Topics on statistical design and analysis of cDNA microarray experiment

Ximin Zhu

A Dissertation Submitted to the

University of Glasgow

for the degree of

Doctor of Philosophy

Department of Statistics

May 2009

Abstract

A microarray is a powerful tool for surveying the expression levels of many thousands of genes simultaneously. It belongs to the new genomics technologies which have important applications in the biological, agricultural and pharmaceutical sciences.

In this thesis, we focus on the dual channel cDNA microarray which is one of the most popular microarray technologies and discuss three different topics:

- Optimal experimental design,
- Estimating the true proportion of true nulls, local false discovery rate (lFDR) and positive false discovery rate (pFDR),
- Dye effect normalization.

The first topic consists of four subtopics each of which is about an independent and practical problem of cDNA microarray experimental design. In the first subtopic, we propose an optimization strategy which is based on the simulated annealing method by Wit et al. (2005) to find optimal or near-optimal designs with both biological and technical replicates. In the second subtopic, we discuss how to apply Q-criterion for the factorial design of microarray experiments. In the third subtopic, we suggest an optimal way of pooling samples, which is actually a replication scheme to minimize the variance of the experiment under the

constraint of fixing the total cost at a certain level. In the fourth subtopic, we indicate that the criterion for distant pair design (Fu and Jansen, 2005) is not proper and propose an alternative criterion instead.

The second topic of this thesis is dye effect normalization. For cDNA microarray technology, each array compares two samples which are usually labelled with different dyes Cy3 and Cy5. It assumes that: for a given gene (spot) on the array, if Cy3-labelled sample has k times as much of a transcript as the Cy5-labelled sample, then the Cy3 signal should be k times as high as the Cy5 signal, and vice versa. This important assumption requires that the dyes should have the same properties. However, the reality is that the Cy3 and Cy5 dyes have slightly different properties and the relative efficiency of the dyes vary across the intensity range in a "banana-shape" way. In order to remove the dye effect, we propose a novel dye effect normalization method which is based on modeling dye response functions and dye effect curve. Real and simulated microarray data sets are used to evaluate the method. It shows that the performance of the proposed method is satisfactory.

The focus of the third topic is the estimation of the proportion of true null hypotheses, lFDR and pFDR. In a typical microarray experiment, a large number of gene expression data could be measured. In order to find differential expressed genes, these variables are usually screened by a statistical test simultaneously. Since it is a case of multiple hypothesis testing, some kind of adjustment should be made to the p-values resulted from the statistical test. Lots of multiple testing error rates, such as FDR, lFDR and pFDR have been proposed to address this issue. A key related problem is the estimation of the proportion of true null hypotheses (i.e. non-expressed genes). To model the distribution of the p-values, we propose three kinds of finite mixture of unknown number of components (the

first component corresponds to differentially expressed genes and the rest components correspond to non-differentially expressed ones). We apply a new MCMC method called allocation sampler to estimate the proportion of true null (i.e. the mixture weight of the first component). The method also provides a framework for estimating lFDR and pFDR. Two real microarray data studies plus a small simulation study are used to assess our method. We show that the performance of the proposed method is satisfactory.

Acknowledgements

I would like to take this opportunity to thank everyone who has supported me throughout the completion of this thesis.

Firstly, I would like to thank my supervisors Dr. Agostino Nobile and Prof. Ernst Wit, who contributed their time, expertise to this thesis. I would also like to thank my second supervisor Prof. Marian Scott for her very important help during my PhD studies. Thanks must also go to all the other members of the statistics department who have helped make the last few years a enjoyable experience. I am very grateful for all the opportunities the department gave me. Also, I must thank the Overseas Research Students (ORS) Awards Scheme for funding me throughout this PhD studies.

Finally, I am very grateful to my family and all my friends for all they have done for me over the years. I must thank my parents and my wife for their love, support and encouragement through both good and hard times.

Declaration

This thesis has been composed by myself and it has not been submitted in any previous application for a degree. The work reported within was executed by myself, unless otherwise stated.

May 2009

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Since worldwide efforts to sequence genomes began formally in 1990, rapid technological advances have been introduced so that over the past few years a large number of organisms have had their genomes completely sequenced, including yeast, worm, fly, mouse and human. But the billions of bases of DNA sequence do not tell us what all the genes do and how sets of genes interact with each other in the genome. In order to solve these problems, a lot of efforts are being made to the functional genomics which is an area of genome research concerned with assigning biological function to DNA sequences. For functional genomics new technologies are being applied to take full advantage of the large and rapidly increasing body of sequence information. Among the most powerful and versatile tools are DNA microarrays, which allow simultaneous monitoring of the expression levels of numerous genes.

The principle of a microarray experiment, as opposed to the classical northern-blotting analysis, is that mRNA from a given cell line or tissue is used to generate a labelled sample (sometimes termed the target), which is hybridized in parallel to a large number of DNA sequences (sometimes termed the probes), immobilized

on a solid surface in an ordered array. Tens of thousands of transcript species can be detected and quantified at the same time. Although many different microarray systems have been developed, the most commonly used systems today can be divided into two groups, according to the arrayed material: complementary DNA (cDNA) and oligonucleotide microarrays. High-density oligonucleotide microarray experiments provide direct information about the expression levels in a mRNA sample of the 200,000-500,000 probed DNA sequences. By contrast, cDNA microrarray experiments typically involve hybridizing two mRNA samples, each of which has been converted into cDNA and labelled with its own fluorophore (Cy3 and Cy5 dyes) respectively, on a single glass slide that has been spotted with as many as 10,000-20,000 cDNA probes. Data from such experiments provide information on the relative expression of the sample genes, which correspond to the probes.

Microarray experiments usually generate large and complex multivariate data sets, and some of the greatest challenges lie not in generating these data but in the development of statistics tools to design the experiment and analyse the large amount of data. In this thesis, our interest is the cDNA microarray and we try to discuss three different topics in the statistical analysis of the cDNA microarray experiments in Chapter 2, 3 and 4 respectively.

The first general topic is relevant to the optimal design of cDNA microarray experiments. As two samples can be applied or "hybridized" to a single cDNA microarray, the array is a blocking factor. Another nuisance factor is the two-level dye factor, as the gene expressions in the two samples on an array are measured via a Cy3 and a Cy5 dye. When more than two sample conditions or treatments are of interest, then not every sample can appear on an array so that some form of an incomplete-block design should be considered. This brings with it a challenge

how to design the experiments (i.e. which samples should be co-hybridized on a single array) so that the efficiency and reliability of the microarray data can be improved and the precise estimates of biologically important parameters can be obtained.

Many of the microarray designs currently used are the so-called reference designs. In this type of design, each sample condition of interest is compared with a fixed, standardized condition. Making all comparisons to a reference sample is however inefficient, because half of the hybridization resources are allocated to the reference sample, which is usually of little or no interest. Alternatives to reference designs have been suggested. Dye swap designs and loop designs have gained some popularity. However, Kerr and Churchill (2001*a*) have pointed out that dye swap designs are quite inefficient and loop designs are optimal only for a relatively small number of conditions. Wit et al. (2005) have shown how the application of a simple optimization algorithm, simulated annealing with local design moves, to incomplete-block designs with block size 2 can find optimal or near-optimal designs for given number of conditions and arrays based on different optimality criterion. However, this optimization strategy just assumes that for each sample condition (treatment) the number of independent biological replicates is not less than the total number of replicates needed. Unfortunately, in some cases this assumption does not stand (i.e. no enough biological replicates available) so that technical replicates have to be used. Then a question arises: How to assign the biological and technical replicates to the arrays in an optimal way? To deal with this problem, we follow the spirit of the simulated annealing framework for optimal design and develop a modified optimization strategy to find the optimal or near-optimal design and allocation of biological and technical replicates in the first part of Chapter 2.

In recent years more and more biologists begin to consider multi-factorial microarray experimental set-ups to identify differentially expressed genes, e.g. Caetano et al. (2004). The problem of how to find optimal (efficient) factorial design has received some attention. For example, Glonek and Solomon (2004) used A-optimality to find optimal designs of factorial experiment with a small number of factors. In the second part of Chapter 2, we use a new multi-factorial design optimality criterion called Q-optimality (Tsai et al., 2000) and show that under the simulated annealing framework it can be used to search near-optimal multi-factorial microarray experimental designs.

Statistical design of microarray aims at reducing unwanted variations to increase the precision of the quantities of interest. Pooling true biological RNA replicates is a cost-effective way to achieve this goal. In the third part of Chapter 2, we make some practical suggestions about optimal pooling samples for a microarray experiment. We find a replication scheme that minimizes the variance of the experiment under the constraint of fixing the total cost at a certain level.

Recently the combined study of gene expression and molecular marker data has been proposed as a novel strategy for the analysis of regulatory networks. Costs of such studies are high and require that resources microarrays and samples are used as efficiently as possible. Fu and Jansen (2005) propose a new design called distant pair design for this kind of studies, which co-hybridizes sample individuals with dissimilar genomes. The corresponding optimality criterion is defined for the case of single marker and is further extended to the case of multiple markers by simply averaging the criterion for single marker. We believe the extension is not very proper and propose a new criterion for the case of multiple markers as an alternative in the final part of Chapter 2.

The second topic in this thesis is about dye effect normalization. The current

technology of cDNA microarray is based on measuring optical intensities of dye labeled cDNA that has hybridized to gene-specific probes on the microarray. Two different types of dyes Cy3 and Cy5 are commonly used for the two samples on the array. Ideally, these two dyes should have the same properties so that the direct comparison between the two gene expression data of the two channels can be meaningful. However, the fact is that the dyes have slightly different properties and the relative efficiency of the dyes usually vary across the intensity range in a "banana-shape" way. In order to remove the dye effect as much as possible, several methods have been proposed, such as dye-swap normalization by Yang et al. (2002*b*) and intensity-dependent dye normalization (LOESS) by Yang and Speed (2003). In Chapter 3 we suggest a new dye effect normalization method based on modeling dye response functions and dye effect curve. The performance of our method is compared to LOESS by using simulated microarray gene expression data and real microarray data.

In a typical cDNA microarray experiment, a large number of gene expressions are usually measured. When these variables are simultaneously screened by a statistical test, it is necessary to consider the adjustment for multiple hypothesis testing. Quite a few error rates of multiple testing such as false discovery rate (FDR), positive false discovery rate (pFDR) and local false discovery rate (lFDR) have been proposed and widely used to address this issue. A related problem is the estimation of the proportion of true null hypotheses, π_0. In Chapter 4, we first review the background of multiple hypothesis testing and its error rates, then we deal with the estimation of π_0 by modeling p-values from the experiment with finite mixtures with unknown number of components. Three different mixture models are considered. A newly developed MCMC method, allocation sampler (Nobile and Fearnside, 2007) is not only applied to estimate π_0 but also pFDR

and lFDR for both real and simulated microarray gene expression data.

Since this thesis deals with three very different topics in the statistical analysis of cDNA microarray experiments in Chapter 2, 3 and 4, we include a more detailed introduction section for each of these chapters.

The Chapter 5 is a conclusion of the whole thesis and discussion of further potential research opportunities.

Chapter 2

Optimal design of cDNA microarray experiments

2.1 Introduction to cDNA microarray experimental design

Spotted complementary DNA (cDNA) microarray is a powerful and cost-effective technology which provides molecular biologists and geneticists with a tool to monitor thousands of genes simultaneously (Brown and Botstein, 1999). Since its introduction in 1995 (Schena et al., 1995), this revolutionary technology has greatly influenced and accelerated the molecular biological and medical research.

A cDNA microarray, also called two-channel microarray or spotted microarray, typically consists of thousands of microscopic spots of DNA oligonucleotides (gene). For each spot, it measures the relative abundance of the DNA samples (under two different treatments) hybridized to the spot. The experiment usually consists of several steps. First, pools of mRNA derived from experimental or

clinical samples under two treatments are reversed-transcribed into cDNA and labelled with Cy3 (green) and Cy5 (red) fluorescent dyes respectively. Second, the two labelled cDNA pools are mixed in equal proportions and hybridized to the probes on a solid surface (i.e. array), which can be glass or a silicon chip. The probes are synthesized prior to being spotted onto the array surface and can be oligonucleotides, cDNA or small fragments of PCR products that correspond to mRNAs. Third, probe-target hybridization occurs on the array: the probe catches the complementary matched cDNA and the unhybridized cDNA is washed away. Finally, the red and green signal intensities are separately read out for each spot on the array by a laser scanner. The ratio of the optical signal intensities represents the relative abundance of the corresponding mRNA under two treatments. A higher intensity of one treatment over the other means that the spot (gene) is more "active" under the former.

2.1.1 Microarray experimental effects

The primary objective of a microarray experiment is to look for changes in gene expression across factors of interest. The factor could be the different type of samples (tissues) or the different drug or stress treatments (conditions) or the different stages of a biological process (time points).

Basically, there are four microarray experimental effects:

1. Treatments (T): the categories of the factor of interest.

2. Genes (G): spotted sequences (e.g. genes, ESTs, or DNAs).

3. Dyes (D): Cy5 (red) and Cy3 (green) labels.

4. Arrays (A): number of arrays over which the hybridization is replicated.

Therefore there are 15 experimental effects in a microarray experiment in total, including four main effects (T, G, D, A), six two-factor interactions (TG, TD, TA, GD, GA, DA), four three-factor interactions (TGD, TGA, TDA, GDA) and one four-factor interactions (TGDA).

Treatment main effects (T) account for overall differences in treatments. Such differences could arise if some treatments have more transcription activity in general.

Gene main effects (G) occur when certain genes emit a higher or lower fluorescent signal overall, compared to other genes. These effects arise because some genes have generally higher or lower levels of expression than others irrespective of treatments, dyes or arrays.

Dye main effects (D) measure the difference in the two dye fluorescent labels. For example, one dye may be consistently brighter than the other when averaged over the other factors.

Array main effects (A) account for differences between arrays, averaged over all genes, dyes, and treatments. These effects arise if, for example, arrays are probed under inconsistent conditions that increase or reduce hybridization efficiencies of the labeled cDNA.

Treatment $\times$ Gene (TG) interactions arise when the relative expressions of specific genes are different from one treatment to the other (when averaged over arrays and dyes). This can be illustrated graphically in Figure 2.1 (a). These effects are the most important in the experiment and their identification and quantification is often the main objective of the experiment.

Dye $\times$ Gene (DG) interaction effects occur when differences in intensity between Red and Green dyes are different from one gene to the other. This can be illustrated graphically in Figure 2.1 (b). This can happen when cDNA sequences,

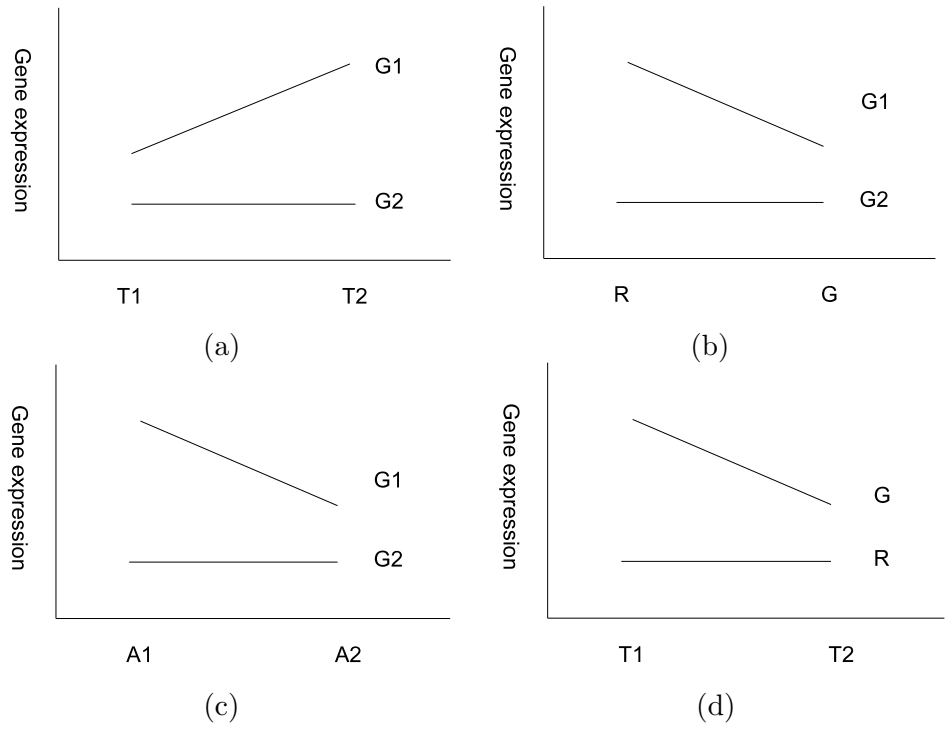


Figure 2.1: The four main effects result in six two-factor interactions (TG, TD, TA, GD, GA, DA). Here we illustrate the four most important interactions effects, which are (a) gene-treatment interaction, (b) gene-dye interaction, (c) gene-array interaction and (d) dye-treatment interaction.

matching specific genes on the chip, incorporate red dye molecules at a different rate than green molecules while sequences specific to other genes show the reverse trend. This effect is quite likely due to the chemistry of dye incorporation and so must be accounted for in any array experiment. Note that if this effect exists and has not been detected, estimates of relative expressions are biased and may lead to misleading results.

Array $\times$ Gene (AG) interaction effects or spot effects may arise because there is no complete control over the amount and concentration of cDNA immobilized from one array to the next. This can be illustrated graphically in Figure 2.1 (c).

A Dye $\times$ Treatment (DT) interaction effect for given gene A is shown in Figure

2.1 (d). It may occur in the experiment when one fluorescent dye hybridizes more with cDNA from treatment T1 than from T2, but the other dye is consistent. If this sort of effect was consistent over many genes and arrays, we should find a DT interaction. However, we do not always see this happen in practice.

Besides the above four main effects and four two-factor interaction effects, it is difficult to relate the other remaining 7 higher-order interaction effects to the microarray experimental process. For example, two-factor interaction effects like Array × Dye (AD), Array × Treatment (AT), and three- factor interaction effect like Array × Dye × Treatment (ADT) do not involve the genes. It is difficult to relate any of these to the process underlying microarrays and to suppose a reason why such interactions would come into play. Array × Dye × Gene (ADG), Array × Treatment × Gene (ATG), Dye × Treatment × Gene (ATG), and Array × Dye × Treatment × Gene (ADTG) effects all do involve the genes. The presence of such interactions would mean there is gene-specific variation attributable to a particular array and dye, a particular array and treatment, a particular dye and treatment, or a particular array, dye, and treatment combination. Again, these high-order interactions are difficult to relate to the physical and chemical processes that make up this technology and so they are generally assumed not to occur. This assumption should, however, be checked in practice.

2.1.2 Replication

In noisy experiments, replication is an important concept. It is necessary in order to reduce the variability inherent in microarray experiments. Generally, there are two types of replication: technical and biological. One form of technical replication is spot duplication. If space permits, cDNAs can be spotted in duplicate

on every array and the degree of conformity between duplicate spot intensities is a good indicator of the quality of the slide and hybridization. It is advisable, however, that duplicate spots be well spaced apart rather than spotted adjacently as this facilitates inspection of the degree of variability across the slide. Another type of technical replication is the array replicate. It is the replication of multiple arrays hybridized with RNA from the same sample (preparation). Due to the length and complexity of a microarray experiment, it is crucial to check that the results were not obtained by mere chance fluctuations, but rather arise from genuine underlying biological variation. Technical replication can be used to obtain an average measurement from each sample or to quantify systemic variation.

Biological replicates could be hybridizations performed using RNA from independent preparations from the same source, or preparations from biologically distinct sources, such as different organisms or different versions of a cell line. The latter type of biological replication is more popular since it encompasses greater variation in measurements. For instance, an experiment investigating drug treatment in mice is subject to the variation within the mice population, such as differences in immune system, sex, and age. The greater variability inherent in this form of replication contributes to a broader generalization of the experimental results.

In conclusion, a researcher should use biological replicates to validate generalizations of conclusions and technical replicates to estimate and eliminate the variability associated with the hybridization.

2.1.3 Pooling

Due to the instability of RNA, it can be difficult to extract sufficient material for hybridization, especially if the sample is to be spread over several replicates. Sometimes the RNA required for even a single array may be unachievable for small organisms. In such circumstances, the RNA from several samples could be pooled by biologists to make up the volume needed, but this practical constraint may alter the objectives of the investigation. After pooling the researcher is no longer able to make inferences about the individual samples, but only about the population from which they were drawn. This restriction may not be too important when the purpose of analyzing individual samples is to make inference on the population, which is typically the case.

When one wishes to characterize a population, pooling might reduce the overall costs of an experiment because arrays are often, though not always, more expensive than the generation of the samples. The cost of an experiment can be substantially reduced by measuring a number of pooled samples on a smaller number of arrays. Pooling multiple replicates will have the effect of decreasing the population variance and diminishing random fluctuations. However, the researchers should be aware of situations where it is not appropriate to pool samples. For example, when studying the effect of a drug on cancer patients, the gene expression in specific patients is of interest. In this case, hybridizations with individual samples should be carried out. On the other hand, in an investigation of two inbred homozygous ecotypes of Arabidopsis, differences between the individual plants are not of interest, so pooling may be justified.

