such as non-specific hybridisation, as background normalisation seems a crude way of doing so given that sequence-specific effects are seen for non-spikes, which are not expected to show differential expression.

As well as the base-compostion of probe sequences, I also considered the fundamental problem of where these sequences map in the genome. A surprising conclusion from the spike-in analysis is the frequency with which sequences may not map to the intended transcript. Such issues have been reported for Affymetrix, but not for Illumina at the time of writing. Inclusion of these misannotated probes in the analysis could lead to significant findings being missed or declaring something to be significant, when actually the signal being measured is due to hybridisation of targets other than the intended one. This is important information to communicate to Illumina users, and the planned release of the reannotation of popular Illumina arrays to Bioconductor should also help in this regard. The reannotation can be used as criteria for excluding misleading probes instead of having to rely on ad-hoc criteria such as expression level or variability. In this thesis, these annotation issues have only been investigated for Version 1 Human and Mouse chips. At the time of writing, these are the chips with the most data available. Naturally, one should check if the annotation of subsequent version of Illumina chips has been improved.

The newer Illumina chips also have a different design in that all 48,000 probes are located on both strips for a given array. Thus, we should not see the intensity difference observed due to all curated content being placed on one strip. If the intensity differences persist on these newer chips, then the summarised values for each gene could be significantly affected by unequal numbers of beads between the two strips. The default processing methods within BeadStudio will ignore this effect and hence being able to analyse bead-level data through beadarray will be essential. A fundamental design question is raised by the strip difference seen for Version 1 chips and researchers should question whether the inclusion of rare transcripts is beneficial for their particular study. For published studies using Human6 chips, the number of significant findings arising from Strip 2 was very low. Also, for GO term enrichment and pathways analysis and comparisons between platforms, reliable probe annotation is a necessity and therefore Strip 2 is of little benefit.

To conclude, this thesis gives an in-depth portrayal of the process of developing new tools for an emerging technology. Although this technology has many unique features appealing to bionformaticians and biologists alike, the analysis of Illumina data can benefit greatly from lessons learnt from other microarray technologies. Although the work described in this thesis concentrates on gene expression studies, I have worked on data from other Illumina assays and the issues of QA and starting analysis at the bead-level are still relevant. Researchers are becoming increasingly aware of bead-level data and this is due to my efforts in publicising these data. The provision of open-source software and example datasets provided by this work should further advance our understanding of this technology and the unique challenges it offers.

Appendix A

Papers published during this work

The following list details papers published during the course of this thesis, or are in the advanced stages of preparation. Papers marked with a (*) will be explicitly discussed.

- *N.L. Barbosa-Morais, M.J. Dunning, M.E. Ritchie, A.G. Lynch, S. Tavaré. Reannotation of Illumina BeadArray probes improves the interpretation of gene expression data. *In Preparation* 2008

- A. Lynch, M. Dunning, M. Iddawela, N. Barbosa-Morais, and M. Ritchie. Considerations for the processing and analysis of GoldenGate based two-colour Illumina platforms *Statistical Methods in Medical Research* Accepted 2008

- A. Git, I. Spiteri, C. Blenkiron, M. Dunning, J.C.M Pole, S.F. Chin, Y. Wang, J. Smith, F.J. Livesey, and C. Caldas. PMC42, a breast progenitor cancer cell line, has normal-like mRNA and miRNA transcriptomes *Breast Cancer Research* 2008 June:10(R54)

- *M.J. Dunning, M.E. Ritchie, N.L. Barbosa-Morais, S. Tavaré S, and A.G. Lynch. Spike-in validation of an Illumina-specific variance stabilizing transformation. *BMC Research Notes* 2008 1(18)

- *M.J. Dunning, N.L. Barbosa-Morais, A.G. Lynch, S. Tavaré S, and M.E. Ritchie. Statistical issues in the analysis of Illumina data. *BMC Bioinformatics* 2008 8(85)