2.1.4 Experimental designs

A single microarray experiment is just a comparison between two RNA samples collected under different treatments, both are applied to the same dual-channel array. The array can be considered as a blocking factor, similar to a plot of land in an agricultural field trial. Therefore, basic microarray experimental design is a block design with block size two. Since design can involve direct or indirect comparisons, there are usually more than one way to pair and label samples for cDNA microarrays.

2.1.4.1 Direct comparisons

Due to the parallel nature of dual-hybridization microarrays, the most efficient design to compare two samples is to directly compare them on the same array. By pairing samples, we can examine the relative abundance of the two samples, while accounting for variation in spot size that would otherwise contribute to the error.

Dye swap is a simple and effective design for the direct comparison of two samples. This design compares two samples by using two arrays instead of one. On array one, one sample is assigned to the red dye, and the other sample is assigned to the green dye. On array two, the dye assignments are reversed. See Figure 2.2 (a). In the vocabulary of experimental design, a dye-swap design is a complete block design, taking the form of a 2×2 Latin Square (Table 2.1). This simple design plan removes dye effect from the measurements by taking the mean log expression ratio on each probes for both dye-swaps. This arrangement can also be repeated by using an even number of arrays (e.g. four or six or more) to compare the same two biological samples. See a simple example in Figure 2.2

	Red dye (Cy5)	Green dye (Cy3)
Array 1	Sample 1	Sample 2
Array 2	Sample 2	Sample 1

Table 2.1: A Latin Square design to compare two samples directly.

(b). Repeated dye-swap experiments are used for reducing technical variation (although not very popular in practice). If independent biological samples are used, the experiment will account for both technical and biological variation.

If a microarray experiment involves more than two samples under different treatments, then not every sample can appear on every array and some form of incomplete block design should be considered instead. This brings with it a challenge of how to design the experiments (i.e. which samples should be co-hybridized on a single array) so that the efficiency and reliability of the microarray data can be improved and precise estimates of biologically important parameters can be obtained.

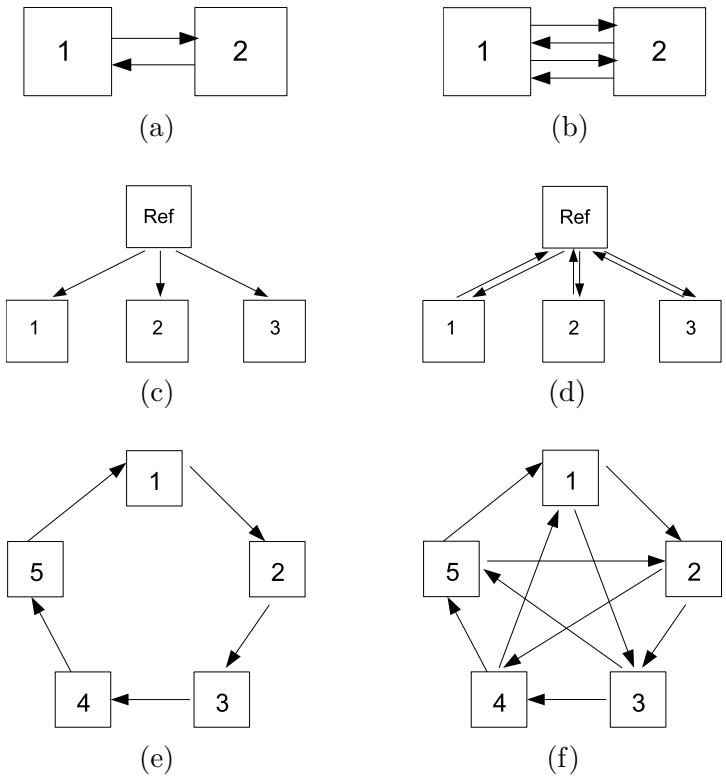


Figure 2.2: Diagrammatic representations of the designs of six microarray experiments. Each microarray array is represented by an arrow. The head of the arrow indicates that the sample was labeled with Cy5, while the tail represents a sample that was labeled with Cy3. (a) Direct comparison (dye-swap) between two samples; (b) A repeated dye-swap experiment between two samples with four arrays; (c) A reference design (indirect comparison) studies three samples; (d) A variation of the reference design (Figure 2.2 (c)) using a dye swap for each comparison; (e) A loop design with five treatments; (f) An interwoven loop design for five treatments and ten arrays.

2.1.4.2 Reference design

The reference design of Kerr and Churchill (2001*a*) affords a means of indirect comparison, and is commonly used for studying multiple treatments of a factor of interest. It is called a reference design because it uses an aliquot of a common reference RNA as one of the samples hybridized to each array (See a simple example in Figure 2.2 (c)). This is done so that the intensity of hybridization to a spot for a test sample is measured relative to the intensity of hybridization to the same spot on the same array for the reference sample (typically of no scientific interest).

The reference sample is usually labelled with one dye and acts as an intermediate and allows an indirect comparison between the samples of interest, all of which are labelled with the other dye. This means that treatment effects are completely confounded with dye effects. Consequently, the effects of interest, treatment × gene (TG) are completely confounded with dye × gene (DG) effect. If the dye × gene (DG) effect is significantly noticeable, then the microarray data from reference designs have to be validated before making conclusions. Alternatively a reverse-dye comparison could be incorporated in a biological replicate to account for the dye effect on specific genes (i.e. use two arrays in a dye-swap configuration, see an example in Figure 2.2 (d)). Another disadvantage is that making all comparisons to a reference sample can be inefficient, because half of the hybridization resources (e.g. arrays) are allocated to the reference sample, which is presumably of little or no interest.

In spite of its inefficiency, the reference designs are very popular among practitioners. There are several reasons. First of all, reference designs are very intuitive to understand: by being measured against the same reference the values across different arrays can be directly compared with one another. Secondly, it is also

very straightforward to use the same reference to control variation in each spot and there are only two path-steps connecting two samples in a reference design, so each comparison can be made with equal efficiency. Thirdly, as long as the amount of reference sample is not limiting, the reference design can be extended to handle a large number of treatment levels. From a practical perspective, every new sample in a reference experiment is handled in the same way. This reduces the possibility of laboratory error and increases the efficiency of sample handling in large projects. Finally, the reference design is robust to loss of arrays resulting from poor quality hybridization, although the loss of one array may entail the complete loss of information about one nonreference sample.

2.1.4.3 Loop design

The loop design is an alternative to the reference design (Kerr and Churchill, 2001*a*). Loop designs compare two treatments via a chain of other treatments without the need for a reference treatment. The nominal last treatment is connected with the nominal first treatment. A simple loop design with five treatments is shown in Figure 2.2 (e).

The loop design is more efficient than the reference design since the former can measure twice the number of replicates by using the same number of arrays as the latter. In simple loop designs, treatments are balanced with respect to dyes because each sample is labeled once with the red dye and once with green dye. This balance means that dye effects are unconfounded with treatment effects, so treatment $\times$ gene effects are unconfounded with dye $\times$ gene effects. Thus the effects of interest will not be biased by any strange behavior of genes with respect to dyes.

Along with these advantages, there are three problems with loop designs.

First of all, contrasting two treatments far apart in the loop involves modeling many indirect effects, corresponding to the arrays linking the two treatments of interest. This adds substantial variance to many of these contrasts (Khanin and Wit, 2004). Thus loop designs are not ideal for large numbers of treatment levels (Kerr and Churchill, 2001*a*). Secondly, loop designs are less robust against the presence of bad quality arrays: two or more bad arrays can break the loop apart and collapse the experiment. However, this problem can be solved by repeating the bad quality arrays. Finally, adding additional treatment levels to the loop design is not as easy as in the reference design.

2.1.4.4 Interwoven loop design

As alternatives to the reference design, loop designs have gained some popularity among practitioners. However, Kerr and Churchill (2001*a*) have pointed out that loop designs are optimal only for a relatively small number of treatments. Wit et al. (2005) identified a type of designs, interwoven loop designs, that seems to have good optimality properties.

The interwoven loop design, sometimes also called the replicated loop design, is an extension of the original loop design (Churchill, 2002). If the number of microarrays is a multiple k of the number of treatments p, then an interwoven loop design $I_p(1, j_2, ..., j_k)$ can be defined as an ordinary loop design (with k replicates) where each sample is also measured with respect to the samples that are $j_2, j_3, \ldots, j_k$ jumps further along the circle. An interwoven loop design example $I_5(1, 2)$ is shown in Figure 2.2 (f). When the number of treatments is quite large, interwoven loop designs have been demonstrated to have very nice properties: easy to implement, high efficiency, automatic dye balance (Wit et al., 2005).

2.1.4.5 Alternative designs

Besides the designs discussed above, it is possible to find other good designs. John and Mitchell (1977) suggested exhaustive search algorithms for finding the optimal design within particular classes of designs, but these have only limited practical applicability. John and Williams (1995) discussed the employment of simulated annealing for optimal row-column designs that could be directly applicable to dual channel microarray designs. Kerr and Churchill (2001*b*) used a computer program for graphs and taking into account other design properties such as balance, they searched exhaustively for non-isomorphic connected designs. However, this is only possible when the number of microarrays is small, typically less than 10.

Inspired by these works, Wit et al. (2005) applied a simple optimization strategy, also based on simulated annealing, to obtain optimal or near optimal microarray experimental designs in the sense of minimizing a criterion based on the variance of all the possible contrasts between treatments.

2.2 Optimal design with biological and technical replicates

Wit et al. (2005) applied an optimization strategy based on simulated annealing to search for near-optimal designs for any number of treatments and any number of arrays. However, the optimization strategy simply assumes that for each sample condition (treatment) the number of biological replicates available should exceed the number of arrays it involves. In other words, there should be enough independent biological samples for each treatment. Unfortunately, this is not always possible. For example, sometimes biological material may be very limited when one is conducting research on mammals (Byrne et al., 2005). In that case, one has to use technical replicates instead of biological replicates. Then we have the following problem: How to assign optimally the biological and technical replicates to the arrays?

In this section, we develop a modified optimization strategy using simulated annealing to find near optimal designs and near optimal allocations of biological and technical replicates.

2.2.1 A statistical model for microarray gene expression intensity

A cDNA microarray experiment contains information about the expression of thousands of genes. Each spot on the array measures two gene expression signals under two treatments associated with two dyes Cy3 and Cy5. Since the signals are essentially positive and typically behave multiplicatively, rather than additively, the logarithmic transformation can be applied to transform the optical intensity

of a spot associated with a particular gene from the multiplicative scale into an additive scale (Chen et al., 1997). Here, we model the gene expression intensity for each channel separately. This allows us to compute the variance of any contrast, and to determine the effect of biological and technical replication on the variance.

Consider the gene expression models for the two channels c_1 and c_2 of an array:

$$\log x_{c_1k_1r_1} = \theta_{c_1} + S + D + \epsilon_{c_1k_1} + \eta_{c_1k_1r_1}, \tag{2.1}$$

$$\log x_{c_2k_2r_2} = \theta_{c_2} + S + D + \epsilon_{c_2k_2} + \eta_{c_2k_2r_2}, \tag{2.2}$$

where x_{ckr} is the signal intensity, θ_c is proportional to the true gene expression under channel c, S is the nuisance effects such as spot effect and spatial effect, D is the dye effect. Note that we set the spot and spatial effect D be the same for the two channels for the reason that they are assumed to affect each of the channels similarly because the two channels has the same spot size and have the same position-dependent sources of variation. Dye effect D is also assumed to be the same for the two channels for the purpose of simplicity.

ϵ_{ck} is the biological variation for individual k under channel c and assumed to be normal distributed with mean zero and variance σ_b^2, η_{ckr} is the technical variation for the rth replicate of the individual k under channel c and assumed to be normal distributed with mean zero and variance σ_t^2. We make several further assumptions for biological variation and technical variation.

- ϵ and η are independent from each other no matter what the subscript.
- When $c_1 = c_2$ and $k_1 = k_2$, $\mathrm{Cov}(\epsilon_{c_1k_1}, \epsilon_{c_2k_2}) = \sigma_b^2$; otherwise, $\mathrm{Cov}(\epsilon_{c_1k_1}, \epsilon_{c_2k_2}) = 0$.
- When $c_1 = c_2$, $k_1 = k_2$ and $r_1 = r_2$, $\mathrm{Cov}(\eta_{c_1k_1r_1}, \eta_{c_2k_2r_2}) = \sigma_t^2$;

otherwise, $\mathrm{Cov}(\eta_{c_1k_1r_1}, \eta_{c_2k_2r_2}) = 0$.

The difference between the gene log expressions of the two treatments in one spot is equal to the log-ratio of the gene expressions. For a particular gene on an array, we can calculate the log-ratio of the gene expressions of the two treatments:

$$\begin{aligned} y_{c_1c_2k_1k_2r_1r_2} &= \log x_{c_1k_1r_1} - \log x_{c_2k_2r_2} \\ &= \theta_{c_1} - \theta_{c_2} + \epsilon_{c_1k_1} - \epsilon_{c_2k_2} + \eta_{c_1k_1r_1} - \eta_{c_2k_2r_2}, \end{aligned} \tag{2.3}$$

2.2.2 Parametrization and estimation

In a comparative microarray experiment across p treatments, the parameters of interest are: $\theta = (\theta_{c_1}, \theta_{c_2}, \ldots, \theta_{c_p})^T$, where θ_{c_i} is the average log gene expression for channel c_i. The log-ratio formulation of the microarray gene expression model is informative about the gene expression θ only up to an additive constant. In order to identify the parameter θ, we should impose some constraint such as setting the sum of θ to zero or setting the first element of θ to be zero, i.e. $\theta_{c_1} = 0$.

Instead of the vector of absolute expression θ, we can reparameterize the model with a vector of $\delta^* = \{\delta_{c_ic_j} | c_i > c_j\}$, where $\delta_{c_ic_j} = \theta_{c_i} - \theta_{c_j}$. This parametrization contains all the possible relative (differential) expressions, but it is over-parameterized. Therefore we use a canonical parametrization δ consisting of $p-1$ terms, $\delta = (\delta_{c_2c_1}, \ldots, \delta_{c_pc_1})^T$. Any other item in δ^* can be regarded as a linear combinations of δ (e.g. $\delta_{c_ic_j} = \delta_{c_ic_1} - \delta_{c_jc_1}$). Thus we can have the log-ratio

of the gene expressions of two treatments c_i and c_j of an array as follows,

$$\begin{aligned} y_{c_i c_j k_i k_j r_i r_j} &= \delta_{c_i c_j} + \epsilon_{c_i k_i} - \epsilon_{c_j k_j} + \eta_{c_1 k_1 r_1} - \eta_{c_2 k_2 r_2} \\ &= \delta_{c_i c_1} - \delta_{c_j c_1} + \epsilon_{c_i k_i} - \epsilon_{c_j k_j} + \eta_{c_1 k_1 r_1} - \eta_{c_2 k_2 r_2}. \end{aligned} \tag{2.4}$$

Assuming that the microarray experiment involves n arrays (i.e. $2n$ channels) and p treatments, we can write the log-ratio gene expression intensity according to the spirit of Equation (2.4) for all the arrays respectively and then we have a system of n equations which can be described in matrix notation as follows,

$$y = X\delta + \sum_{i=1}^{p} Z_i \epsilon_i + \eta, \tag{2.5}$$

where y is a $n \times 1$ vector of observations, X is a $n \times (p-1)$ design matrix, δ is the $(p-1) \times 1$ canonical parametrization, Z_i is a $n \times m_i$ random effect matrix for treatment i, ϵ_i is a $m_i \times 1$ vector of biological variation for treatment i and is assumed to be normally distributed with zero-mean and covariance matrix $\sigma_b^2 I_{m_i}$, m_i is the total number of independent biological replications available under treatment i, η is a $n \times 1$ vector of technical variation and is assumed to be normally distributed with zero-mean and covariance matrix $2\sigma_t^2 I_n$. Here I_{m_i} denotes the $m_i \times m_i$ identity matrix and I_n denotes the $n \times n$ identity matrix.

If we let $\varepsilon = \sum_{i=1}^{p} Z_i \epsilon_i + \eta$, then Equation (2.5) can be rewritten as

$$y = X\delta + \varepsilon, \tag{2.6}$$

where ε is a $n \times 1$ vector of normally distributed with zero-mean and covariance

matrix Σ, which can be calculated from Equation (2.5) as

$$\Sigma = \sum_{i=1}^{p} \sigma_b^2 Z_i Z_i^t + 2\sigma_t^2 I_n. \tag{2.7}$$

If the elements of ε are uncorrelated with each other, Σ is a multiple of identity matrix. If not, then Σ can be found according to the experiment design. An example of computing Σ is discussed in the next subsection.

By using maximum likelihood estimation (MLE), we get a generalized least squares (GLS) estimator,

$$\hat{\delta} = (X^t \Sigma^{-1} X)^{-1} X^t \Sigma^{-1} y, \tag{2.8}$$

which is the best linear unbiased estimator (BLUE) for δ, in the sense of having smallest sampling variability in the class of linear unbiased estimators, provided Σ is known (according to the Gauss-Markov Theorem). The variance of the estimator not only depends on the design matrix X but also on the covariance matrix Σ,

$$\mathrm{Var}(\hat{\delta}) = (X^t \Sigma^{-1} X)^{-1}. \tag{2.9}$$

2.2.3 An example: computation of Σ

The computation of Σ is just the computation of the covariance of the expressions of any two arrays. Here we summarize the rules of computation in the following (see the details in Appendix A):

1. When the two arrays have one common treatment and have the same technical replicate under that treatment, the covariance is $\sigma_b^2 + 2\sigma_t^2$ (if the arrays have the same type of dye attached on that treatment) or $-\sigma_b^2 + 2\sigma_t^2$ (if the

arrays have different types of dye attached on that treatment).

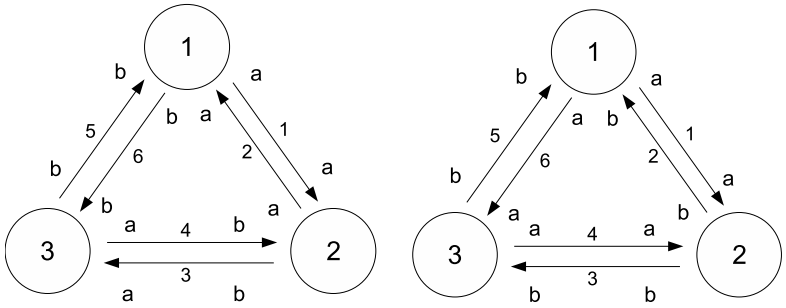
2. When the two arrays have two common treatments and have the same technical replicates under both of the treatments, the covariance is $2\sigma_b^2 + 2\sigma_t^2$ (if the arrays have the same types of dye attached on the treatments) or $-2\sigma_b^2 + 2\sigma_t^2$ (if the arrays have different types of dye attached on the treatments).

3. When the two arrays have two common treatments and have the same technical replicate under one treatment, the covariance is $\sigma_b^2 + 2\sigma_t^2$ (if the arrays have the same type of dye attached on that treatment) or $-\sigma_b^2 + 2\sigma_t^2$ (if the arrays have different type of dye attached on that treatment).

4. Under other situations, the covariance is zero.

Following the above rules, we can get the explicit form of covariance matrix Σ according to the experiment layout. As an example, let's consider an experiment with 3 treatments and 6 arrays. Each treatment has 2 biological (a and b) samples split into 2 technical replicates. Two design layouts of this experiment are shown in Figure 2.3, and two corresponding explicit forms of Σ are deduced in the following. For the dye-swap design in Figure 2.3 (a), we have

$$\Sigma = 2\sigma_b^2 \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} + 2\sigma_t^2 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$



a. A dye-swap design b. An alternative design

Figure 2.3: Two microarray experimental design with the same layout (3 treatments and 6 arrays) but different allocation of sample replicates.

For the alternative design in Figure 2.3 (b), we have

$$\Sigma = 2\sigma_b^2 \begin{bmatrix} 1 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 1 & 0 & -\frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & 1 & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 1 & 0 \\ \frac{1}{2} & 0 & 0 & -\frac{1}{2} & 0 & 1 \end{bmatrix} + 2\sigma_t^2 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Although both designs have the same design matrix, the different allocation of biological and technical replicates makes these designs have very different Σ. The non-zero entries off the diagonal in the covariance matrix reflects that some arrays (gene expressions) are correlated with each other due to having common biological replicates.

Note that when there are enough independent biological replicates available (e.g. 4 biological replicates for each treatment), we would have a very simple Σ:

$$\Sigma = 2\sigma_b^2 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \sigma_t^2 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

In this special situation, the variance of the estimate $\hat{\delta}$ can be simplified to be $\text{Var}(\hat{\delta}) = (X^t X)^{-1}\sigma^2$, where $\sigma^2 = 2\sigma_b^2 + 2\sigma_t^2$, which is exactly the case discussed in Wit et al. (2005).