- C. Blenkiron, L.D. Goldstein, N.P. Thorne, I. Spiteri, S.F. Chin, M.J. Dunning, N.L. Barbosa-Morais, A.E. Teschendorff ,A.R. Green, I.O. Ellis, S. Tavaré, C. Caldas, and E.A. Miska. MicroRNA expression profiling of human breast cancer identifies new markers of tumour subtype. *Genome Biol.* 2007 Oct 8;8(10):R214

- B.E. Stranger, A.C. Nica, M.S. Forrest, A. Dimas, C.P. Bird, C. Beazley, C.E. Ingle, M.J. Dunning, P. Flicek, D. Koller, S. Montgomery, S. Tavaré, P. Deloukas, and E.T. Dermitzakis. Population genomics of human gene expression *Nat Genet.* 2007 Oct;39(10) :1217-24.

- *M.J. Dunning, M.L. Smith, M.E. Ritchie, and S. Tavaré. beadarray: R classes and methods for Illumina bead-based data. *Bioinformatics.* 2007 Aug 15;23(16):2183-4. Epub 2007 Jun 22.

- B.E. Stranger, M.S. Forrest, M. Dunning, C.E. Ingle, C. Beazley, N. Thorne, R. Redon, C.P. Bird, A. de Grassi, C. Lee, C. Tyler-Smith, N. Carter, S.W. Scherer, S. Tavaré, P. Deloukas, M.E. Hurles, and E.T. Dermitzakis. Relative impact of nucleotide and copy number variation on gene expression phenotypes. *Science.* 2007 Feb 9;315(5813):848-53.

- *M.J. Dunning, M. Smith, N.P. Thorne, and S. Tavaré. beadarray: an R package to analyse Illumina BeadArrays. *R News*, 6(5), 17-23, 2006.

- *M.J. Dunning, N.P. Thorne, I. Camilier, M.L. Smith, and S. Tavaré. Quality control and low-level statistical analysis of Illumina BeadArrays. *REVSTAT*, 4, 1-30, 2006.

Appendix B

Code for Chapter 4

This chapter contains the R code required for the analysis of the BioC07 dataset in Chapter 4 (Preliminary Investigation into low-level Illumina data) and assumes that the beadarray has been installed and file SAMExample.zip downloaded into the current R working directory. These commands are intended to give a guide to how the figures and data referred to in the chapter were generated. Therefore to save space, some of the graphical options (e.g. colours and labels of plots have been omitted.

First we load the package and read the bead level data

```
 library(beadarray)

targets = read.table("targets.txt", sep="\t", header=TRUE, as.is=TRUE)

BLData = readIllumina(arrayNames=targets$arrayID, textType=".csv",
+ targets=targets, backgroundMethod="none")
BLData.bc = backgroundCorrect(BLData)
an=arrayNames(BLData)
```

Boxplots of the foreground, background and background corrected intensities can be generated as follows

```
##Boxplots of foreground and background

ylim = c(4,16)
par(mfrow=c(1,3))
boxplotBeads(BLData,ylim=ylim)
boxplotBeads(BLData,ylim=ylim)
boxplotBeads(BLData.bc,ylim=ylim)
```

Now generating imageplots and plots of outlier locations

```
par(mfrow=c(2,5))
zlim = c(6,16)
for(i in 1:10){
  imageplot(BLData.bc, array=i, nrow=50, ncol=50, high="red", low="yellow")
}

##Plot outlier locations for 3 arrays with apparent spatial aretefacts

par(mfrow=c(1,3))
for(i in c(1,3,6)){
  o=findAllOutliers(BLData.bc, array=i)
  plotBeadLocations(BLData.bc, array=i, BeadIDs=o,SAM=TRUE)
}

##Calculate number of outliers

outliers = NULL
for(i in 1:10) {
  outliers[i] = length(findAllOutliers(BLData.bc, array=i))
}

outliers/numBeads(BLData)*100
```

Now create bead summary data, which uses the default method of Illumina, and make boxplots of expression values and number of beads.

```
BSData = createBeadSummaryData(BLData, imagesPerArray=1)

##Boxplots of expression values and number of beads

par(mfrow=c(1,2))
boxplot(as.data.frame(log2(exprs((BSData)))))
boxplot(as.data.frame(NoBeads(BSData)[-1265,]))



par(mfrow=c(2,3))
plotMA(exprs(BSData), 1,2)
plotMA(exprs(BSData), 1,3)
plotMA(exprs(BSData), 2,3)
plotMA(exprs(BSData),6,7)
plotMA(exprs(BSData),6,8)
plotMA(exprs(BSData),7,8)

##Correlations between replicates

cor(exprs(BSData))
```