In the same way, for any microarray experimental design, we can compute Σ and decompose it into two separate parts: a biological part and a technical part.

$$\Sigma = 2\sigma_b^2\Sigma_B + 2\sigma_t^2\Sigma_T,$$

where Σ_T is always an identity matrix and Σ_B can be easily computed given the details of the biological and technical replicates allocation. We can also rewrite Equation (2.10) as:

$$\Sigma = 2\sigma_t^2(\rho\Sigma_B + \mathbf{I}).$$

where $\rho = \frac{\sigma_b^2}{\sigma_t^2}$ and ρ is assumed to be a constant and we should know its value before planning the experimental design. In practice, ρ can not be known before experiment because it can only be estimated from the result of the experiment. As a way out, for each gene (spot) ρ could be estimated from previous experimental

data by using restricted maximum likelihood (REML) or maximum likelihood (ML), but these methods are very inaccurate with the small sample sizes often used in microarray studies. In recent approaches, using all the spots on the array has been suggested to improve estimation. For example, Smyth, Michaud and Scott (2005) use empirical Bayes estimation to improve the estimate of σ_t^2 and assume a single ρ value that can be computed from all the genes. Cui et al. (2005) use shrinkage estimation to improve the estimation of all the variance components and so on. In this section, we do not concern ourself with the estimation of ρ (σ_t^2 and σ_b^2) and assume ρ is already known.

2.2.4 Optimality criteria

Optimal design is a matter of applying the observations to the treatments in such a way that the parameters of interest are estimated most "optimally". For microarray experiments, there is a limited number of arrays available as well as a certain amount of RNA from several biological treatments of interest. The question then is which samples should we put on which arrays in order to maximize the precision of resulting parameter estimates?

The definition of precision depends on what optimality criterion is used. There are are several quite popular forms of design optimality, such as D-optimality, A-optimality and its related L-optimality (Wit et al., 2005). The covariance matrix of the parameter estimates plays a key role in all of these three forms of design optimality.

D-optimal design seeks to minimize the determinant of the covariance matrix of the parameter estimates. A-optimal design is the design for which the average variance of the parameter estimates is minimal. L-optimal design is a modified A-optimal design which minimizes the average variance of the estimates of several

linear functions of the parameters.

The appropriate criterion for comparing designs for a specific experiment should be closely related to the objectives of that experiment. If the aim is to acquire maximal precision of all differential gene expressions, it is better to use the canonical parametrization and choose L-optimality rather than A-optimality, because A-optimality depends on the particular parametrization that is chosen while L-optimality's linear functions can map the canonical parameters into all possible contrasts between the treatments. If the aim is to minimize the generalized variance of all differential gene expressions, D-optimality is a choice which does not depend on the parametrization of the model.

In the next section we use simulated annealing to search for optimal or near-optimal designs, which not only consider the optimality of the design matrix, but also take into account the allocation of independent biological and technical replicates.

2.2.5 Simulated annealing implementation for finding near-optimal designs

We denote the class of possible designs for n arrays, p treatments and $(s_1, \ldots, s_p)$ biological replicates with respect to parametrization β as $\chi(n, p, s, \beta)$, where $s = (s_1, \ldots, s_p)$ the number of biological replicates available for treatments $1, \ldots, p$.

One way to select the optimal design consists in using discrete optimization over the space of design matrices $\chi(n, p, s, \beta)$. Since the design space is large, exhaustive searches are infeasible even for only moderately large n and p. We follow the simulated annealing framework in Wit et al. (2005) to find near optimal designs for arbitrary n, p, X and Z.

The simulated annealing algorithm to maximize an objective function $f(x)$ works as follows. First of all, let (X, Z) be the current state, where X is the design matrix and Z is the random effect matrix. Secondly, propose a new candidate state (X', Z'), where X' is the new design matrix and Z' is the new random effect matrix, from some proposal distribution $q((X, Z) \to (X', Z'))$. Then, the candidate is accepted as the next state with probability:

$$\min\left\{1, \left(\frac{f(X', Z')}{f(X, Z)}\right)^{1/T_i} \frac{q((X', Z') \to (X, Z))}{q((X, Z) \to (X', Z'))}\right\}, \tag{2.10}$$

where T_i is the current temperature parameter that decreases with the iteration index i. If the proposal p satisfies $q(X', Z' \to X, Z) = q(X, Z \to X', Z')$ for all (X, Z) and (X', Z'), we have a simpler form of the acceptance probability:

$$\min\left\{1, \left(\frac{f(X', Z')}{f(X, Z)}\right)^{1/T_i}\right\}. \tag{2.11}$$

If the candidate is rejected the next state is set to be the current state. The simulated annealing algorithm is started at a relatively high temperature T_0, so that at the beginning virtually all candidates are accepted. As the temperature is gradually decreased to zero, it becomes increasingly more difficult to accept moves to states that decrease $f(X, Z)$. van Laarhoven and Aarts (1987) prove that under some conditions on the proposal distribution q (essentially irreducibility of the resulting Markov chain) and on the cooling schedule (T_i proportional to

$1/\log(i)$) the simulated annealing algorithm converges with probability 1 to a global optimum.

In this paper, we choose exponential cooling schedules, such as $T_i = T_0 c^i$ where c is a constant that is smaller than but close to 1. We use $T_0 = 10$ and $c \in [0.99, 1)$. The total number of iterations was set to achieve a preset low final temperature $T_{final} = 0.0001$. The last visited state (X, Z) is returned after the last iteration.

Our implementation is very similar to the one in the paper by Wit et al. (2005). One difference is that we search over a larger design space here, not only the fixed design matrix X, but also the random effect matrix Z (from which we can compute biological replicates allocation matrix Σ). The other difference is that we implement a new schedule of proposals to explore the complete design space $\chi(n, p, s, \beta)$. Given a design D (e.g. (X, Z)) at iteration t, we propose a combination of the following moves.

1. Update X and Z:

   (a) Single edge move: pick at random one comparison in design D, say a biological replicate a of treatment i and a biological replicate b of treatment j. Pick at random two treatments, say a biological replicate c of treatment k and a biological replicate d of treatment l and propose a new design D', where the comparison between (i, a) and (j, b) has been replaced by a comparison between (k, c) and (l, d). Note that the move is not symmetric: $q(old \rightarrow new) \propto \frac{1}{n_k} \times \frac{1}{n_l}$, $q(new \rightarrow old) \propto \frac{1}{n_i} \times \frac{1}{n_j}$, where n_c is the number of biological replicates for treatment c.

   (b) Single vertex move: pick at random one comparison in design D, say

a biological replicate a of treatment i and a biological replicate b of treatment j. Pick at random one of the two treatments, say i, and pick at random one of the treatments except i and j, say k which contains a biological replicate c. Propose a new design D', where the comparison between (i, a) and (j, b) has been replaced by a comparison between (i, a) and (k, c). Note that the move is not symmetric: $q(old \rightarrow new) \propto \frac{1}{n_k}$, $q(new \rightarrow old) \propto \frac{1}{n_j}$, where n_c is the number of biological replicates for treatment c.

(c) Balanced two-edge move: pick at random two non-overlapping comparisons in design D, say the first between a biological replicate a of treatment i and a biological replicate b of treatment j, and the second between a biological replicate c of treatment k and a biological replicate d of treatment l, where i, j, k and l are all distinct. Propose a new design D', where the comparison between (i, a) and (j, b) is changed to (i, a) and (l, d) and the comparison between (k, c) and (l, d) is changed to (k, c) and (j, b). This balancing move guarantees that all the treatments remain measured equally often in D' as they are in D. Note that the move is symmetric: $q(old \rightarrow new) \propto \frac{1}{n_l} \times \frac{1}{n_j}$, $q(new \rightarrow old) \propto \frac{1}{n_j} \times \frac{1}{n_l}$, where n_c is the number of biological replicates for treatment c.

2. Keep X fixed, update Z:

(a) Single replicate move: take a random comparison; select randomly one of the two treatments and replace this replicate by another available biological replicate.

(b) Balanced replicate move: Randomly pick a treatment i, then randomly

pick two comparisons that both involve the treatment i. Exchange the biological replicate for treatment i between the two comparisons.

It is easy to find that each of the above two moves has symmetric proposal probabilities.

Note that it is guaranteed that the whole design space can be visited (i.e. the resulting Markov chain is irreducible), by simply using move 1(a). The reason for proposing the other moves is to improve the efficiency of finding the optimum.

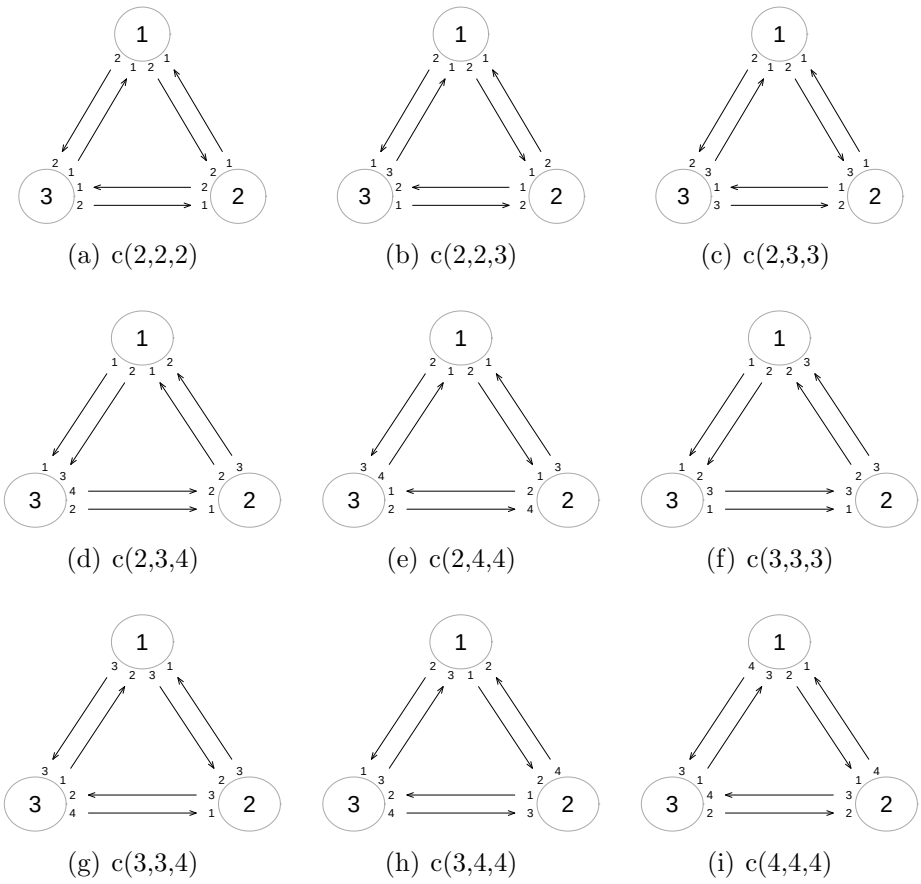
One can start from any arbitrary state, but starting at a good initial design can clearly save a lot of computational time. At each iteration one of the five moves that are described above is selected, with respective probabilities $p1$, $p2$, $p3$, $p4$ and $p5 = 1 - p1 - p2 - p3 - p4$. In our experience, using $p1 = 0.15$, $p2 = 0.4$, $p3 = 0.2$, $p4 = 0.15$ seems to work reasonably well.

2.2.6 Results

For each gene, the design matrix X and the random effect matrix Z are exactly the same and consequently the covariance structure among the parameters is proportional to $(X^t\Sigma^{-1}X)^{-1}$. Therefore, although we consider optimality for one gene at a time, the same design is simultaneously optimal for all genes.

2.2.6.1 Example one

Consider a microarray experiment for 3 treatments and 6 arrays. If we assume that each treatment can have 2 or 3 or 4 independent biological replicates, then there are 9 different scenarios in total. By using the simulated annealing algorithm we have developed in the last section, we are able to find the (possibly near) L-optimal design for each of the scenarios. The results are represented in Figure



(a) c(2,2,2) (b) c(2,2,3) (c) c(2,3,3)

(d) c(2,3,4) (e) c(2,4,4) (f) c(3,3,3)

(g) c(3,3,4) (h) c(3,4,4) (i) c(4,4,4)

Figure 2.4: The L-optimal designs of microarray experiment for 3 treatments and 6 arrays with respect to different combinations of numbers of independent biological replicates for each treatment. In the caption of subfigure, the notation $c(x, y, z)$ is used to indicate that treatment 1, 2 and 3 has x, y, z independent biological replicates respectively.

2.4. From the layouts, we see that each treatment uses as many biological replicates as possible: for treatments with enough biological replicates available (e.g. the treatments with 4 biological replicates), an independent biological replicate is used for each array it is involved with; for treatments with a limited number of biological replicates (i.e. less than the number of arrays on which they hybridize, like the treatments with only 2 biological replicates), the optimal design has to use technical replicates, i.e. repeats of certain biological replicates.

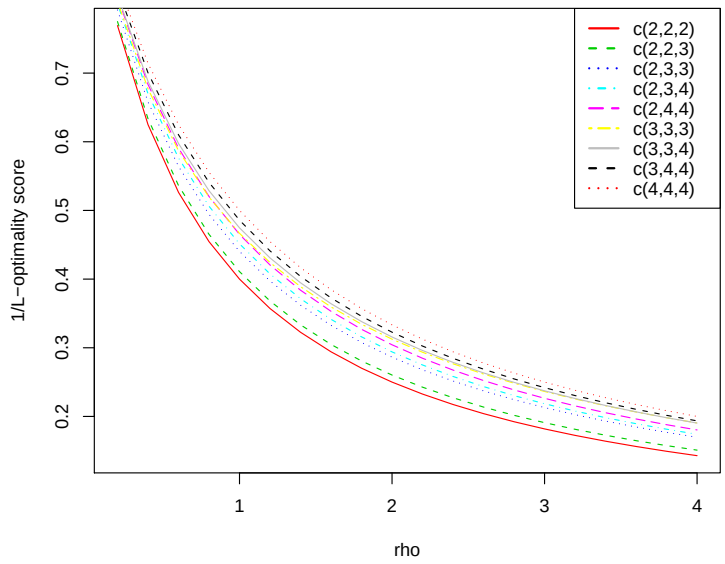


Figure 2.5: Comparisons of the reciprocals of L-optimality scores for the 9 designs shown in Figure 2.4 across the range of ρ from 0 to 4. Larger value of reciprocal of L-optimality score means higher L-optimality efficiency. Here, the design c(4,4,4) and c(2,2,2) has the highest and lowest L-optimality value respectively.

For example, in Figure 2.4 (e), 4 samples are needed for treatment 1, but only 2 biological replicates are available and therefore it has to use an extra 2 technical replicates, one technical replicate for each biological replicate.

The comparison of the reciprocals of L-optimality scores for the 9 scenarios across a range of ρ (i.e. the ratio of variance of biological variation to variance of technical variation) from 0 to 4 is shown in Figure 2.5. From these results, we find that the designs with more independent biological replicates would have high L-optimality efficiency. If one design has more biological replicates than the other design for all the treatments, then the former is definitely more efficient than the latter. For example, the design $c(4, 4, 4)$ has the highest L-optimality

score, i.e. highest L-optimality efficiency, while the design $c(2, 2, 2)$ has the lowest L-optimality score, i.e. lowest L-optimality efficiency, where $c(x, y, z)$ indicates the experimental scenario that treatment 1, 2 and 3 has x, y and z independent biological replicates respectively. On the other hand, if a design only has more biological replicates than the other design for some of the treatments, then it may be difficult to judge which one is more efficient. For example, out of the three treatments, the design $c(3, 3, 3)$ only has only one treatment with more biological replicates than the design $c(2, 4, 4)$, but the former is still slightly more L-optimality efficient than the latter.

2.2.6.2 Example two

Now we consider a bigger scenario that we have a microarray experiment for 5 treatments (conditions) and 15 arrays, each of the treatments has 6 biological replicates except the first one which has only two biological replicates, and ρ is assumed to be in the range of $[0.5, 2]$. By using the simulated annealing algorithm, we are able to find the L-optimal (or near L-optimal) design which is shown in the Figure 2.6. From the layout, we see that each of the treatment uses as many independent biological replicates as possible. For treatments with enough independent biological replicates, like condition 2, 3, 4 and 5, they use an independent biological replicate for each array they involve. For treatments with limited number of biological replicates available and less than the number of arrays on which they should hybridize, like treatment 1, they have to use technical replicate which is the copy of the corresponding independent biological replicate. For the treatment 1 in Figure 2.6, it needs six samples but only two biological replicates are available therefore four extra technical replicates are used (two technical replicates for each of the two biological replicates).

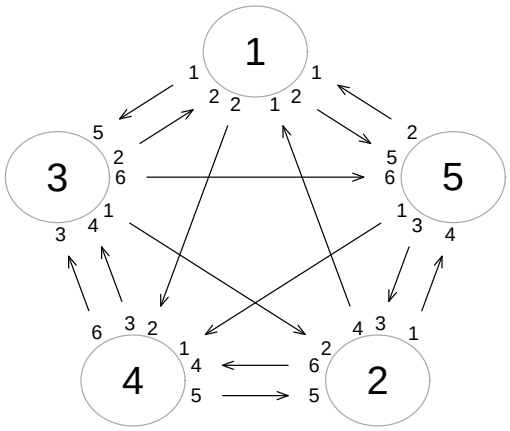


Figure 2.6: A cDNA microarray L-optimal design with 5 treatments and 15 arrays. The first treatment has only two independent biological replicates available while the rest of treatments have six independent biological replicates.

2.2.6.3 Are dye-swap designs optimal?

In a typical dye-swap experiment like in Figure 2.3 (a), each hybridization is done twice with the dye assignments reversed in the second hybridization using technical replicates, i.e. the same biological replicates of the first hybridization. Since it is useful for reducing systematic differences in the red and green intensities, the dye-swap design has been quite popular among practitioners. The alternative design in Figure 2.3 (b) is not a dye-swap design although it has the same design matrix as Figure 2.3 (a). The only difference between the two designs is that the dye-swap design uses technical replicates in the second hybridization while the alternative design uses independent biological replicates in the second hybridization.

One way to compare two designs is to calculate the relative efficiency, which is defined as the ratio of their optimality score (which depends on the optimality

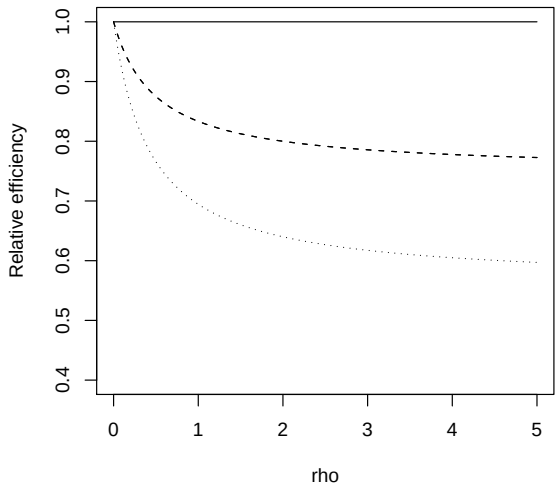


Figure 2.7: Comparisons across the dye-swap design and the alternative design under L-optimality and D-optimality criteria. When the ratio of biological variance to technical variance, ρ, varies from 0 to 5, the relative efficiencies are calculated as the ratio of the scores under the dye-swap design and the alternative design. The dashed curve represents the case of L-optimality and the dotted curve represents the case of D-optimality.

criterion one uses). Here we choose L-optimality and D-optimality as the criteria. As the score also depends on the value of ρ to some extent, the relative efficiency of these two designs with different ρ value, from 0 to 5, is computed and displayed in Figure 2.7. It shows that the alternative design of Figure 2.3 (b) is not only more L-efficient but also more D-efficient than the dye-swap design of Figure 2.3 (a) (e.g. the relative efficiency is smaller than 1) except when ρ is zero which is not possible in practice. With the increase of ρ value, the efficiency of the alternative design with respect to the dye-swap design increases steadily (e.g. lower relative efficiency).

2.3 Optimal design for factorial experiment

So far, this chapter has only considered single factor experiments (typical factors being time, genotype, tissue type or treatment). Microarray experiments investigating two or more factors require a more complex design, like factorial experiment design which can be used to study the expression profiles resulting from the combined effect of multiple factors.

Since in recent years more and more biologists have begun to consider multi-factorial microarray experimental set-ups to identify differentially expressed genes, e.g. Caetano et al. (2004), the problem of how to find efficient factorial designs has received some attention. For example, Glonek and Solomon (2004) used A-optimality to find optimal designs for factorial experiments with a small number of factors. Their approach enables the selection of an efficient design subject to the information available on the parameters of interest to biologists.

One way to design a factorial experiment is to consider all factor combinations as treatments and do a one-way optimal design. It corresponds to a full factorial model which assumes that all interactions are equally important. However, this is not necessary and higher-order interactions could be down-weighted. Therefore we suggest that using the Q criterion rather than A-optimality, L-optimality or D-optimality would be a proper choice for the optimal design of factorial experiments. In the next section we first introduce a gene expression model in a multi-factorial way before introducing the Q criterion.

2.3.1 Statistical gene expression models for $p \times q$ factorial experiment

In Section 2.2.1, we have introduced Equation (2.1) and (2.2) as general statistical microarray gene expression models for the two channels of an array. If we ignore dye effects, spatial effects and the difference between biological error and technical error, then for one channel the gene expression x under treatment c after taking logarithms can be reduced to

$$\log x_c = \theta_c + \varepsilon, \tag{2.12}$$

where θ_c is the true gene expression under treatment c, ε is the total error.