A simple DE analysis after applying a quantile normalisation on $\log_2$ transformed data. We will fit two linear models using limma; the first model to all 10 arrays and the second with Array 1 removed. The effect on the volcano plot can be used to judge the difference of removing the array.

```
normData = normaliseIllumina(BSData,transform="log2")

design = matrix(nrow=10, ncol=2,0)
design[1:5,1]=1
design[6:10,2]=1
colnames(design) = LETTERS[1:2]

fit = lmFit(exprs(normData), design)

contrast = makeContrasts(AvsB = A -B, levels=design)

AvsB = contrasts.fit(fit,contrast)

ebFit = eBayes(AvsB)


fit2 = lmFit(exprs(normData)[,-1], design[-1,])

AvsB2 = contrasts.fit(fit2, contrast)

ebFit2 = eBayes(AvsB2)


par(mfrow=c(1,2))

volcanoplot(ebFit)
volcanoplot(ebFit2)
```

Appendix C

Supplementary Figures

This appendix shows modified versions of selected TIFF images from the BioC07 dataset. The contrast and colour balance of each image have been modified in order to highlight spatial artefacts on the arrays. Obviously making such adjustments manually is time-consuming for large experiments and unnecessary if we have tools to automatically detect such artefacts.

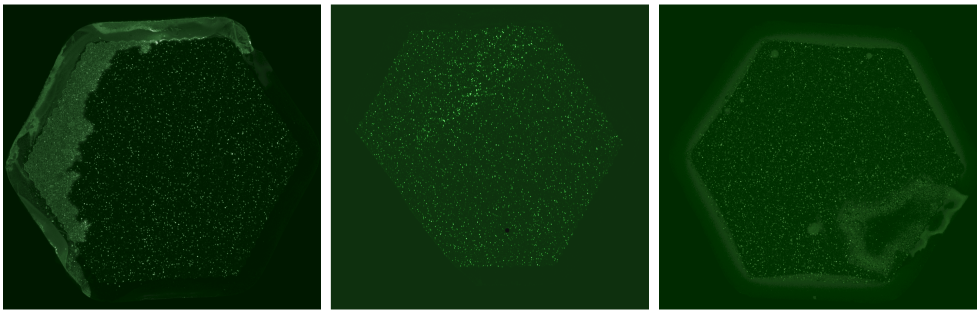


Figure C.1: Modified TIFF images for arrays 1, 3 and 6 from the BioC07 dataset.

Bibliography

Allison, D., X. Cui, G. Page, and M. Sabripou, 2006 Microarray data analysis: from disarray to consolidation and consensus. Nature Reviews: Genetics **7**: 55–65.

Barbosa-Morais, N., M. Dunning, M. Ritchie, A. Lynch, and S. Tavaré, 2008 Reannotation of Illumina BeadArray probes improves the interpretation of gene expression data. In Preparation .

Barnes, M., J. Freudenberg, S. Thompson, B. Aronow, and P. Pavlidis, 2005 Experimental comparison and cross-validation of the Affymetrix and Illumina gene expression analysis platforms. Nucleic Acids Research **33**: 5914–5923.

Barrett, T., D. Troup, S. Wilhite, P. Ledoux, D.Rudnev, *et al.*, 2007 NCBI GEO: mining tens of millions of expression profiles database and tools update. Nucleic Acids Research **35**: D760–765.

Benjamini, Y., and Y. Hochberg, 1995 Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society. Series B (Methodological) **57**: 289–300.

Bolstad, B., R. Irizarry, M. Astrand, and T. Speed, 2003 A comparison of normalization methods for high density oligonucleotide array data based on bias and variance. Bioinformatics **19**: 185–193.

Bykhovskaya, Y., E. Mengesha, and N. Fischel-Ghodsian, 2007 Pleiotropic effects and compensation mechanisms determine tissue specificity in mitochondrial myopathy and sideroblastic anemia (MLASA). Molecular Genetics and Metabolism **91**: 148–156.

Calza, S., W. Raffelsberger, A. Ploner, J. Sahel, T. Leveillard, *et al.*, 2007 Filtering genes to improve sensitivity in oligonucleotide microarray data analysis. Nucleic Acids Research **35**: e102.

Choe, S., M. Boutros, A. Michelson, G. Church, and M. Halfon, 2005 Preferred analysis methods for Affymetrix GeneChips revealed by a wholly defined control dataset. Genome Biology **6**.

Chuaqui, R. F., R. Bonner, C. Best, J. Gillespie, M. Flaig, *et al.*, 2002 Post-analysis follow-up and validation of microarray experiments. Nature Genetics **32**: 509–514.

Chudin, E., S. Kruglyak, S. C. Baker, S. Oeser, D. Barker, *et al.*, 2006 A model of technical variation of microarray signals. J Comput Biol **13**: 996–1003.

Cope, L. M., R. A. Irizarry, H. A. Jaffee, Z. Wu, and T. P. Speed, 2004 A benchmark for Affymetrix GeneChip expression measures. Bioinformatics **20**: 323–331.

Cui, X., and G. Churchill, 2003 Statistical tests for differential expression in cdna microarray experiments. Genome Biology **4**.

Dabney, A., and J. Storey, 2006 A reanalysis of a published Affymetrix GeneChip control dataset. Genome Biology **7**.

Dai, M., P. Wang, A. Boyd, G. Kostov, B. Athey, *et al.*, 2005 Evolving gene/transcript definitions significantly alter the interpretation of GeneChip data. Nucleic Acids Research **33**.

Davis, S., 2008 *GEOquery: Get data from NCBI Gene Expression Omnibus (GEO).* R package version 2.4.0.

de Jonge, H., R. Fehrmann, E. D. Bont, R. Hofstra, F. Gerbens, *et al.*, 2007 Evidence based selection of housekeeping genes. PLoS ONE **9**: e898.

Dennis, G., B. Sherman, D. Hosack, Y. J, W. Gao, *et al.*, 2003 DAVID: Database for Annotation, Visualization and Integrated Discovery. Genome Biology **4**.

Deregibus, M., V. Cantaluppi, R. Calogero, M. L. Iacono, C. Tetta, *et al.*, 2007 Endothelial progenitor cell derived microvesicles activate an angiogenic program in endothelial cells by a horizontal transfer of mRNA. Blood **110**: 2440–2448.

Du, P., W. Kibbe, and S. Lin, 2007 nuID: a universal naming scheme of oligonucleotides for Illumina, Affymetrix, and other microarrays. Biology Direct **2**.

Du, P., W. Kibbe, and S. Lin, 2008 lumi: a pipeline for processing illumina microarray. Bioinformatics **Epub ahead of print**.

Dudoit, S., Y. Yang, M. Callow, and T. Speed, 2002 Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Statistica Sinica **12**: 111–140.

Dunning, M., N. Barbosa-Morais, A. Lynch, S.Tavaré, and M. Ritchie, 2008a Statistical issues in the analysis of Illumina data. BMC Bioinformatics **8**.

Dunning, M. J., M. Ritchie, N. Barbosa-Morais, S. Tavaré, and A. Lynch, 2008b Spike-in validation of an Illumina-specific variance stabilizing transformation. BMC Research Notes **1**.

Dunning, M. J., M. L. Smith, M. E. Ritchie, and S. Tavaré, 2007 beadarray: R classes and methods for Illumina bead-based data. Bioinformatics **23**: 2183–2184.

Dunning, M. J., M. L. Smith, N. P. Thorne, and S. Tavaré, 2006a beadaray: an R package to analyse Illumina BeadArrays. Rnews **6**.

Dunning, M. J., N. P. Thorne, I. Camilier, M. L. Smith, and S. Tavaré, 2006b Quality control and low-level statistical analysis of Illumina BeadArrays. RevStat **4**: 1–30.