If we consider a $p \times q$ factorial experiment and think of all factor combinations as treatments, then the factorial experiment has $p \times q$ different treatments in total. According to Equation (2.12), we can denote by θ_i the gene expression of a gene for a certain treatment i, where the parameters of interest are $\theta = (\theta_1, \theta_2, \ldots, \theta_{p\times q})^t$, which correspond to all the treatments respectively. Instead of a vector of absolute gene expression values θ, we use the canonical parametrization δ consisting of $p \times q - 1$ terms, $\delta = (\delta_{21}, \delta_{31}, \ldots, \delta_{p\times q,1})^t$, where $\delta_{i1} = \theta_i - \theta_1$ for $i = 2, \ldots, p \times q$. Computationally speaking, using the canonical parametrization δ is equivalent to using θ with the constraint $\theta_1 \equiv 0$.

On the other hand, if θ_c is considered to have some factorial structure, e.g. $c = (p, q)$ treatment specified by level p from the first factor and level q from the second factor, then we have an extended model of gene expression for the $p \times q$ factorial experiment:

$$\log x_{st} = \mu + \alpha_s + \beta_t + (\alpha\beta)_{st} + \varepsilon, \quad s = 1, \ldots, p, \quad t = 1, \ldots, q, \tag{2.13}$$

where the intercept μ is the baseline intensity with each factor at its lower level, α is a main effect parameter for the difference in intensities among the p levels of the first factor, β is a main effect parameter for the difference in intensities among the q levels of the second factor, $\alpha\beta$ is the interaction of the two factors. We impose the sum-to-zero constraints on the parameters in each of these factorial models that:

$$\sum_{i=1}^{p} \alpha_i = 0, \quad \sum_{j=1}^{q} \beta_j = 0, \quad \sum_{i=1}^{p} (\alpha\beta)_{ij} = 0, \quad \sum_{j=1}^{q} (\alpha\beta)_{ij} = 0.$$

Therefore there are $p-1$ parameters for α, $q-1$ parameters for β and $pq-p-q+1$ parameters for $(\alpha\beta)$ in this model. Further, because we study differential expressions, μ is set to be zero. As a result, the new parametrization of this extended model is different from the canonical parametrization δ and consists of $p \times q - 1$ terms in total. We denote it by $\varphi = (\alpha, \beta, (\alpha\beta))^t$, where α denotes $\{\alpha_i\}_{i=2,\ldots,p}$, β denotes $\{\beta_j\}_{j=2,\ldots,q}$ and $\alpha\beta$ denotes $\{(\alpha\beta)_{ij}\}_{i=2,\ldots,p;j=2,\ldots,q}$.

2.3.2 Q-criterion

Tsai et al. (2000) suggested a new multi-factorial design optimality criterion called the Q-criterion. It is an approximation to the mean A efficiency, ignoring the intercept, over all models that will be used for fitting. The models may include main effects as well as interactions.

Assume that the maximal model of interest is $E(y) = X\gamma$, where y is an $N \times 1$ vector of observations, X is an $N \times (v+1)$ design matrix and $\gamma = (\gamma_0, \ldots, \gamma_v)^t$ is a $(v+1) \times 1$ vector of parameters for a particular factorial model. The information matrix for this model is X^tX and the covariance matrix of the least squares estimator of γ divided by σ^2 is $(X^tX)^{-1}$. The elements c_{ii} for $i = 1, \ldots, v$ on

the diagonal of the matrix $(X^tX)^{-1}$ are approximated (for details see Tsai et al. (2000)) as

$$c_{ii}(X) = \sum_{j=0}^{v} \frac{1}{a_{ii}} \frac{a_{ij}^2}{a_{ii}a_{jj}},$$

where a_{ij} for $i, j = 0, \ldots, v$ are the elements of X^tX. Then the Q-criterion for design matrix X is defined as the average of the sum of weighted c_{ii} (i.e. the criterion of A-optimality) over n_0 models:

$$Q(X) = \frac{1}{n_0} \sum_{i=1}^{v} \sum_{j=0}^{v} \frac{1}{a_{ii}} \frac{a_{ij}^2}{a_{ii}a_{jj}} w_{ij}, \tag{2.14}$$

where n_0 is the total number of factorial submodels of X.

A weight matrix W is calculated such that its element w_{ij} (for $i, j = 0, \ldots, v$) stores the number of models that contains both effect terms i and j, i.e. the number of factorial submodels of X, both of which include γ_i and γ_j:

$$w_{ij} = \sum_{s=1}^{n_0} M_s(i, j)$$

and

$$M_s(i,j) \quad = \quad \begin{cases} 1 & \text{if effects } i \text{ and } j \text{ are both included in model } M_s, \\ 0 & \text{otherwise.} \end{cases}$$

where M_s is a model for fitting and contains a subset of effects of the maximal model.

Generally, the Q-criterion depends on the parametrization, the information

matrix and the weight matrix. It is a weighted average of approximated A-efficiency, where the weights enhance main effects and lower-order interactions.

Due to the absence of the intercept γ_0 in the microarray log-expression ratio model, we need a slight modification of the Q-criterion:

$$Q(X) = \frac{1}{n_0} \sum_{i=1}^{v} \sum_{j=1}^{v} \frac{1}{a_{ii}} \frac{a_{ij}^2}{a_{ii} a_{jj}} w_{ij} \tag{2.15}$$

In case one chooses to use the old Q-criterion, then that corresponds to including an explicit dye-effect in the expression model.

2.3.3 Simulated annealing implementation for finding near Q-optimality design

In this section we discuss how to find Q-optimality design by using the simulated annealing algorithm.

First of all we give a definition of Q-optimality. Consider designs for n samples and m treatments with respect to parametrization ψ with design matrices from the class $\chi(n, m, \psi)$. A design is a Q-optimal design if its design matrix $X_{\text{Q-opt}}(\psi)$ minimize the Q-criterion value:

$$X_{\text{Q-opt}}(\psi) = \text{argmin}_{X \in \chi(n,m,\psi)} Q(X). \tag{2.16}$$

Then we are allowed to follow the framework of the simulated annealing algorithm proposed in the Section 3 of Wit et al. (2005) to find a near Q-optimality design. However, the application of the simulated annealing is not straightforward. The problem is that the simulated annealing algorithm we use depends on

the canonical parametrization δ rather than the new parametrization ψ. Therefore we have to find a transformation matrix T, such that

$$\psi = T^{-1}\delta, \tag{2.17}$$

because then

$$E(y) = X\delta = XT\psi$$

and, so,

$$V(\hat{\psi}) \propto T^{-1}(X^tX)^{-1}(T^{-1})^t$$

and, therefore, the information matrix for the model is

$$I(\hat{\psi}) = T^t(X^tX)T, \tag{2.18}$$

This is useful when we implement the simulated annealing algorithm, because T will always be fixed for every design matrix X and so T has to be calculated only once.

As T can be easily found from standard software (e.g. model.matrix in R), we can extend the simulated annealing for finding near-optimal design by using Q-criterion.

2.3.4 An example: 2×4 factorial microarray experiment

We now demonstrate with an example how to find Q-optimal or near Q-optimal designs for a two-factor microarrray experiment. The discussion will be given in terms of a single gene and it is intended that the same parametrization be applied separately for every gene on a slide.

Consider a two-factor microarrary experiment: two lines of pigs whose ovary material is studied 2, 3, 4 and 6 days after inducing luteal regression. It is anticipated that measuring changes over time would distinguish genes involved in promoting or blocking differentiation. We are interested in genes differentially expressed between the two lines (the first factor with 2 levels denoted as A and B) and 4 different time points (the second factor with 4 levels denoted as 1, 2, 3 and 4), i.e. in the main effects. If we think of all factor combinations as treatments, the 2×4 factorial experiment has 8 different treatments, which are denoted as $A1$, $A2$, $A3$, $A4$, $B1$, $B2$, $B3$ and $B4$, respectively.

According to the discussions in the Section 2.3.1, if the gene expression observations from the experiment are modelled as in Equation (2.12), then we have the corresponding canonical parametrization $\delta = (\delta_{21}, \ldots, \delta_{81})$, where δ_{i1} is the difference between the gene expression of the ith treatment and the first treatment, for $i = 2, \ldots, 8$; if the gene expressions are modelled as in Equation (2.13), then we have the new parametrization $\psi = (\alpha_2, \beta_2, \beta_3, \beta_4, (\alpha\beta)_{22}, (\alpha\beta)_{23}, (\alpha\beta)_{24})^t$, where everything is measured relative to line 1 at time point 1: α_i is the average difference between line 1 and line i over all the time points, β_j is the average difference between time point j and time point 1 across both lines, $(\alpha\beta)_{ij}$ is the difference between the actual difference between line 1 and line i at time point j and the difference between those lines at time point 1, for $i = 2$ and $j = 1, \ldots, 4$. Note that in this case α_i is a main effect parameter for the difference in intensities between two lines, β_j is a main effect parameter for different days after inducing luteal regression, $(\alpha\beta)_{ij}$ is an interaction.

The relationship between the new parametrization ψ and canonical parametrization δ is summarized in matrix notation:

$$\psi = C\delta$$

where

$$C = \begin{bmatrix} -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & 0 & -\frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{bmatrix}.$$

Note that C corresponds to T^{-1} in Equation (2.17) and in this situation the information matrix of the model using the new parametrization is $(C^{-1})^t(X^tX)C^{-1}$ rather than X^tX.

In this example, the total number of factorial submodels $n_0 = 5$, and the corresponding weight matrix W is shown in Table 2.2. According to Equation (2.15), only the elements $\{w_{ij}\}_{i,j=1,\ldots,7}$ are useful for computing Q-criterion values in our application.

w_{ij}	0	1	2	3	4	5	6	7
0	5	3	3	3	3	1	1	1
1	3	3	2	2	2	1	1	1
2	3	2	3	3	3	1	1	1
3	3	2	3	3	3	1	1	1
4	3	2	3	3	3	1	1	1
5	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1

Table 2.2: The elements of weight matrix $W = \{w_{ij}\}_{i,j=0,\ldots,7}$ are computed for 2×4 factorial design. Note that i and j denote the index of two effects in the maximal model respectively, and $w_{ij} = w_{ji}$, for $i \neq j$.

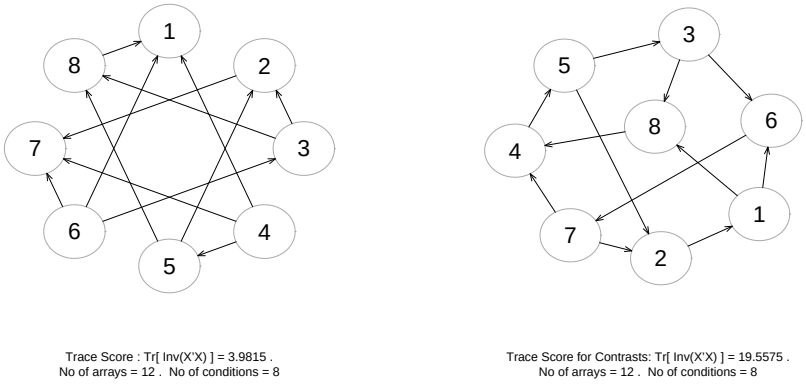
Assuming this 2×4 factorial experiment involves 12 arrays, we can search for the Q-optimal and L-optimal designs. The results are given in Figure 2.8 (a) and Figure 2.8 (b) respectively. Obviously, the Q-optimal design does not correspond to the L-optimal design.

The detail of the Q-optimal design layout for this 2×4 factorial experiment is represented in Figure 2.9 (a). It is interesting to see the difference between our Q-optimal design and a more "intuitive" design (Figure 2.9 (b)).

2.3.5 Conclusion

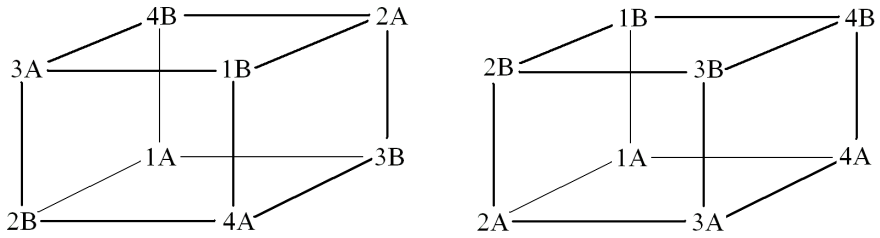
In this section, we introduced the Q-optimality criterion for optimal factorial design of cDNA microarray experiment. We discussed step by step how to incorporate this new optimality criterion into the simulated annealing framework. One simple example was used to show that the optimal finding by Q-optimality is different from that by L-optimality criterion. It should be pointed out that this Q-optimality criterion is applicable only when we feel interested in the main effects and lower-order interactions of the experiment. If high-order interactions

are our interest or all the effects are equally important, then alternative criteria or methods could be considered to find optimal designs.



a. Q-optimal design b. L-optimal design

Figure 2.8: Different design criterions make different optimal designs.



a. Q-optimal design b. Non Q-optimal design

Figure 2.9: Q-optimal and non Q-optimal design.

2.4 Optimal pooling strategy

Since microarray experiments are quite expensive but funding for biological research projects is usually limited, it is necessary for us to study how to pool and replicate RNA samples to achieve minimum variance using the minimal amount of resources. We call this approach optimal pooling. The choice for pooling is a trade-off between the cost of sampling RNA and the cost of a microarray. Note that only when the RNA sample is much cheaper than the array, then pooling a lot of RNA samples on an array is a good choice.

2.4.1 Methods

In this subsection we suggest an approach for optimal pooling: we try to find out which pooling scheme to use to minimize the variance of expression given a fixed budget.

If each RNA pool contains only one biological sample (i.e. replicate) from each subject then the observed log expression of a particular gene in pool i for the jth technical replication of that pool to an array is:

$$x_{ij} = \theta + \epsilon_i + \eta_{ij} \tag{2.19}$$

where θ is the true gene expression, ϵ is the biological variation (i.e. between-pool variation) among subjects and η is the technical variation (i.e. within-pool variation). It is assumed that both biological and technical variation variations are independent from each other and they are normally distributed respectively, that is, $\epsilon_i \sim N(0, \sigma_\epsilon^2)$ and $\eta_{ij} \sim N(0, \sigma_\eta^2)$, where $i = 1, ..., n_a$, $j = 1$, n_a is the number of pools in the experiment.

If each RNA pool contains several independent biological samples then we have a new expression model x_{ij}^p as follows,

$$x_{ij}^p = \theta + \epsilon_i^p + \eta_{ij}, \tag{2.20}$$

where $\epsilon_i^p \sim N(0, \sigma_\epsilon^2/s_i)$ and s_i is the number of samples in pool i. The idea behind this is that if a biological sample of a particular RNA under a treatment is expressed with a standard deviation of σ, then by mixing an independent collection of n RNA samples, the observed biological variation reduces to only $\sigma/\sqrt{n}$. In other words, the effect of the biological variation depends on the number of samples in the pool. The more samples in a pool, the less distinguishable the pools become.

The total number of samples in the whole experiment is $t_s = \sum_{i=1}^{n_a} s_i$. The estimation of the expression level θ is $\overline{x}$, the mean of x_{ij}^p. If we assume that each pool contains the same number of samples, n_s, the variance of $\overline{x}$ is,

$$\begin{aligned} V(\overline{x}) &= V\left(\frac{1}{n_a}\sum_{i,j}[\theta + \epsilon_i^p + \eta_{ij}]\right) \\ &= \frac{1}{n_a^2}\left[\sum_{i=1}^{n_a} V(\epsilon_i^p) + \sum_{i,j} V(\eta_{ij})\right] \\ &= \frac{\sum_{i=1}^{n_a} 1/s_i}{n_a^2}\sigma_\epsilon^2 + \frac{1}{n_a}\sigma_\eta^2 \\ &= \frac{\sigma_\epsilon^2}{n_a n_s} + \frac{\sigma_\eta^2}{n_a}. \end{aligned} \tag{2.21}$$

It is obvious that increasing both the number of arrays n_a as well as increasing the number of samples in each pool n_s will reduce the overall variance of estimation of expression. However increasing both will increase the cost. We assume that there are two types of cost associated with the number of samples in each

pool and the number of arrays used in the experiment. Let C_s be the cost of a single RNA extraction and preprocessing and C_a is the cost of a single microarray including cost of reverse transcription and label. Note that for dual-channel microarray, if the reference design is chosen, C_a is the cost of microarray plus both dyes; if the loop design (both channels contains sample of interest) is used, then C_a is half of the value.

Therefore the optimization problem can be formulated as minimizing the variance under the constraint of fixing the total cost at a certain level $B = t_s C_s + n_a C_a$.

We use Euler-Lagrange optimization to minimize the variance under the constraint of keeping the budget to a preset level. We find the minimum of the objective function f,

$$f(n_s, n_a, \lambda) = \frac{\sigma_\epsilon^2}{n_s n_a} + \frac{\sigma_\eta^2}{n_a} + \lambda(n_a C_a + n_a n_s C_s - B), \tag{2.22}$$

where λ is the Lagrange multiplier, by setting the following first derivatives to zero:

$$\frac{\partial f(n_s, n_a, \lambda)}{\partial n_s} = \lambda C_s n_a - \frac{\sigma_\epsilon^2}{n_a} \frac{1}{n_s^2},$$

$$\frac{\partial f(n_s, n_a, \lambda)}{\partial n_a} = \lambda C_s n_s + \lambda C_a - \frac{\sigma_\epsilon^2}{n_s} \frac{1}{n_a^2} - \frac{\sigma_\eta^2}{n_a^2},$$

$$\frac{\partial f(n_s, n_a, \lambda)}{\partial \lambda} = C_a n_a + C_s n_a n_s - B.$$

The system of equations is solved and the minimum can be found at

$$n_s = \sqrt{\frac{C_a}{C_s}}\frac{\sigma_\epsilon}{\sigma_\eta} \quad \text{and} \quad n_a = \frac{B}{C_a + \sqrt{C_a C_s}\sigma_\epsilon/\sigma_\eta}. \tag{2.23}$$

The result shows that the solution depends on the array cost C_a, sample cost C_s, the prescribed budget B and the ratio of biological variation and technical variation $\sigma_\epsilon/\sigma_\eta$. Practically, the values of C_a, C_s and B can be decided by biologists according to the specifics of theirs own experiment. The value of σ_ϵ should be truly fixed, depending only on the particular gene, the value of σ_η changes from lab to lab and from platform to platform. But both of σ_ϵ and σ_η are unknown to us. One way to solve this problem is to estimate them by using REML. In the following we would like to follow a simple method proposed by Wit and McClure (2004) to deduce reasonable values for them.

Churchill (2002) finds that the correlation between two arrays hybridized with the same RNA is approximately 0.70, while this correlation for arrays with RNA from different biological replicates is just over 0.30. If θ is considered as the expression of a randomly selected gene,

$$\text{Cor}(\theta + \epsilon_1 + \eta_{11}, \theta + \epsilon_1 + \eta_{12}) \approx 0.7$$

$$\text{Cor}(\theta + \epsilon_1 + \eta_{11}, \theta + \epsilon_2 + \eta_{21}) \approx 0.3$$

where $\theta+\epsilon_i$ is the actual expression of a particular sample i and η_{ij} is the technical error associated with the jth technical replicate of sample i. This results in the following:

$$\frac{\sigma_a^2 + \sigma_\epsilon^2}{\sigma_a^2 + \sigma_\epsilon^2 + \sigma_\eta^2} \approx 0.7$$

$$\frac{\sigma_a^2}{\sigma_a^2 + \sigma_\epsilon^2 + \sigma_\eta^2} \approx 0.4$$

where σ_a is the variation on a single microarray. Wit and McClure (2004) observes that σ_a^2 varies between 0.6 and 1.0 for the log expression values in several full genome arrays, therefore here we assume it is 0.8. These figures and the correlations allow us be able to estimate the values for σ_ϵ and σ_η:

$$\sigma_\epsilon = 0.9 \quad \text{and} \quad \sigma_\eta = 0.7.$$

Note that the above result of variances can be just used as approximate and candidate values for computing n_s and n_a. Since different genes have different σ_ϵ and different experiments result in different σ_η, consequently the variance ratio $\sigma_\epsilon/\sigma_\eta$ might vary to some extent.

The optimization problem can also be formulated as minimizing the total cost $B = t_s C_s + n_a C_a$ under the constraint of fixing the variance at a certain level $v(\bar{x}) = \sigma_0^2$. In the same way, Euler-Lagrange optimization can be used to get the expression of n_s and n_a: see Wit and McClure (2004) for the results.

The variability constraint $v(\bar{x}) = \sigma_0^2$ is difficult to interpret. However, it can be reformulated in terms of detectable fold-changes, type I error and type II error (Wernisch, 2002).

If we assume that the log expressions x_{ij} of the same gene under two different treatments have the same error model as Equation (2.19), then the differential expression is a normal distribution with mean $\log f_0$ with variance $2\sigma_0^2$, where f_0 is the fold-change.

The probability p that a gene with no differential expression (i.e. fold-change

zero) has a sample differential expression at least f_0 fold-change is:

$$p = 1 - \Phi\left(\frac{\log f_0}{\sqrt{2\sigma_0^2}}\right) \tag{2.24}$$

where Φ is the standard normal cumulative distribution. In a same way, the probability q that a gene with differential expression f fold-change has a sample differential expression smaller than f_0 fold-change is:

$$q = 1 - \Phi\left(\frac{\log f - \log f_0}{\sqrt{2\sigma_0^2}}\right) \tag{2.25}$$

If we let $p = \alpha$ and $q = \beta$, after solving of Equation (2.24) and (2.25) we can obtain:

$$\sigma_0^2 = \frac{1}{2}\left[\frac{\log f}{\Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta)}\right]^2 \tag{2.26}$$

where $\log f$ is the target fold-change which should be detectable at significance level of α for the probability of a type I error while the probability of making a type II error is controlled at level β (i.e. at power of $1 - \beta$). Therefore, we can use the above expression for σ_0^2 in the optimum result of n_s and n_a.