Durbin, B., J. Hardin, D. Hawkins, and D. Rocke, 2002 A variance-stabilizing transformation for gene-expression microarray data. Bioinformatics **18**: S105–10.

Edwards, D., 2002 Non-linear normalization and background correction in one-channel cDNA microarray studies. Bioinformatics **19**: 825–833.

Eggle, D., and J. Schultze, 2007 IlluminaGUI: Graphical User Interface for analyzing gene expression data generated on the Illumina platform. Bioinformatics **23**: 1431 – 1433.

Elvidge, G., L. Glenny, R. Appelhoff, P. Ratcliffe, J. Ragoussis, *et al.*, 2006 Concordant regulation of gene expression by hypoxia and 2-oxoglutarate-dependent dioxygenase inhibition: the role of HIF-1alpha, HIF-2alpha, and other pathways. Journal of biological chemistry. **281**: 15215–15226.

Falcon, S., and R. Gentleman, 2007 Using GOstats to test gene lists for GO term association. Bioinformatics **23**: 257–258.

Fan, J. B., K. L. Gunderson, M. Bibikova, J. M. Yeakley, J. Chen, *et al.*, 2006 Illumina universal bead arrays. Methods Enzymol **410**: 57–73.

Gallinsky, V., 2003 Automatic registration of microarray images. I Rectangular grid. Bioinformatics **19**: 1824 – 1831.

Gautier, L., L. Cope, B. Bolstad, and R. Irizarry, 2004a affy–analysis of Affymetrix GeneChip data at the probe level. Bioinformatics **20**: 307–15.

Gautier, L., M. Moller, L. Friis-Hansen, and S. Knudsen, 2004b Alternative mapping of probes to genes for Affymetrix chips. **5**.

Gentleman, R., V. Carey, W. Huber, and F. Hahne, 2008 *genefilter: methods for filtering genes from microarray experiments*. R package version 1.16.0.

Gentleman, R. C., V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, *et al.*, 2004 Bioconductor: Open software development for computational biology and bioinformatics. Genome Biology **5**: R80.

Golubkov, V., A. Chekanov, A. Savinov, D. Rozanov, N. Golubkova, *et al.*, 2006 Membrane type-1 matrix metalloproteinase confers aneuploidy and tumorigenicity on mammary epithelial cells. Cancer Research **66**: 10460–5.

Greber, B., H. Lehrach, and J. Adjaye, 2007 Silencing of core transcription factors in human EC cells highlights the importance of autocrine FGF signaling for self-renewal. BMC Developmental Biology **7**.

Gunderson, K. L., S. Kruglyak, M. S. Graige, F. Garcia, B. G. Kermani, *et al.*, 2004 Decoding randomly ordered DNA arrays. Genome Research **14**: 870–877.

HapMap Consortium, I., 2003 The International HapMap Project. Nature **18**: 789–96.

Harbig, J., R. Sprinkle, and S. Enkemann, 2005 A sequence-based identification of the genes detected by probesets on the Affymetrix U133 plus 2.0 array. Nucleic Acids Research **33**: e31.

Holloway, A., A. Oshlack, D. S. Diyagama, D. Bowtell, and G. Smyth, 2006 Statistical analysis of an RNA titration series evaluates microarray precision and sensitivity on a whole-array basis. BMC Bioinformatics **7**.

Huber, W., A. V. Heydebreck, H. Sultmann, A. Poustka, and M. Vingron., 2002 Variance stabilization applied to microarray data calibration and to the quantfication of differential expression. Bioinformatics **18**: S960S104.

Ihaka, R., and R. Gentleman, 1996 R: A language for data analysis and graphics. Journal of computational and graphical statistics **5**: 299–314.

Irizarry, R., B. Bolstad, F. Collin, L. Cope, B. Hobbs, *et al.*, 2003a Summaries of Affymetrix GeneChip probe level data. Nucleic Acids Research **31**.

Irizarry, R., L. Cope, and Z. Wu, 2006 Feature-level exploration of a published Affymetrix GeneChip control dataset. Genome Biology **7**.

Irizarry, R., B. Hobbs, F. Collin, Y. Beazer-Barclay, K. Antonellis, *et al.*, 2003b Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics **4**: 249–264.

Kanehisa, M., and S. Goto, 2000 KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Research **28**: 27–30.

Kent, W., 2002 BLAT—the BLAST-Like Alignment Tool. Genome Research **12**: 656–664.

Kooperberg, C., T. Fazzio, J. Delrow, and T. Tsukiyama, 2002 Improved background correction for spotted DNA microarrays. Journal of Computational Biology **9**: 55–56.

Krig, S., V. Jin, M. Bieda, H. O'Geen, P. Yaswen, *et al.*, 2007 Identification of genes directly regulated by the oncogene ZNF217 using chromatin immunoprecipitation (ChIP)-chip assays. Journal of biological chemistry **282**: 9703–9712.

Kuhn, K., S. C. Baker, E. Chudin, M. H. Lieu, S. Oeser, *et al.*, 2004 A novel, high-performance random array platform for quantitative gene expression profiling. Genome Research **14**: 2347–2356.

Kuhn, R., D. Karolchik, A. Zweig, H. Trumbower, D. Thomas, *et al.*, 2007 The UCSC Genome Browser database: update 2007. Nucleic Acids Research **35**: D668–73.

Leisch, F., 2002 Sweave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle and B. Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*. Physica Verlag, Heidelberg, 575–580.

Lenk, G., G. Tromp, S. Weinsheimer, Z. Gatalica, R. Berguer, *et al.*, 2007 Whole genome expression profiling reveals a significant role for immune function in human abdominal aortic aneurysms. BMC Genomics **8**.

Li, C., and W. Wong, 2001 Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. Genome Biology **2**.

Lin, S., P. Du, W. Huber, and W. Kibbe, 2008 Model-based variance-stabilizing transformation for Illumina microarray data. Nucleic Acids Research **36**.

Lockhart, D., H. Dong, M. Byrne, M. Follettie, M. Gallo, *et al.*, 1996 Expression monitoring by hybridization to high-density oligonucleotide arrays. Nature Biotechnology **14**: 1675 – 1680.

Lonnstedt, I., and T. Speed, 2002 Replicated microarray data. Statistica Sinica **12**: 31–46.

LYNCH, A., C. CURTIS, and S. TAVARÉ, 2007 Correcting for probe-design in the analysis of gene-expression. In P. B. S. Barber and K.V.Mardia, editors, *Systems Biology and Statistical Bioinformatics*. Leeds University.

MAQC CONSORTIUM, 2006 The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements.. Nat Biotechnol **24**: 1151–1161.

MCCLINTICK, J., and H. EDENBERG, 2006 Effects of filtering by preset call on analysis of microarray experiments. BMC Bioinformatics **7**.

PEARSON, R., 2008 A comprehensive re-analysis of the Golden Spike data: Towards a benchmark for differential expression methods. BMC Bioinformatics **9**.

PLATTS, A., D. DIX, H. CHEMES, K. THOMPSON, R. GOODRICH, *et al.*, 2007 Success and failure in human spermatogenesis as revealed by teratozoospermic RNAs. Human Molecular Genetics **16**: 763–773.

PRUITT, K., T. TATUSOVA, and D. MAGLOTT, 2007 NCBI reference sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Research **1**: D61–5.

QUACKENBUSH, J., 2002 Microarray data normalization and transformation. Nature Genetics **32**: 496 – 501.

RAMILO, O., W. ALLMAN, W. CHUNG, A. MEJIAS, M. ARDURA, *et al.*, 2007 Gene expression patterns in blood leukocytes discriminate patients with acute infections. Blood **109**: 2066–77.

RITCHIE, M. E., D. DIYAGAMA, J. NEILSON, R. VAN LAAR, A. DOBROVIC, *et al.*, 2006 Empirical array quality weights in the analysis of microarray data. BMC Bioinformatics **7**: 2105–2107.

RITCHIE, M. E., J. SILVER, A. OSHLACK, M. HOLMES, D. DIYAGAMA, *et al.*, 2007 A comparison of background correction methods for two-colour microarrays. Bioinformatics **23**: 2700–2707.