2.4.2 Example

As an example, we assume that total budget for microarray experiment is 8,000 British pounds, the cost of one microarray is 700 British pounds and the cost of one subject is 100 British pounds. If the value of σ_ϵ and σ_η are 0.9 and 0.7 respectively, then the optimal number of samples in a pool is 3 and the corresponding number of arrays is 8. We also explore the relationships among the variance ratio, the number of samples in pool and the variance of gene expression

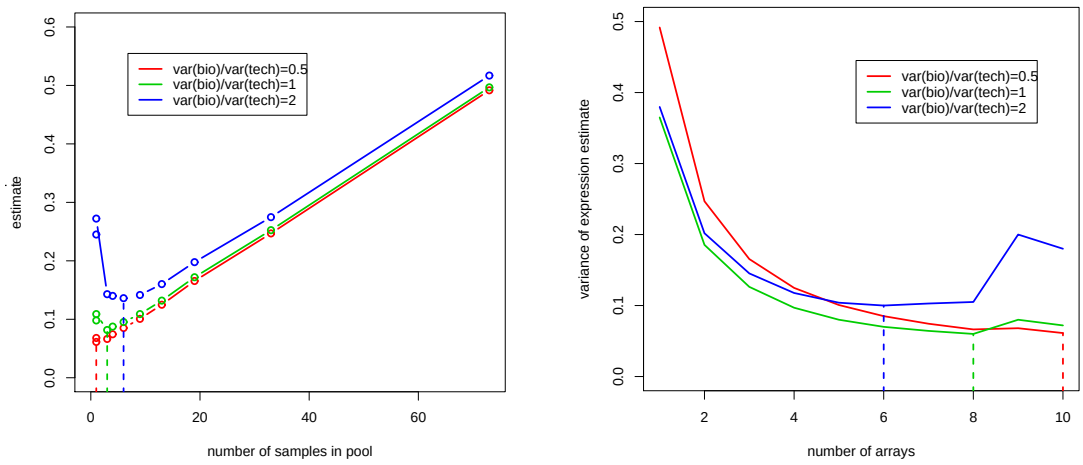


Figure 2.10: An example of optimal pooling by minimizing the estimation variance $V(\bar{x}) = \frac{\sigma_\epsilon^2}{n_s n_a} + \frac{\sigma_\eta^2}{n_a}$, subject to not overrunning one's budget $B = n_s n_a C_s + n_a C_a$. Here, $B = 8000$, $C_a = 700$, $C_s = 100$, $\sigma_\eta^2 = 0.7$. The yellow, green and red curves represent the relationship between gene expression variance and number of samples in pool when variance ratio is set to be 0.5, 1 and 2 respectively.

value when the variance ratio $(\sigma_\epsilon/\sigma_\eta)$ is not fixed. Figure 2.10 shows that when the variance ratio is 0.5, 1 and 2, the optimal number of samples in a pool is 1, 3 and 6, respectively, and the corresponding optimal number of array is 10, 8 and 6, respectively.

Note that usually the results are not integer, we have to round them to the nearest integer as the numbers of arrays and samples should be always integers. The ratio of the biological to technical variance is an important value for deciding the optimal pooling strategy. Biological variation is fixed but the technical variation σ_η would be shrunk by continual improvements of technology so that the cost associated with the array can be reduced.

2.5 Optimal distant pair design

2.5.1 Introduction

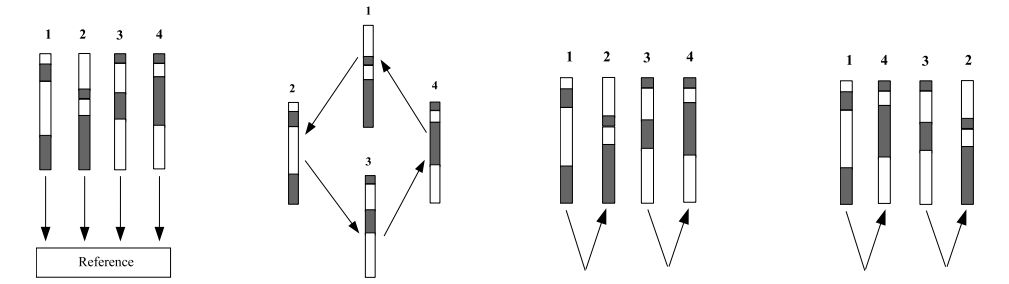
In recent years, the combined study of microarray gene expression and molecular marker data has been proposed as a novel strategy for the analysis of gene regulatory networks (Darvasi, 2003; Jansen and Nap, 2001, 2004; Kraft and Horvath, 2003).

Such a kind of study usually involves relatively large number of genotypes (i.e. number of conditions), since for n markers on a genome the corresponding number of potential genotypes is 2^n. Therefore, even for a small number of markers studied, the number of genotypes will be quite large and the resulting experimental costs may become prohibitive. So it requires that resources (i.e. microarrays and biological replicates) should be used as efficiently as possible.

Instead of the popular reference and loop designs where samples are compared to a common reference sample or to each other in a loop order, Fu and Jansen (2005) proposed a new strategy for two-color cDNA microarrays, called distant pair design.

To illustrate the design issues involved, we consider expression profiling a population of recombinant inbred lines (RILs). RILs are homozygous individuals, which result from repeated self-self mating or sibling mating, starting from a F1 of two homozygous parents, carrying alleles of type A and B respectively. The genome of a RIL is therefore a mosaic of the "founder" genomes, which can be viewed with the aid of molecular markers. See Figure 2.11.

The idea of the reference design is to compare all conditions (RILs) to one common reference. The loop design co-hybridizes the first RIL with a second RIL on one array, this second RIL with a third RIL on a second array, and so on.



Reference design Loop design Random pair design Distant pair design

Figure 2.11: Illustration of four alternative experimental designs (Fu and Jansen, 2005). The hypothetical compositions of four genetically different homozygous individuals are shown, each individual carrying different mixtures of two founder genome (dark and light). Four alternative designs to pair samples with two-channel microarrays are indicated: the reference design, the loop design, the random pair design (samples are randomly paired), and the distant pair design (samples with dissimilar genomes are paired).

This way all RILs can be profiled, not just once as in the reference design, but twice, which is a great improvement in the use of microarray resources.

One could alternatively use a random pair design, where the first RIL is compared to a randomly chosen second RIL, a randomly chosen third RIL to a fourth RIL, and so on. For each direct comparison between RIL i and RIL j with red and green dye respectively, there are four possible combinations at a given marker: A/B (RIL i carries allele A, RIL j carries allele B), B/A (RIL i carries allele B, RIL j carries A), A/A or B/B (RIL i and j carry the same alleles). These four combinations occur with equal probability in a random pair design. We are primarily interested in detecting differential expression between A and B, thus A/B and B/A are of interest, and A/A and B/B are not. Therefore, a natural next step is to improve the random pair design in such a way that the number of A/B and B/A comparisons is maximized and with minimal extra variation

of total numbers A/B and B/A across the different markers. For this purpose, Fu and Jansen (2005) proposed the optimality criterion for distant pair design which co-hybridizes RILs that show to be genetically distant according to their molecular marker fingerprints. The optimality criterion is initially proposed for the case of single marker and then extended for the case of multiple markers by simply averaging the criterion for single marker. In this section, we first introduce the gene expression model for multiple markers and then propose an alterative (and more proper) A-optimality criterion for the case of multiple markers.

2.5.2 Model

The variation of gene expression is caused by genetic variation at a regulatory locus. Since the expression level of the gene under study may be high or low when the regulator locus has genotype A (B) or vice versa, we can observe two types of ratio: the informative A/B and B/A, and the relatively uninformative A/A and B/B. The expression ratios of the latter type should be close to unity (i.e. one in original scale), unless there is dye-bias. This can be formulated into mathematical models.

For the microarray gene expression on a single channel with dye d (i.e. Cy3 or Cy5), we have:

$$z_{d,i} = \alpha_d + \sum_{j=1}^{k} \beta_j x_{d,ij} + \varepsilon_i \tag{2.27}$$

where $z_{d,i}$ is the gene expression with dye d for individual i after taking logarithm. $x_{d,ij}$ corresponds to the genotype (A or B) at marker j for array i with dye d and takes the following values: 0 for A and 1 for B. α_d is the gene-specific effect for dye d. β_j is the effect of allele expression at jth marker under study. k is the

number of different markers (regulatory loci) on the genome. ε_i is the normal distributed error with mean zero and variance σ^2. Note that Equation (2.27) can be considered as a special case of $2 \times k$ multi-factorial design where each marker is a factor although it ignores the higher order effects (the interactions among the k markers).

For array i, we observe the log-ratio of gene expression from its two channels:

$$\begin{aligned} y_i &= z_{Cy3,i} - z_{Cy5,i} \\ &= \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \end{aligned} \tag{2.28}$$

where $\beta_0 = \alpha_{Cy3} - \alpha_{Cy5}$, $x_{ij} = x_{Cy3,ij} - x_{Cy5,ij}$. The possible values for x_{ij} are -1 when $x_{Cy3,ij} = 0$ and $x_{Cy5,ij} = 1$ (i.e. A/B), 1 when $x_{Cy3,ij} = 1$ and $x_{Cy5,ij} = 0$ (i.e. B/A) and 0 when $x_{Cy3,ij} = 0$ and $x_{Cy5,ij} = 0$ or $x_{Cy3,ij} = 1$ and $x_{Cy5,ij} = 1$ (i.e. A/A or B/B). The sign of x_{ij} (i.e. from 1 to -1 or from -1 to 1) is determined by the way of dye assignment to the two channels (RILs).

If the experiment contains n arrays, we can write in matrix notation that:

$$y = X\beta + \varepsilon \tag{2.29}$$

where $y = (y_1, \ldots, y_n)^t$, $\beta = (\beta_1, \ldots, \beta_k, \beta_0)^t$, X is a n by $k+1$ matrix where $X_{ij} = x_{ij}$ for $i = 1, \ldots, n$, $j = 1, \ldots, k$ and $X_{ij} = 1$ for $i = 1, \ldots, n$, $j = k+1$.

Using the least squares method, we have the estimate of β: $\hat{\beta} = (X^t X)^{-1} X^t y$ and its variance-covariance matrix: $V(\hat{\beta}) = (X^t X)^{-1} \sigma^2$.

2.5.3 Optimality criteria

Based on the single-marker gene expression model (Equation (2.28) for $j = 1$), Fu and Jansen (2005) proposes an A-optimality criterion of the distant pair design for the case of single marker, which finds the minimum of

$$\frac{n + \sum_{i=1}^{n} x_{i1}^2}{n \sum_{i=1}^{n} x_{i1}^2 - (\sum_{i=1}^{n} x_{i1})^2}, \tag{2.30}$$

then they extend it to the case of multiple markers by summing or averaging over all markers of the variance of $\hat{\beta}$,

$$S = \sum_{j=1}^{k} \left\{ \frac{n + \sum_{i=1}^{n} x_{ij}^2}{n \sum_{i=1}^{n} x_{ij}^2 - (\sum_{i=1}^{n} x_{ij})^2} \right\}, \tag{2.31}$$

where j refers to the jth marker and k is the number of markers. This is identical to optimizing $\frac{n+\sum_{i=1}^{n} x_{ij}^2}{n \sum_{i=1}^{n} x_{ij}^2 - (\sum_{i=1}^{n} x_{ij})^2}$ for $j = 1, \ldots, k$ separately.

As an alternative, here we propose an A-optimality criterion based on the multiple-markers gene expression model (Equation (2.28)). It minimizes the sum of the variances of $\hat{\beta}$ for given markers. This is equivalent to choosing X such that the trace of the matrix $(X^t X)^{-1}$ is smallest. Obviously, this criterion is applicable to not only the case of single marker (i.e. $k = 1$) but also that of multiple markers (i.e. $k = 2, 3, \ldots$).

2.5.4 Example

Now we use a simple example to show that our proposed A-optimality criterion for the case of multiple markers is more proper than that of Fu and Jansen.

We assume that there is a microarray experiment with $n = 4$ arrays. Each array pairs two RILs, so that $2n = 8$ RILs are involved. We also assume that

Table 2.3: The corresponding x_{i1} and x_{i2} values are listed for the 16 possible combinations from the four types of RILs a, b, c and d.

	1	2	3	4	5	6	7	8	9	10
RILs	a/a	a/b	a/c	a/d	b/b	b/c	b/d	c/c	c/d	d/d
Marker 1	A/A	A/A	A/B	A/B	A/A	A/B	A/B	B/B	B/B	B/B
Marker 2	A/A	A/B	A/A	A/B	B/B	B/A	B/B	A/A	A/B	B/B
x_{i1}	0	0	-1	-1	0	-1	-1	0	0	0
x_{i2}	0	-1	0	-1	0	1	0	0	-1	0
		11	12	13		14	15		16	
RILs		b/a	c/a	d/a		c/b	d/b		d/c	
Marker 1		A/A	B/A	B/A		B/A	B/A		B/B	
Marker 2		B/A	A/A	B/A		A/B	B/B		B/A	
x_{i1}		0	1	1		1	1		0	
x_{i2}		1	0	1		-1	0		1	

the number of markers on the genome is $k = 2$ and there are 4 types of RIL: $a = \{A, A\}$ (RIL carries allele A and A on the first and second marker), $b = \{A, B\}$ (RIL carries allele A and B on the first and second marker), $c = \{B, A\}$ (RIL carries allele B and A on the first and second marker) and $d = \{B, B\}$ (RIL carries allele B and B on the first and second marker).

Given four types of RILs, there are 16 possible RIL combinations in an array. Recall that x_{ij} in Equation (2.28) takes -1 for A/B, 1 for B/A, and 0 for A/A and B/B. Then, we know the corresponding x_{i1}, x_{i2} values of all these combinations for this example, see Table 2.3.

Because of the small number of markers and arrays (i.e. $k = 2$, $n = 4$), it is easy for us to give the expression of A-optimality score explicitly. If we ignore

the dye effect item β_0 in Equation (2.29), we have the design matrix:

$$X = \begin{bmatrix} x_{11} & x_{21} & x_{31} & x_{41} \\ x_{12} & x_{22} & x_{32} & x_{42} \end{bmatrix}^t,$$

then we have

$$X^tX = \begin{bmatrix} \sum_{i=1}^4 x_{i1}^2 & \sum_{i=1}^4 x_{i1}x_{i2} \\ \sum_{i=1}^4 x_{i1}x_{i2} & \sum_{i=1}^4 x_{i2}^2 \end{bmatrix},$$

and

$$(X^tX)^{-1} = \frac{1}{\sum_{i=1}^4 x_{i1}^2 \sum_{i=1}^4 x_{i2}^2 - (\sum_{i=1}^4 x_{i1}x_{i2})^2} \begin{bmatrix} \sum_{i=1}^4 x_{i2}^2 & -\sum_{i=1}^4 x_{i1}x_{i2} \\ -\sum_{i=1}^4 x_{i1}x_{i2} & \sum_{i=1}^4 x_{i1}^2 \end{bmatrix},$$

therefore finding A-optimal design is to minimize the score:

$$(X^tX)^{-1} = \frac{\sum_{i=1}^4 x_{i1}^2 + \sum_{i=1}^4 x_{i2}^2}{\sum_{i=1}^4 x_{i1}^2 \sum_{i=1}^4 x_{i2}^2 - (\sum_{i=1}^4 x_{i1}x_{i2})^2}. \tag{2.32}$$

where i refers to the ith array. Note that it favors large $\sum_{i=1}^4 x_{i1}^2$, $\sum_{i=1}^4 x_{i2}^2$ and small $\sum_{i=1}^4 x_{i1}x_{i2}$. Here $\sum_{i=1}^4 x_{i1}^2$ and $\sum_{i=1}^4 x_{i2}^2$ represent the total number of informative A/B and B/A comparisons for the two markers respectively (should be large), $\sum_{i=1}^4 x_{i1}x_{i2}$ represents the difference of the number of arrays with different comparison on the two markers (i.e. A/B and B/A) and the number of arrays with the same comparison on the two markers (i.e. A/B and A/B or B/A and B/A) (should be small).

For the optimality criterion proposed by Fu and Jansen (2005), it is to find

the minimum of

$$S = \sum_{j=1}^{2} \left\{ \frac{4 + \sum_{i=1}^{4} x_{ij}^2}{4 \sum_{i=1}^{4} x_{ij}^2 - (\sum_{i=1}^{4} x_{ij})^2} \right\}, \tag{2.33}$$

where j refers to the number of markers. Note that it is an average of the optimality scores for multiple markers. When single marker is considered ($j = 1$), this criterion favors a design with large $\sum_{i=1}^{4} x_{i1}^2$ and small $\sum_{i=1}^{4} x_{i1}$. Here $\sum_{i=1}^{4} x_{i1}^2$ represents the total number of informative A/B and B/A comparisons (should be large), and $\sum_{i=1}^{4} x_{i1}$ represents the difference between the number of A/B comparisons and the number of B/A comparisons (should be small, i.e. dyes should be well balanced).

In this example if we propose a experimental design by randomly selecting 4 RIL pairs from the 16 types of RIL pairs in Table 2.3 (ordered and with replacement), there are $16^4 = 65536$ possibilities.

In order to find the optimal designs, we compute the optimality scores for all the 65536 possible designs using Equation (2.32) and (2.33) respectively and determine the optimal designs which have the smallest optimality score.

Under our proposed A-optimality criterion, we find 96 optimal designs with the same score 0.5. If we ignore the permutation of the RIL pairs in the design, the number of optimal design is reduced from 96 to 9. Table 2.4 shows these results, which are almost exactly the same (i.e. all consist of two pairs of a and d and two pairs of b and c) except the difference in the RIL's order in a pair (i.e. a/d or d/a, b/c or c/b). The reason that they have the same score can be found in Equation (2.32), which shows that the optimality score is invariant to the change of the RIL's order in a pair (i.e. the change of sign of x_{ij}). Further, if we check the corresponding markers for the pairs a/d, d/a, b/c and c/b in

Table 2.3, we find all the pairs have the maximum number (i.e. two) of informative A/B and B/A comparisons possible (a/d={A/B,A/B}, d/a={B/A,B/A}, b/c={A/B,B/A} and c/b={B/A,A/B}), which validates the optimality of these findings.

For simplicity, we neglect the dye effect item β_0 in Equation (2.29) in the derivation of Equation (2.32) so that in the optimal findings the RIL's order in a pair is irrelevant. Now, if we include the dye effect item β_0, then we have a new design matrix:

$$X = \begin{bmatrix} x_{11} & x_{21} & x_{31} & x_{41} \\ x_{12} & x_{22} & x_{32} & x_{42} \\ 1 & 1 & 1 & 1 \end{bmatrix}^t,$$

and the corresponding expression form of $(X^tX)^{-1}$ is cumbersome to derive explicitly. As a simple way out, in this example we do not show the explicit expression but compute the value of this expression directly by using `R`. We find 24 optimal designs with the same score 0.75. If we ignore the permutation of the order of RIL pairs in the design, then the number of findings is reduced from 24 to 1, which is listed in the first row of Table 2.4. In this design (i.e. { a/d, d/a, b/c, c/b }), all the pairs not only have the maximum number (i.e. two) of informative A/B and B/A comparisons possible, but also take the dye balance into account.

Under Fu & Jansen's criterion, we find 36 optimal designs with score 1.0. If we ignore the permutation of the order of RIL pairs in the design, the number of optimal designs is reduced from 36 to 3. Table 2.5 shows the findings.

Under our proposed criterion, only 1 out of the 3 designs is found to be A-optimal (which is exactly the first design shown in Table 2.4) while the other 2 designs are not optimal, because they are unable to estimate all the effects in

Table 2.4: The 9 optimal designs found by our A-optimality criterion (dye effect excluded) and also their corresponding optimality scores under our criterion (dye effect included) and Fu & Jansen's criterion.

	design				optimality score		
					ours	ours with dye effect	Fu & Jansen's
1	a/d	d/a	b/c	c/b	0.50	0.75	1.00
2	a/d	d/a	b/c	b/c	0.50	1.25	1.33
3	a/d	d/a	c/b	c/b	0.50	1.25	1.33
4	a/d	a/d	b/c	c/b	0.50	1.25	1.33
5	a/d	a/d	b/c	b/c	0.50	NaN	Inf
6	a/d	a/d	c/b	c/b	0.50	NaN	Inf
7	d/a	d/a	b/c	c/b	0.50	1.25	1.33
8	d/a	d/a	b/c	b/c	0.50	NaN	Inf
9	d/a	d/a	c/b	c/b	0.50	NaN	Inf

Table 2.5: The 3 optimal designs found by Fu & Jansen's criterion and also their corresponding optimality scores under our criteria (dye effect ignored or considered)

	design				optimality score		
					Fu & Jansen's	ours	ours with dye effect
1	a/d	a/d	d/a	d/a	1.00	Inf	NaN
2	a/d	b/c	d/a	c/b	1.00	0.5	0.75
3	b/c	b/c	c/b	c/b	1.00	Inf	NaN

the additive main effects model (see their extreme large optimality scores or not being a number).