SANDBERG, R., and O. LARSSON, 2007 Improved precision and accuracy for microarrays using updated probe set definitions. BMC Bioinformatics **8**.

Scharpf, R., J. S. C.A. Iacobuzio-Donahue, and G. Parmigiani, 2006 When should one subtract background fluorescence in 2-color microarrays? Biostatistics **8**: 695–707.

Schena, M., D. Shalon, R. Davis, and P. Brown, 1995 Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science **270**: 467–470.

Scholtens, D., and A. V. Heydebreck, 2005 Analysis of differential gene expression studies. In R. Gentleman, V. Carey, S. Dudoit, R. .Irizarry and W. Huber, editors, *Bioinformatics and Computational Biology Solutions using R and Bioconductor*. Springer, New York, 229–248.

Smyth, G., and T. Speed, 2003 Normalization of cDNA microarray data. Methods **31**: 265–273.

Smyth, G., Y. Yang, and T. Speed, 2003 Statistical issues in cDNA microarray data analysis. Methods in Molecular Biology **224**: 111–136.

Smyth, G. K., 2004 Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. Stat Appl Genet Mol Biol **3**: Article 3.

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

Stajich, J., D. Block, K. Boulez, S. Brenner, S. Chervitz, *et al.*, 2002 The Bioperl toolkit: Perl modules for the life sciences. Genome Research **12**: 1611–1618.

Stranger, B., M. Forrest, A.G., Clark, M.J., *et al.*, 2005 Genomewide associations of gene expression variation in humans. PLoS Genet **1**: e78.

Stranger, B., M. Forrest, M. Dunning, C. Ingle, C. Beazley, *et al.*, 2007 Relative impact of nucleotide and copy number variation on gene expression phenotypes. Science **315**: 848–53.

Su, A., M. Cooke, K. Ching, Y. Hakak, J. Walker, *et al.*, 2002 Large-scale analysis of the human and mouse transcriptomes. Proc Natl Acad Sci U S A **99**.

Tesar, P., J. Chenoweth, F. Brook, T. Davies, E. Evans, *et al.*, 2007 New cell lines from mouse epiblast share defining features with human embryonic stem cells. Nature **448**: 196–199.

The Gene Ontology Consortium, 2000 Gene Ontology: tool for the unification of biology. Nature Genetics **25**: 25–29.

Thorrez, L., K. V. Deun, L. Tranchevent, L. V. Lommel, K. Engelen, *et al.*, 2008 Using Ribosomal Protein Genes as Reference: A Tale of Caution. PLoS ONE **3**: e1854.

Tusher, V. G., R. Tibshirani, and G. Chu, 2001 Significance analysis of microarrays applied to the ionizing radiation response. Proc Natl Acad Sci U S A **98**: 5116–21.

van 't Veer, L., H. Dai, M. van de Vijver, Y. He, A. Hart, *et al.*, 2002 Gene expression profiling predicts clinical outcome of breast cancer. Nature **415**: 530–536.

Wang, L., T. Schulz, E. Sherrer, D. Dauphin, S. Shin, *et al.*, 2007 Self-renewal of human embryonic stem cells requires insulin-like growth factor-1 receptor and ERBB2 receptor signaling. Blood **110**: 4111–9.

Workman, C., L. Jensen, H. Jarmer, R. Berka, L. Gautier, *et al.*, 2002 A new non-linear normalization method for reducing variability in DNA microarray experiments. Genome Biology **3**.

Wu, Z., and R. A. Irizarry, 2005 Stochastic models inspired by hybridization theory for short oligonucleotide arrays. Journal of Computational Biology **12**: 882–893.

Yang, Y., M. Buckley, and T. Speed, 2001 Analysis of cDNA microarray images. Briefings in Bioinformatics **2**: 341–349.

Yang, Y., S. Dudoit, P. Lu, and T. Speed, 2002a Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide variation. Nucleic Acids Research **30**: e15.

Yang, Y. H., M. J. Buckley, S. Dudoit, and T. P. Speed, 2002b Comparison of methods for image analysis on cDNA microarray data. Journal of Computational and Graphical Statistics **11**: 108–136.