Although Fu & Jansen's criterion not only aims to maximize the number of A/B or B/A comparisons but also ask for dye balance, it is only applied for one marker. That's why it finds designs like {a/d, a/d, d/a, d/a} and {b/c, b/c, c/b, c/b}, which are not truly optimal for the case of multiple markers. In contrast, our criterion (including the dye effect) is for all the markers on the genome and takes into account of the joint effect of genes.

As a conclusion, in this example we show that Fu and Jansen's A-optimality criterion might result in non-optimal findings and our proposed A-optimality criterion is a more proper choice when the studied gene expression is affected by multiple markers. The key reason is that our criterion is formulated from the multiple-markers gene expression model which inherently considers the joint effect of markers while Fu and Jansen's single-marker gene expression model plus taking average can not grasp such joint effects properly. We recommend our A-optimality criterion for distant pair design.

Chapter 3

Dye effect normalization

3.1 Introduction

The current technology of dual-channel cDNA microarray is based on measuring optical intensities of dye labeled cDNA that has hybridized to gene-specific probes on the microarray. Two types of dyes Cy3 and Cy5 are commonly applied in the experiment so that the corresponding two labeled cDNA samples on the array can be distinguished. Despite similarities, the dyes have slightly different properties (Wit and McClure, 2004). Firstly, the quantum yield from the dyes is different. Secondly, the sizes of the Cy3 and Cy5 molecules differ slightly, which leads to different numbers of dye molecules attaching to the samples. Thirdly, the dyes react differently to photo-bleaching, an effect that occurs as a result of multiple scans of the array (Chris and Ghazal, 2003). Besides the different efficiencies of dyes, the unequal quantities of the two samples being mixed is another source of a dye effect. As a result of all these issues, the direct comparison of dual-channel gene expression data is difficult and some ways should be found to remove as much bias as possible.

This chapter consists of three sections. In the first section, we review the background of the dye effect in cDNA microarray experiments and discuss the normalization methods proposed to account for the dye effect. In the second section, we propose our dye response model and, based on this model, we suggest a new normalization method for the dye effect. In the final section, we compare the normalization methods by using simulated microarray gene expression data and real microarray data.

3.1.1 Linear and non-linear dye effects

Several normalization methods have been proposed to deal with the dye effect. Early research by Kerr et al. (2000) suggested correcting the effect by a constant, possibly different for each array (also called "dye-array interaction"). This means that the dyes are assumed to have efficiencies that differ by an multiplicative constant. This corresponds to a linear relationship between the log-transformed expressions in the Cy3 and Cy5 channels. However in most cases this assumption does not always stand up to further scrutiny.

To better understand it, we take a look at a real microarray dataset example which was produced by Dr Nighean Barr, a researcher at the Cancer Research UK Beatson Laboratories in Glasgow. The experiment that she carried out investigated differences between gene expressions in cancerous and normal fibroblast cells. These cells are a key constituent of connective tissue within the body and make fibres and the extracellular matrix. In the skin, these cells are susceptible to become cancerous if exposed to UV radiation from sunlight. By finding which genes are differentially expressed in the cancerous versus normal cells, one can focus research into treatments for cancer. This skin cancer experiment is a direct

Table 3.1: The design details of the skin cancer experiment.

Array	Cy3 sample	Cy5 sample
1	Cancer	Normal
2	Normal	Cancer
3	Cancer	Normal
4	Normal	Cancer

comparison of cancerous fibroblast cells with normal cells. From each of these two cell lines, four technical replicates were created, and hybridized to dual channel cDNA arrays which contained 4,608 genes replicated twice. Finally, four arrays were produced in the experiment. For the first and third arrays, the normal tissue was stained with Cy5 dye and the cancerous tissue was stained with Cy3. On the other two arrays the dye assignments were swapped. The design details are given in the Table 3.1.

Figure 3.1(a), (b), (c) and (d) represent the scatter-plots of the log-transformed data from the array 1, 2, 3 and 4 respectively. Note that spatially normalized data is used, rather than the original log-transformed data. The main reason for it is that the dye effect might be confounded with a spatial effect on the array. It is essential to perform spatial normalization in advance, or else it may lead to bias. See Wit and McClure (2004, pg. 132) for more details. Global normalization methods are not enough when dye bias depends on the overall spot intensity. None of the four plots suggest a linear relationship. Besides Figure 3.1(a) which looks relatively "linear", all of the remaining plots show a very clear deviation from a constant dye effect. The relative efficiency of the dyes seem to vary across the intensity range. However, some obvious differences can be found: in Figure 3.1(b), Cy5 dye seems to have been incorporated more efficiently while in Figure 3.1(c) and (d), Cy3 seems to have been incorporated more efficiently, these plots

share a very common pattern: Cy3 dye seems to have gained in efficiency relative to Cy5 in the middle of the intensity range, which is known as “banana effect”.

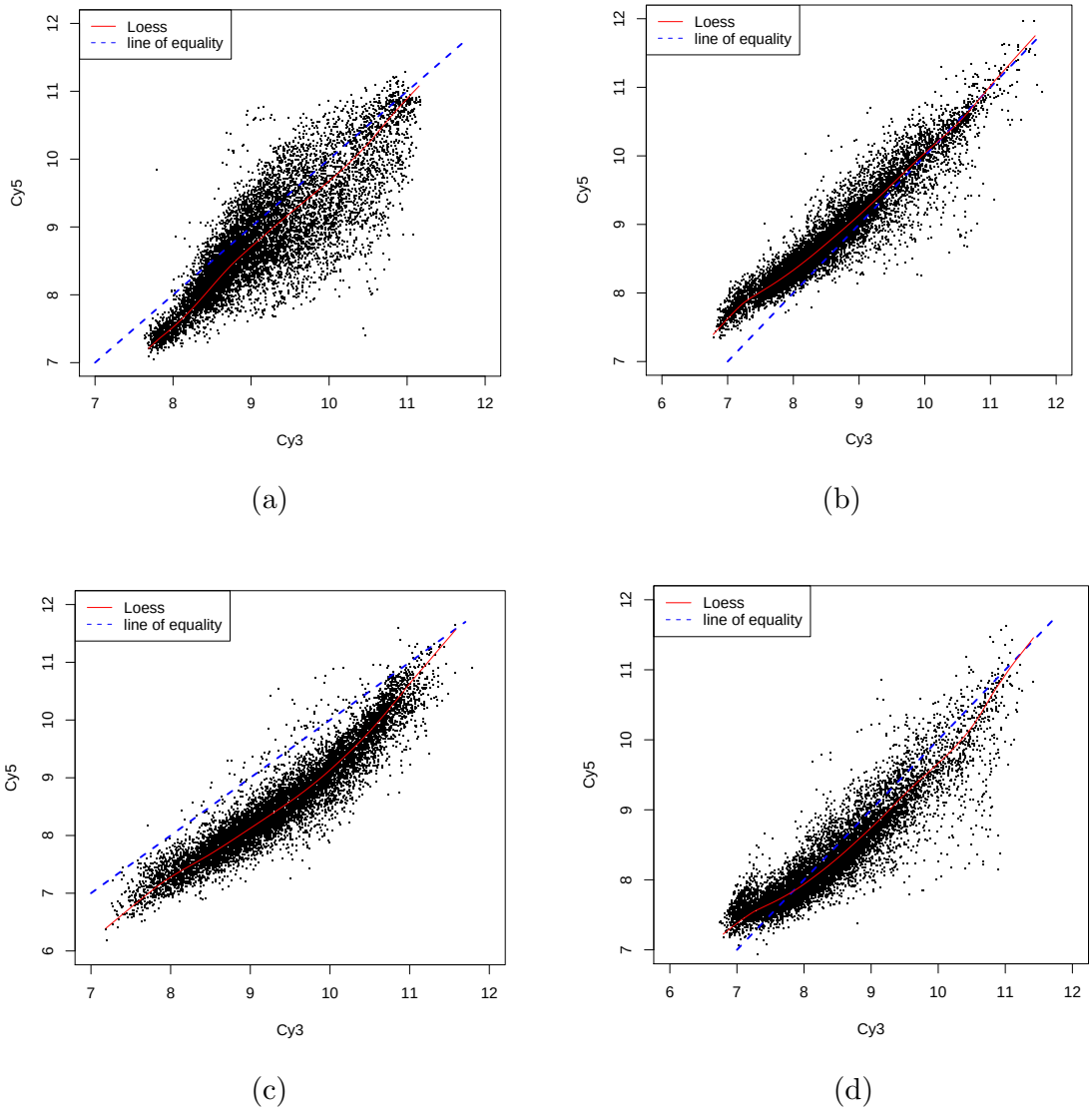


Figure 3.1: The log-transformed data (after taking global normalization) from four different cDNA slides from the skin cancer experiment. Each point in the scatter-plots represents a spot (gene) on the array. The x-axes and y-axes stand for the Cy3 and Cy5 value respectively. The lines in both plots correspond with the line of equality and a loess smoother through the points. Although plots of all the four arrays show a non-linear relationship between the dyes, plots (b), (c) and (d) are more obvious.

3.1.2 Dye effect normalization methods

The dye effects are intensity-dependent. Three methods have been proposed for dealing with intensity-dependent dye effects so far. The simplest one is called dye-swap normalization, which consists of repeating a hybridization twice with the dyes swapped and averaging the expression values for each spot over the Cy3 and Cy5 channel. The second method is to consider the dye effect as a nuisance effect in an ANOVA model. The third method includes two steps, firstly estimating the relative dye efficiency at each intensity and then subtracting it from the original data.

3.1.2.1 Dye swap method

Dye-swap normalization is an easy and intuitive way of eliminating dye effects and is very popular among practitioners. In spite of these merits, it has two potential disadvantages. The first one is that the dye effect might not be the same from array to array, which means there is no guarantee that this method can effectively remove the dye effect. The second one is that a dye-swap experiment design is not very efficient especially, since it needs twice as many resources as the ordinary loop design.

3.1.2.2 ANOVA method

The analysis of variance (ANOVA) method handles normalization and data analysis simultaneously, by modelling the nuisance effect and the estimation of condition-specific gene expression.

A single gene expression data can be denoted as y_{ijkg}, which is the fluorescence measurement for the mRNA of gene g under condition k, labelled with dye j on

the ith array. In order to account for these sources of variation in a microarray experiment, Kerr et al. (2000) proposed an ANOVA model for the gene expression data under the logarithmic scale:

$$\log(y_{ijkg}) = \mu + A_i + D_j + T_k + G_g + (AG)_{ig} + (TG)_{kg} + \epsilon_{ijkg}, \tag{3.1}$$

where μ is the overall mean value, A is the main effect of arrays, D is the main effect of dyes, T is the main effect of treatments, G is the main effect of genes, AG is the interaction effect of arrays and genes and accounts for the spot-to-spot variation, TG is the interaction effect of treatments and genes and is the effect of interest. ϵ is the random error which is assumed to be independent and identically distributed with mean zero.

Unfortunately, it has been widely recognized that many artifacts are non-linear or intensity dependent and a simple linear model like Equation (3.1) is not sufficient (Tseng et al., 2001; Wolkenhauer et al., 2002; Yang and Speed, 2002). Although it is theoretically possible to propose more complex models involving all the effects simultaneously, the computation is usually infeasible.

3.1.2.3 Two-step intensity-dependent dye normalization method

The idea of the two-step method of intensity-dependent dye normalization is to fit a smooth curve to a scatter plot of Cy5 versus Cy3 values, such as in Figure 3.1. The method has several problems. First, the model is not invariant under the exchange of the axes. Given that neither Cy3 nor Cy5 is a natural response value, this is not very satisfactory. Second, the usual residuals are not the smallest distances to the smoothed line. Orthogonal distances, the perpendicular distances

from the data points to the fitted line, could be used, but this tends not to be very standardly implemented. Instead, Yang and Speed (2003) suggest the smoothing of the data on a transformed scale. Basically, the average of the two dye values (A) are considered a predictor variable for the differences of the two dye values (M) and it is on this transformed version of the data that the function conducts the smoothing. This method is sometimes known as the MA scatter plot.

The two-step method of intensity-dependent dye normalization uses all or part of the data to estimate a line of equal expression and then to define individual gene expressions as deviations from that line. In general, there are two criteria for the choice of a normalization set. First, the expression of the selected genes should be expected to be approximately equal across both dyes, which implies that they are very likely to be non-differential genes, so that the risk of "normalizing away" true differential expressions can be minimized. Second, the normalization set should be relatively large and ideally, the expressions of the selected genes should distribute evenly across the whole range of the intensity so that the experimental noise of the normalization curve can be reduced to minimum.

The details of a variation of the two-step method of intensity-dependent dye normalization are described below.

1. For each probe i in the invariance set N, transform the raw Cy3 and Cy5 values, G and R respectively, via 45 degree log transformation as follows:

$$m_i = \log(R_i) - \log(G_i), \tag{3.2}$$

$$a_i = 0.5 \times (\log(R_i) + \log(G_i)). \tag{3.3}$$

2. Find a smooth curve function $\tilde{f}$ through points by using a scatter plot smoother.

3. In order to remove the "trend" from the differences (M), we subtract the data from the smoothing line to get the residuals of the data, which constitute a normalized MA plot,

$$\tilde{m}_i = m_i - f(a_i), \tag{3.4}$$

$$\tilde{a}_i = a_i. \tag{3.5}$$

4. By taking inverse of Equations (3.2) and (3.3) and using normalized MA in Equations (3.4) and (3.5), we have the dye-normalized gene expression values in the original scale,

$$\log(\tilde{R}_i) = \tilde{a}_i + 0.5\tilde{m}_i, \tag{3.6}$$

$$\log(\tilde{G}_i) = \tilde{a}_i - 0.5\tilde{m}_i. \tag{3.7}$$

By scatter plot smoother we mean a method that draws a smooth curve through the scatter-plot of M vs A. If we think M as response variable and A as the explanatory variable, then there is a wide range of methods available, such as local polynomial regression (Cleveland, 1979; Cleveland and Devlin, 1988) or smoothing splines.

Local polynomial regression, also known as LOESS, is a smoothing method. It combines much of the simplicity of linear least squares regression with the

flexibility of nonlinear regression. It does this by fitting simple models to localized subsets of the data to build up a function that describes the deterministic part of the variation in the data point by point. In detail, at each point in the data set a low-degree polynomial is fit to a subset of the data, with explanatory variable values near the point whose response is being estimated. The polynomial is fit using weighted least squares, giving more weight to points near the point whose response is being estimated and less weight to points further away. The value of the regression function for the point is then obtained by evaluating the local polynomial using the explanatory variable values for that data point. The LOESS fit is complete after regression function values have been computed for each of the data points.

LOESS depends on smoothing parameters. There are two types of parameter. One is the degree of the polynomial fitted locally to the data and the other is the fraction of the data to be included in the smoothing of each point, the larger the fraction, the smoother the fit. We use the `loess` function from the statistical software package `R` to perform intensity-dependent dye normalization. Yang et al. (2002*a*) recommended using 20% of the data to be included in the smoothing of each point, that is `span` = 0.2. The default degree of the polynomials in `loess` is two, but we recommend to use linear functions, that is, polynomials of the first degree. The reason for it is that high-order polynomials tend to be unstable, particularly near the edges.

Smoothing splines is another popular smoothing method, which fits a cubic smoothing spline to the supplied data. In `R`, the function `smooth.spline` is implemented.

For illustration purposes, let's consider a numerical example. The data we use is from the second skin cancer array in Figure 3.1(d) which shows a clear

non-linear dye effect. The array contains no information about spiking controls or housekeeping genes, which are expected to be similarly expressed across the two dyes. We decided to use the part of the whole data which is defined as those genes whose relative rank among all 9,216 values has not changed by more than 250. In this way, 3,404 genes are selected as invariant ones. The Cy3 and Cy5 values are transformed according to Equations (3.2) and (3.3). Figure 3.2 (b) shows the transformed values for this array. The invariant genes are used to compute the smoothed curve of equality. Figure 3.2 (c) shows the results of using two different smoothing approaches in `R`, namely `loess` and `smooth.spline`. For the LOESS method, the parameter setting `span` = 0.2 and `degree` = 1 is more stable than the default setting `span` = 0.75 and `degree` = 1, especially at the edges. The smoothing spline's performance is much worse than LOESS in this case. In fact we have also applied the smoothing spline to other data in the skin cancer experiment and we have found that it isn't a very stable method (so we do not recommend it for smoothing). After subtracting the LOESS line from the MA-plot, the data are back-transformed to give the dye normalized Cy3 and Cy5 values on the original scale. This final result is shown in Figure 3.2 (d), which does not show any intensity-dependent deviation from the line of equality.

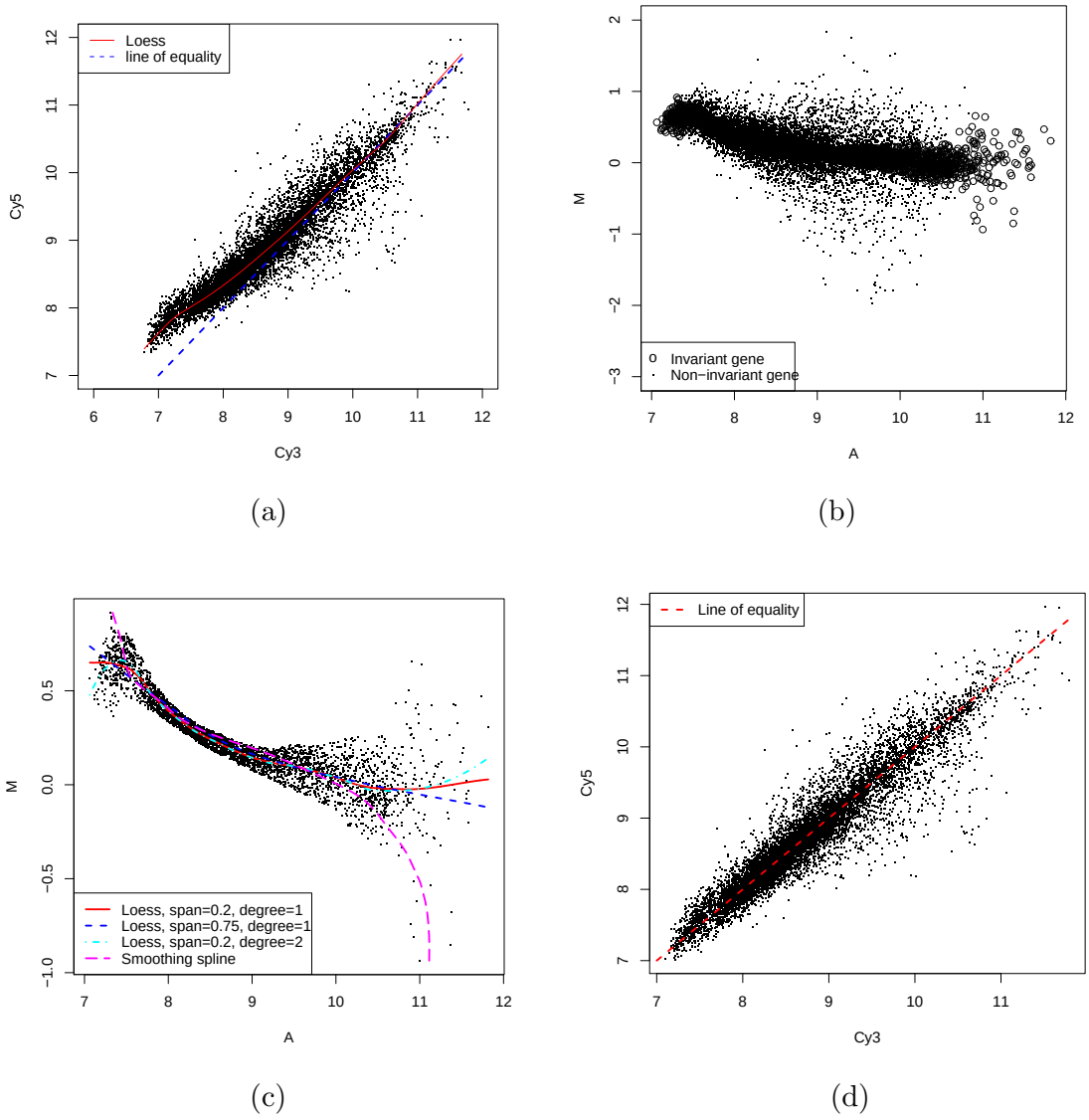


Figure 3.2: Dye normalization for the second skin cancer array. (a) the scatter plot shows the unequal dye efficiencies in the original data; (b) transform the original data into the MA scale; (c) for the invariant genes, a smoothed line is fitted to the scatter plot by using LOESS and smoothing spline; (d) the residuals of the smoothed regression are transformed back to the original scale.

3.2 Method

In a typical cDNA microarray experiment, the dye effect can be decomposed in two parts: an additive part, which comes from the unequal quantities of two samples onto the array (i.e. the interaction of dye and array) and a nonlinear part, which is non-linear response of a dye (i.e. the interaction of dye and gene). The former part is easy to remove but the latter part is quite difficult to deal with. As we mention in Section 3.1, most of people circumvent this challenge by using simple dye-swap normalization or two-step intensity-independent dye normalization with smoothing techniques. Although these methods can give good normalization results, there are no strong or direct scientific reasons supporting them, because none of the methods take into account the cause of the non-linear dye effect (i.e. the non-linear dye response). In the following, we first study the dye effect by considering the different responses of dyes, then based on it we suggest a novel normalization method for dye effect.

3.2.1 Dye response model

We set out to propose a model for dye response. Initially, we should consider this problem at pixel level. For a pixel of a spot on a microarray, we assume that the magnitude of signal intensity response of a pixel is strictly linear with the number of dye molecules on that pixel. This assumption is the foundation of dual-channel microarray technology. However, the scatter-plots of real microarray data (e.g. see Figure 3.1) show that there are always some non-linear dye effects remaining in the data, which means the assumption does not always hold strictly. Therefore we argue that, strictly the dye response to the quantity of the sample is not exactly linear, but contains a non-linear component.

3.2.1.1 Model one

We propose a simple dye response model for the ith pixel of the jth spot labeled with dye d as follows,

$$f_{d,ij}(x_{ij}) = \begin{cases} 1 & x_{ij} \in [0, a_{d,ij}], \\ \frac{M}{b_{d,ij}-a_{d,ij}}(x_{ij} - a_{d,ij}) + \varepsilon & x_{ij} \in (a_{d,ij}, b_{d,ij}), \\ M & x_{ij} \in [b_{d,ij}, N]. \end{cases} \tag{3.8}$$

where x_{ij} is the number of dye molecules on the pixel, $f(x_{ij})$ is the corresponding intensity signal, a is the intercept of the line of dye response with the lower horizontal limit line of the signal intensity and b is the intercept of the line of dye response with the upper horizontal limit line of the signal intensity. N denotes the largest possible number of dye molecules in a pixel, which is in fact unknown, M and 1 are set to be the largest and smallest possible signal intensity (M is $2^{16} - 1$ for a 16-bit microarray platform) so that $f_{d,ij}(x_{ij})$ will be truncated to be M and 1 when x_{ij} is larger than $b_{d,ij}$ or smaller than $a_{d,ij}$, ε is the error term which is assumed to be normally distributed with mean zero and variance σ_ε^2. Note that the dye response model for each pixel is different so that we use a subscript for the purpose of discrimination. The possible value of dye type d is 3 or 5 corresponding to Cy3 or Cy5 respectively.

In this model the range of the number of dye molecules is partitioned into three parts: the left end part, the central part and the right end part. When the number of dye molecules lies in the left and right end part of its range, the pixel signal intensity is set to be one and maximum respectively. When the number of dye molecules lies in the central part of its range, the pixel signal intensity is

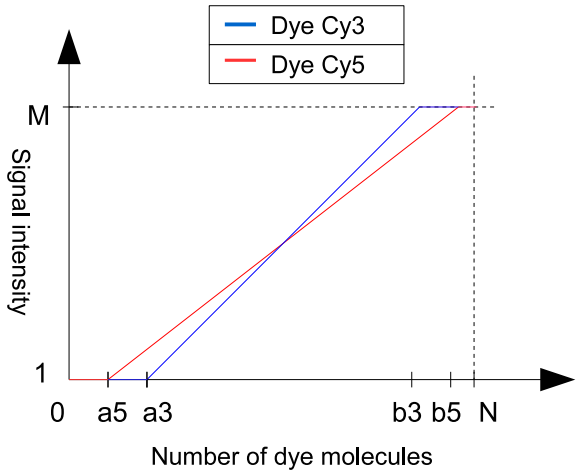


Figure 3.3: The two dye response models for a pixel in a spot on a dual-channel microarray. For Cy3 dye the linearity between the signal intensity and the number of dye molecules is assumed to be true when the number of dye molecule is in the range between $a3$ and $b3$. When the number of dye molecules is in the range between 0 and $a3$ or $b3$ and N, the signal intensity is truncated to be 1 and M respectively. For Cy5 dye the linearity between the signal intensity and the number of dye molecules is assumed to be true when the number of dye molecule is in the range between $a5$ and $b5$. When the number of dye molecules is in the range between 0 and $a5$ or $b5$ and N, the signal intensity is truncated to be 1 and M respectively.

strictly linear with the number of dye molecules, starting from one and ending at maximum. Since each of the pixels in the spot has some slightly different characteristics from each other, the dye molecules of a pixel might be more or less sensitive to the laser scanning. This would result in slightly different dye response patterns for different pixels. In our model it is achieved by setting the starting and ending points of the central part on the axis of dye molecule numbers as random variables. Figure 3.3 displays the dye response functions for two dye types according to the Equation (3.8). If the mean of the starting point and the ending point are closer to 0 and N respectively, the resulting curve of spot

dye response tends to be more linear. If the mean of the starting point and the ending point are far away from 0 and N respectively, the resulting curve of spot dye response then tends to be more nonlinear.

We define the signal of a spot, the gene expression, as the average of all pixel intensities in the spot: we have the jth spot signal intensity as:

$$y_{d,j} = \frac{1}{n} \sum_{i=1}^{n} f_{d,ij}(x_{ij}), \tag{3.9}$$

where n is the number of pixels in the spot. If each pixel has exactly the same dye response properties, then the dye response model for pixel intensity can be used directly for spot intensity.

Let us assume that there are a large number of genes on a microarray and for most genes the dye molecule numbers for the two channels are the same (e.g. most genes have very similar expression value under the two conditions, which is true in practice) and these genes are distributed evenly along the whole range of signal intensity. Then we can use these unchanged genes to study the relationship between the dissimilarity of the two dye response curves and the pattern of the scatter-plot of the two channel microarray gene expression data: According to the dye response functions in Figure 3.3, each non-differential expressed gene has two separate signal intensity values (i.e. Cy3 and Cy5 channels), given its dye molecule number. After taking logarithms of the values, we can determine the points for all the genes in the scatter-plot of the two channel microarray gene expression data. Then we can draw a smooth line through these points, which is called the "dye effect curve". By comparing it to the line of equality we can judge the dye effect pattern in the microarray gene expression data. In essence,

this method visualizes the dissimilarity of the efficiencies of the two dyes across the range of signal intensity from zero to the maximum.

In the Figure 3.4 we show that the simple dye response model (i.e. Equation (3.8)) is able to generate a variety of typical dye effect patterns. To avoid unnecessary complexity, firstly, we do not specify the exact type of dye in the figure, instead we just denote channel one or channel two to distinguish the two different data; secondly, we scale the range of number of dye molecules from $[0, N]$ into $[0, 1]$, which can be understood as the fraction of maximal number of dye molecules.

Figure 3.4(a) shows that if the curves of the two dye responses overlap completely then the resulting dye effect curve is just on the top of the line of equal expression which means no dye effect is found. Figure 3.4(b) shows that if the diagonal line part of the two dye response curves have the same intersection on the axis of dye molecule number, then the resulting dye effect curve parallels to the line of equal expression, which is a linear (constant) dye effect. The reason for it is quite simple. The diagonal line part of the two dye response curves only differs in slope. After taking logarithm of the original signal intensity, the difference is reduced to be the logarithm of the ratio of the slopes, which is a constant. Figure 3.4(c) shows that if the diagonal line part of the two dye response curves have no point of intersection, then the resulting dye effect curve is not linear any more and the efficiency of dye of channel one increases when the signal intensity decreases. Figure 3.4(d) shows that if the diagonal line part of the two dye response curves have only one intersection not on the axis of dye molecule number, then the resulting dye effect curve is also non-linear and the efficiency of dye of channel two increases when the signal intensity decreases. The dye effect curve intersects with the line of equality and the intersection corresponds to the

intersection of the diagonal line part of the two dye response curves. For Figure 3.4(c) and Figure 3.4(d), the absolute distance between the two intersections of the dye response curves and the axis of dye molecule number greatly affects the degree of non-linearity of the dye effect curve: the longer the distance, the larger the degree of non-linearity. Note that if we swap channel one data with channel two data in the scatter-plot we obtain a symmetrical dye effect curve with respect to the line of equality.

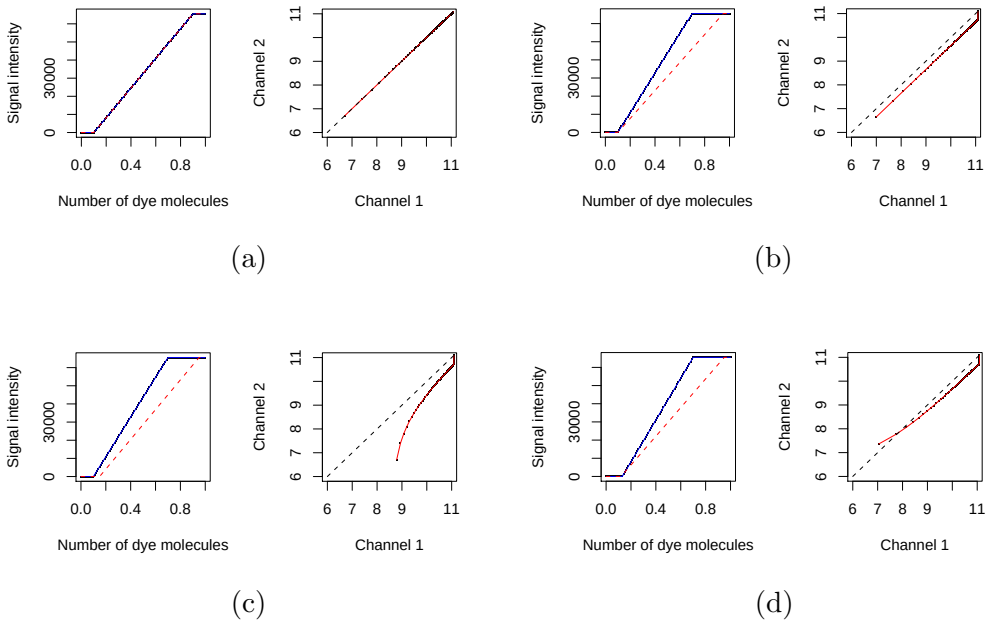


Figure 3.4: Dye effect patterns are caused by the dissimilarity of the two dye response curves (see Equation (3.8)). Several typical ones are shown here for illustration purpose: (a) no dye effect; (b) linear dye effect; (c) one type of non-linear dye effect and (d) another type of non-linear dye effect. For each figure the left subfigure shows the two dye response curves (red and blue curves) in original scale and the right subfigure shows the resulting dye effect pattern (red curve) in logarithm scale.

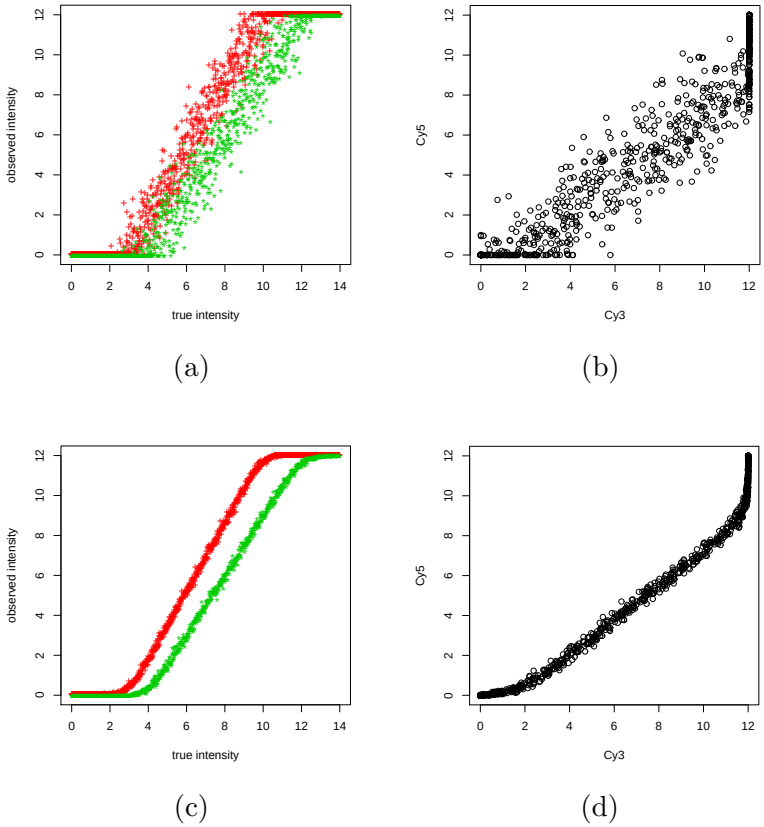


Figure 3.5: An example of the pixel level and spot level relationship for simple dye response model. (a) shows that 1000 observed gene expressions (pixel level) are generated to scatter across the whole range of true log intensity $[0, 14]$ according to two different simple dye response models (for the Cy3 dye response model, $a = 3$ and $b = 10$, for the Cy5 dye response model, $a = 4$ and $b = 12$. $M = 12$). Red "+" and green "*" stand for a gene in Cy3 and Cy5 channel respectively. (b) is the corresponding scatterplot of Cy3 vs Cy5. (c) shows the 1000 observed gene expressions at spot level (each spot consists of 40 pixels). (d) is the corresponding scatterplot of Cy3 vs Cy5.

Figure 3.4(c) and Figure 3.4(d) are similar to the real dye bias patterns we see in Figure 3.1. This finding justifies the effectiveness of our dye response model to some extent. Besides Figure 3.4, Figure 3.5 also clearly shows that the simple dye response model can result in "banana effect" pattern at the pixel level (see Figure 3.5 (a) and (b)) and at the spot level (see Figure 3.5 (c) and (d)).

Since the simple dye response model (see Equation (3.8)) is completely determined by specifying the starting point and ending point of the central part of the range of dye molecule number, it is natural for us to estimate these four parameters. If the dye response model is known, we can easily transform the observed spot signal intensity into the corresponding number of dye molecules which we regard as the real signal intensity. One standard way to estimate the parameters is to use the method of maximum likelihood estimation. For each gene (spot), we can write two equations according to Equation (3.9) and (3.8), one for dye Cy3 and the other for dye Cy5. Therefore, assuming that there are n non-differentially expressed observations (spots), we have $2n$ independent equations:

$$y_{d,j} = \left[\frac{M}{b_d - a_d}(x_j - a_d) + \varepsilon\right]_1^M, \qquad j = 1, \ldots, n, \quad d \in 3, 5. \tag{3.10}$$

where x_j is the number of dye molecules in the jth spot, $[\cdots]_1^M$ denotes that the value of the expression inside the square bracket is truncated to M if it is larger than M, and 1 if it is smaller than 1. Here we assume that all gene share the same a_3, a_5, b_3 and b_5.

By rearranging Equation (3.10), we manage to have the likelihood as follows,

$$\begin{aligned}&\text{Likelihood}(x_1, \cdots, x_n, a_3, b_3, a_5, b_5, \sigma^2) = \\ &\prod_{d=3,5}\prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{1}{2\sigma^2}\left[y_{d,j} - \left[\frac{M}{b_d - a_d}(x_j - a_d)\right]_1^M\right]^2\right\}.\end{aligned} \tag{3.11}$$

After taking the logarithm of Equation (3.11), we have the log-likelihood,

$$\begin{aligned}&\text{Log}-\text{likelihood}(x_1,\cdots,x_n,a_3,b_3,a_5,b_5,\sigma^2)=\\&\sum_{d=3,5}\sum_{j=1}^{n}\left\{-\log\sigma-\frac{1}{2\sigma^2}\left[y_{d,j}-\left[\frac{M}{b_d-a_d}(x_j-a_d)\right]_1^M\right]^2\right\}.\end{aligned}\tag{3.12}$$

Taking the first derivatives of the expression of Equation (3.12) with respect to $x_1,\cdots,x_n$, a_3, b_3, a_5, b_5, σ^2 and setting them to be zero leads to a system of $n+5$ independent equations for the $n+5$ unknown parameters. The system of equations is not uniquely soluble, because there are many combinations of x, a_3, a_5, b_3, b_5 and σ that give the same max log-likelihood.

It is important to note that the model is over-parametrized. It is possible for us to solve the equations by fixing several parameters. For example, we may let a_3 and b_3 fixed, then x_j is easy to estimate iteratively by using the following steps:

1. Use only data $y_{3,j}$ to estimate x_j.
2. With these x_j we can fit a_5 and b_5.
3. Use a_3, b_3, a_5, b_5 and all data $y_{3,j}$, $y_{5,j}$ to fit x_j.
4. Go back to Step 2.

After a few iterations it will converge. Note that this method assumes that the majority of the genes are not differentially expressed. Once we know the expression of the dye response model, we could do the dye normalization by mapping the observed gene expression (with dye effect) into the dye molecule numbers which represent the true gene expression intensity.

3.2.1.2 Model two

We can find an alternative dye response model whose parameters could be easily estimated. Recall that we made the assumption in Section 3.2.1.1 that each pixel of a spot has the same dye response properties so that the dye response model for the pixel intensity can be used for the spot intensity directly. This assumption is too strict and it is not likely to be true in practice. It is more reasonable to assume that each pixel has a slightly different dye response from each other. Equation (3.8) takes account of it by setting a and b as random variables. Then the resulting dye response model for a spot (i.e. Equation 3.9) will be different from that for a pixel. In particular, it will smooth the curves around the two turning points in Figure 3.3.

To illustrate this issue, we first scale the original pixel intensity and the number of dye molecules on a pixel by dividing by M and N respectively, and then calculate the average. Figure 3.6 shows such an example in a scenario where the number of pixels in a spot is 60, a is normally distributed with $\mu = 0.15$ and $\sigma = 0.08$ and b is normal distributed with mean $\mu = 0.85$ and $\sigma = 0.08$. Besides Figure 3.6, Figure 3.5 (c) also supports the argument to some extent.

We believe that the "S" curve in Figure 3.6 is a more reasonable model for spot dye response. There are several reasons for it: first of all, it accords with the main characteristics of dye response: continuous and strictly monotonic ascending; secondly, the "S" curve is not very different from a straight line, which is important, because it is unlikely for the spot dye responses to deviate far away from linearity; thirdly, it is relatively flexible and different "S" dye response curve for spot could be generated if the parameters of the model are adjusted. Therefore, we are motivated to find a model which can describe the "S" curve for spot dye response.

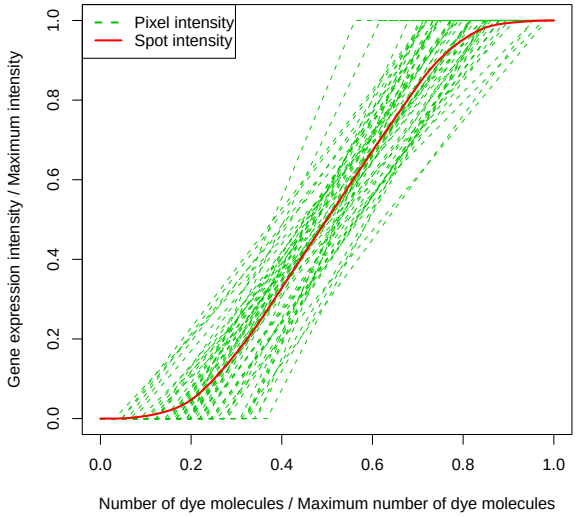


Figure 3.6: A spot's nonlinear dye response is generated by taking the average of its pixels' linear dye responses.

Inspired by the "S" shape curve in Figure 3.6, we propose that the cumulative distribution function (cdf) of the normal distribution is a nice candidate for the dye response model. The cdf, evaluated at a specific number x, is defined to be the probability of the event that a random variable with a normal distribution is less than or equal to that number. It is expressed in terms of the normal density function as follows,

$$\begin{aligned} \Phi_{\mu,\sigma^2}(x) &= \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(u-\mu)^2}{2\sigma^2} \right) du \\ &= \Phi\left(\frac{x-\mu}{\sigma} \right), \quad x \in \mathbb{R}, \end{aligned} \tag{3.13}$$

where the standard normal cdf Φ is the general cdf evaluated with mean $\mu = 0$

and standard deviation $\sigma = 1$:

$$\Phi(x) = \Phi_{0,1}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{u^2}{2} \right) du, \quad x \in \mathbb{R}. \tag{3.14}$$

The inverse cumulative distribution function, or quantile function associated with the normal distribution (probit function), can be expressed as:

$$\Phi^{-1}_{\mu,\sigma^2}(p) = \mu + \sigma\Phi^{-1}(p) = \mu + \sigma\sqrt{2}\text{erf}^{-1}(2p-1), \quad p \in (0,1), \tag{3.15}$$

where erf is called the error function and is defined as

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} \exp(-u^2) du. \tag{3.16}$$

erf^{-1}(x) can be represented by a series expansion as

$$\text{erf}^{-1}(x) = \sum_{k=0}^{\infty} \frac{c_k}{2k+1} \left(\frac{\sqrt{\pi}}{2} x \right)^{2k+1}, \tag{3.17}$$

where $c_0 = 1$ and

$$c_k = \sum_{m=0}^{k} \frac{c_m c_{k-1-m}}{(m+1)(2m+1)}. \tag{3.18}$$

If the cdf of normal distribution is used as a reference, the change of mean value of normal distribution can shift the "S" curve horizontally while the change of the value of variance can adjust the slope degree of the 'S' curve. Therefore it is natural to use the probit function to mimic a curve like the nonlinear dye response function in Figure 3.6 by adjusting these two parameters simultaneously.

There are some problems needed to be addressed before we apply the cdf of the normal distribution to model the dye response. Firstly, the range of the cdf

of normal distribution is only $[0, 1]$, and not $[1, 65535]$ which is original range of gene expression. Secondly, the "S" curve pattern of the cdf of normal distribution (our interest) does not cover the whole domain of x, $[-\infty, \infty]$, instead it largely lies on a small interval of x ranging from the mean minus three times standard deviation to the mean plus three times standard deviation.

For the first one, it can not be a problem if we consider it as the scaled gene expression (spot) intensity. We can linearly transform the range of the cdf of the normal distribution from $[0, 1]$ to $[0, 11.1]$ which approximately corresponds to the range $[1, 65535]$ of gene expression on the logarithmic scale.

For the second problem, we should realize that a spot's dye molecule number is equivalent to its true gene expression (also gene log expression) in concept, then we can arbitrarily let one of the two dye response function, say Cy3, be fixed to be the cdf of the normal distribution with $\mu = 8$ and $\sigma = 1$ (the other dye response function for Cy5 is slightly different from that of Cy3 with different mean and variance). Therefore the "S" curve pattern mostly lies on the interval of x from 5 to 11 in log scale, which approximately corresponds to the original gene expression interval from 148 to 59874 (very close to the limiting range of $[1, 65535]$). Note that the reason that one dye response function is fixed will be explained later.

Suppose that we use the cdf of the normal distribution for the dye response function as we describe above, can we model the dye response curve and do the dye normalization? If we use maximum likelihood, then we will meet the same problem as we have in the Section 3.2.1.2: it is difficult to estimate the parameters (i.e. mean, variance and x's) of the cdf function. So we have to find another way out.

In the following, we propose a new way to estimate the parameters of dye response function. The key idea is described as follows: From the microarray data, we get the dye effect pattern (i.e. original dye effect curve). Then we model a pair of dye response functions (Cy3 and Cy5) by using two separate cdf of normal distribution. Since the two dye response functions lead to a proposed dye effect curve, then we can compute the distance between the original dye effect curve and the proposed dye effect curve. In this way, we repeatedly propose alternative pair of dye response functions and calculate the corresponding distance until the minimum of distance is found. In a word, we just want to search for such a pair of dye response functions whose corresponding dye effect curve is equal to or very close to the original dye effect curve. However, we can not find an unique pair of dye response functions unless we specify one of the dye response functions in advance. That is why we have to fix one of the dye response function by specifying its mean and variance (i.e. the Cy3 dye response function). After we estimate the parameters of the Cy5 dye response function, we can do the dye normalization for the observed two channel spot signal intensities by mapping them to true gene expression data (i.e. dye molecule number) via the two dye response functions respectively.

The detailed steps of the algorithm are described below:

1. Read the observed Cy3 and Cy5 gene expression data from a microarray which may contain tens of thousand of genes. We select part of the whole dataset which is defined as those genes whose relative rank among all the values has not changed by a relatively small number, let's say, 250. Using the selected genes, we can draw the dye effect curve in the scatter-plot of microarray data Cy3 vs Cy5. We call this the target dye effect curve.

2. Assuming the Cy3 dye response function is known (i.e. the cdf of the normal distribution with $\mu = 8$ and $\sigma = 1$), we propose the cdf of normal distribution for Cy5 dye response function with initial mean and variance value (i.e. Equation (3.13)). Subsequently we can determine the resulting proposed dye effect curve in the scatter-plot of microarray gene expression data Cy3 vs Cy5. Then the distance between the target curve and the proposed dye effect curve is computed according to Equation (3.19) below.

3. Keep proposing alternative mean and variance values for the cdf of normal distribution for Cy5 dye response function until the resulting distance between the target curve and the proposed curve is minimized. A general-purpose optimization method based on Nelder-Mead (or quasi-Newton or conjugate-gradient algorithms) could be used here to obtain the "optimal" cdf of normal distribution for Cy5 dye response function.

4. For the observed Cy3 channel microarray data, we transform it back to the original data via the inverse of the cdf of the normal distribution for the Cy3 dye response function (i.e. Equation (3.15)). For the observed Cy5 channel microarray data, we transform it back to the original data via the inverse of the optimal cdf of normal distribution for the Cy5 dye response function.

In the above, we calculate the so called "distance" between two curves in a two dimensional plane. The "distance" is defined as following: assuming that there are two curves A and B in a two dimensional space and each of the curves is approximately determined by n points which are evenly distributed along the curve, that is, for A we have points $a_1,\cdots,a_n$ and for B we have points $b_1,\cdots,b_n$,

then the distance between A and B is defined as

$$\sum_{i=1}^{n} \min_{j \in N} \{|b_i - a_j|\}, \quad N = \{1, \ldots, n\}, \tag{3.19}$$

where $|b_i - a_j|$ denotes the Euclidean distance between the two points b_i and a_j.

The optimization method we implement here is Nelder and Mead (1965). The Nelder-Mead method is a commonly used nonlinear optimization algorithm. It is a simplex method for finding a local minimum of a objective function of several variables. In our case, the objective function is the distance between the proposed cdf of normal distribution and the target cdf of normal distribution and there are two unknown variables: mean and variance of the proposed cdf. For two variables, a simplex (a generalized triangle in N dimensions) is a triangle, and the method is a pattern search that compares function values at the three vertices of a triangle. The worst vertex, whose function value is the largest, is rejected and replaced with a new vertex. A new triangle is formed and the search is continued. The process generates a sequence of triangles (which might have different shapes), for which the function values at the vertices get smaller and smaller. The size of the triangles is reduced and the coordinates of the minimum point are found. In `R`, the function `optim` provides us with a variety of optimization methods including Nelder-Mead which is the default method.

3.3 Results

In this section, we study the performance of model two.

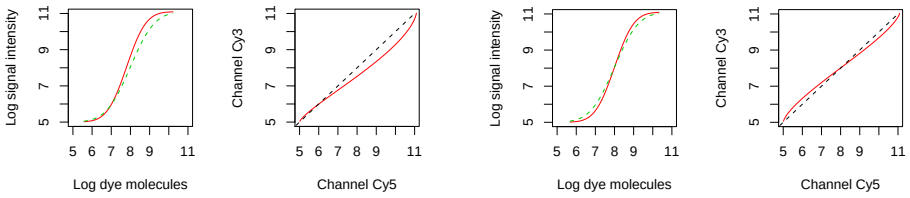
3.3.1 Evaluating the model

We try a variety of combinations of mean and variance values for the cdf of Cy5 dye response function to evaluate our model. Figure 3.7 shows that the model of dye response function discussed in Section 3.2.1.2 is flexible enough to generate different typical dye effect patterns. In general, it is better than the simple model in Section 3.2.1.1, because it can generate much smoother dye effect curves and the resulting nonlinear dye effect patterns (e.g. Figure 3.7 (a), (d) and (g)) strongly resemble the banana effect which we often see from real microarray experimental data.

3.3.2 Evaluating the method

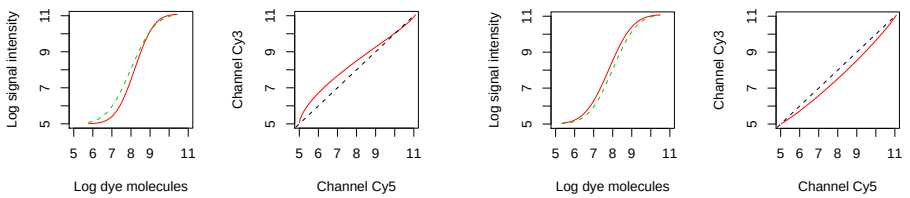
Besides the dye response model, we are also interested in knowing the performance of the method. One way to evaluate it is to use the cdf of normal distribution for dye response function with known parameters as input information to test the optimization method of the algorithm. If the estimation of parameters in the "optimal" cdf of normal distribution for dye response function matches the input prespecified parameters, then it would mean that the method works well.

Here we use three cases from Figure 3.7 as the examples. The prespecified mean and variance parameter values are 7.8 and 0.8, 7.8 and 1.0, and 7.8 and 1.2 for the Cy5 dye response function (red curve) in Figure 3.7 (a), (d) and (g) respectively. For all these cases, the parameters have been estimated by the optimization method and the results turn out to be exactly the same as these prespecified ones. That isn't a surprise, because in essence our proposed algorithm is just an inverse calculation of the generation of the dye effect pattern from two dye response functions.



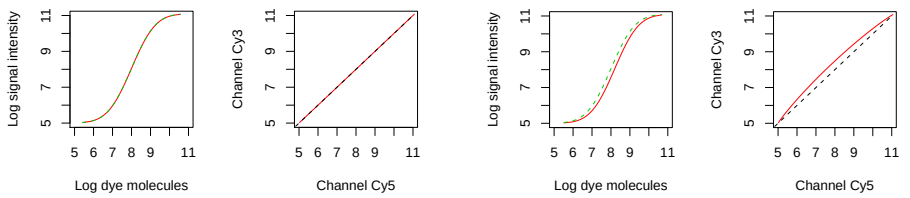
(a) $\mu = 7.8$, $\sigma = 0.8$

(b) $\mu = 8.0$, $\sigma = 0.8$



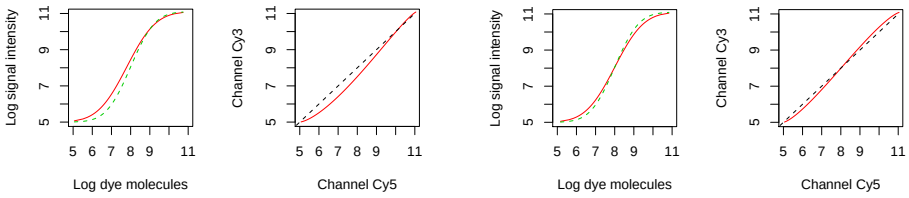
(c) $\mu = 8.2$, $\sigma = 0.8$

(d) $\mu = 7.8$, $\sigma = 1.0$

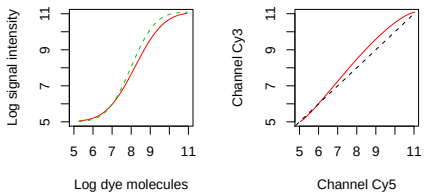


(e) $\mu = 8.0$, $\sigma = 1.0$

(f) $\mu = 8.2$, $\sigma = 1.0$



(g) $\mu = 7.8$, $\sigma = 1.2$ (h) $\mu = 8.0$, $\sigma = 1.2$



(i) $\mu = 8.2$, $\sigma = 1.2$

Figure 3.7: The difference between the Cy3 dye response function (the cdf of normal distribution with mean 8 and variance 1) and the Cy5 dye response function (the cdf of normal distribution with different combinations of mean and variance value) results in a variety of dye effect patterns. For each figure, the left subfigure shows the two dye response functions (the green dash curve is for Cy3 and the red curve is for Cy5) and the right subfigure shows the dye effect curve (red curve).

Then we can take further ways to study the performance of the method versus that of LOESS. The basic idea is: firstly, for each channel, we make up observed gene expression data by transforming true data via Equation (3.13); secondly, apply the new method and LOESS method respectively to the transformed data and get the resulting normalized data; thirdly, compare the results with the original data so as to evaluate the two methods.

Let's consider a more detailed scenario as follows. We assume that in a microarray experiment we have a number of genes, let's say 500, whose relative rank between Cy3 and Cy5 channels among all the values is relatively stable. We assume the true gene expression (after taking logarithm) for these 500 genes is normally distributed with mean 8 and variance 1. The generated values bigger than 11.1 are set to be 11.1 and the values smaller than 1 are set to 1. The dye response function for Cy3 is fixed and assumed to be the cdf of normal distribution with $\mu = 8$ and $\sigma = 1$. The Cy5 dye response function is the cdf of a normal distribution with unknown μ and σ.

Once we propose the mean and variance values for the Cy5 dye response function, we are able to generate 500 observed Cy3 and Cy5 gene expression data (contain dye effect) by transforming the true gene expression data via the Cy3 and Cy5 dye response functions respectively. Note that before the transformation, we add some small variation ε to the true expression data, where $\varepsilon \sim N(0, \sigma_\varepsilon^2)$. It makes possible that the resulting observed gene expression data in the Cy3 vs Cy5 scatter-plot do not overlap on the dye effect curve but distribute around it. Figure 3.8 shows such an example of simulated data when mean and variance of Cy5 dye response function is set to be 7.7 and 1.1, and σ_ε, the standard deviation of the error, be 0.1.

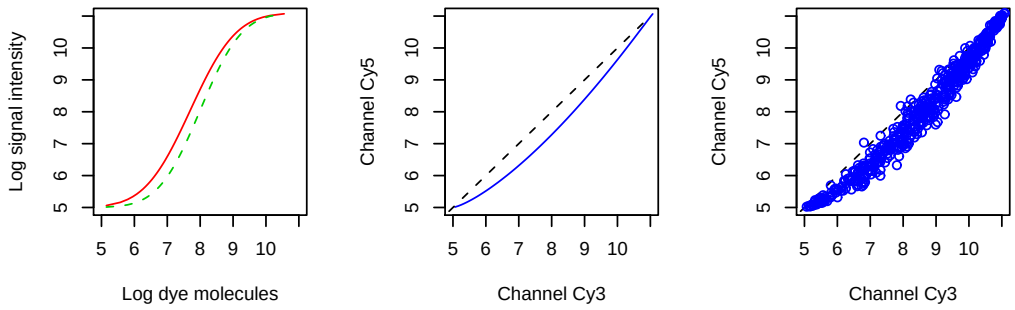


Figure 3.8: An example of gene expression simulation when the mean and variance of Cy5 dye response function is set to be 7.7 and 1.1, the mean and variance of Cy3 dye response function is fixed to be 8 and 1, and the standard deviation of variation, σ_ε, is set to be 0.1. The left figure shows Cy3 dye response function (green dot curve) and Cy5 dye response function (red curve); The middle figure shows the resulting dye effect pattern (blue curve); The right figure shows the simulated 500 gene expressions (blue points) scattering around the dye effect curve, which is like the "banana effect" from practical experiment.

Then we can apply our method and LOESS to the simulated gene expression data respectively. Each of the methods does the dye normalization and gives out its corresponding reconstructed gene expression data. Since we know the original microarray data in advance, we are able to evaluate the performance of these two methods quantitatively by calculating the sum of squares of the difference between the reconstructed data and the original data for each channel. The method with low sum of squares of the difference is preferable.

In order to compare the methods properly, we should transform all the data before comparison by standardization,

$$z = \frac{g - \mathrm{E}(G)}{\sqrt{\mathrm{Var}(G)}} \tag{3.20}$$

Table 3.2: Comparison of LOESS and the new method. In the scenario of the example in the Figure 3.8, the new method has smaller amount of the sum of the squares of the difference between the normalized reconstructed data and original data than LOESS in both of the Cy3 and Cy5 channels.

Channel	New method	LOESS
Cy3	5.67	22.17
Cy5	5.14	23.70

where the reconstructed gene expression data is assumed to be a random variable G, g is an observation of G and z is the corresponding standardized observation.

For the example in Figure 3.8, Table 3.2 shows that the new method seems to be better than LOESS. Figure 3.9 (b) and (c) show the result from the new method and LOESS. Figure 3.9 (d) and (e) show the comparison of the normalized result from the new method and LOESS in Cy3 and Cy5 channel respectively. It shows that the result from LOESS not only tends to be an underestimate when the original gene expression data is close to the upper-limit of its possible range, but also be an overestimate when the original gene expression data is close to the lower-limit of its possible range. In contrast, the new method has no such drawback.

Figure 3.10 considers a variety of combinations of mean and variance values for the Cy5 dye response function and σ_ε to evaluate the performance of the new method and LOESS by computing the sum of squares of the difference between the standardized reconstructed data and standardized original data for Cy3 and Cy5 respectively. The results unanimously show that the reconstructed data by the new method has much smaller sum of square of the difference than that by LOESS, which means the new method has much better performance than LOESS

for simulated gene expression data from the dye response model.

Figure 3.11 gives the reconstructed gene expression result from real microarray gene expression data on skin study by using the new method and LOESS respectively. Basically, from the figure we see that generally the performance of new method is comparable to that of LOESS, although it is impossible for us to judge quantitatively.

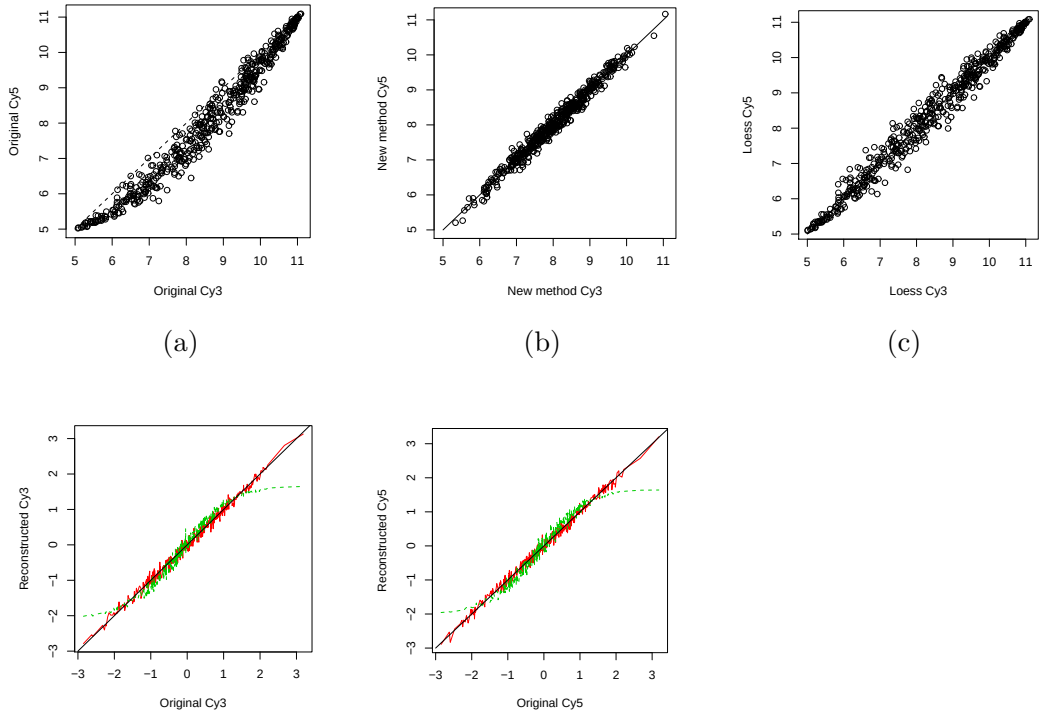
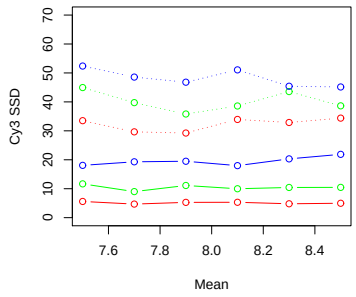
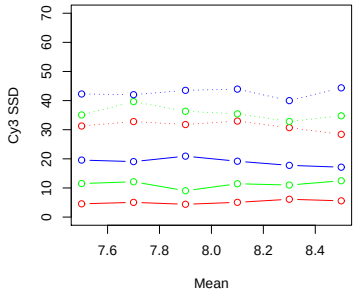
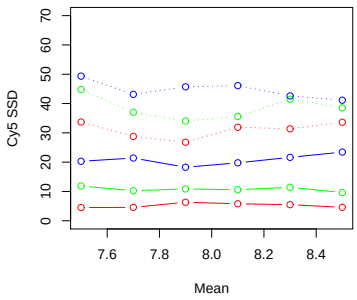
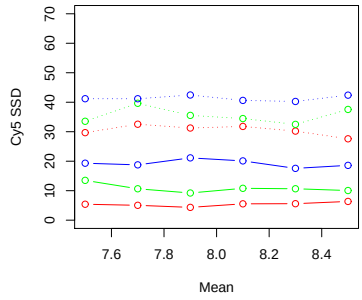


Figure 3.9: An example of comparison of the performance of the new method and LOESS method. The input Cy3 and Cy5 gene expression data (with dye effect) is simulated from the example shown in Figure 3.8. (a) the scatter plot of original Cy3 and Cy5 data with dye effect; (b) the scatter plot of the reconstructed Cy3 and Cy5 data from the new method; (c) the scatter plot of the reconstructed Cy3 and Cy5 data from LOESS method; (d) the comparison between the standardized original Cy3 data and the standardized reconstructed Cy3 data, the green curve stands for LOESS method and the red curve stands for the new method; (e) the comparison between the standardized original Cy5 data and the standardized reconstructed Cy5 data, the green curve stands for LOESS method and the red curve stands for the new method.

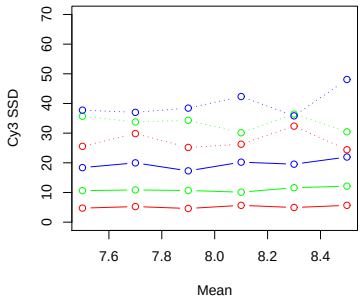

Cy3 SSD
0 10 20 30 40 50 60 70
7.6 7.8 8.0 8.2 8.4
Mean

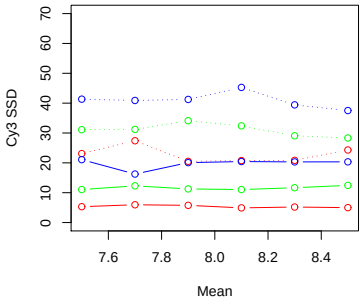

Cy3 SSD
0 10 20 30 40 50 60 70
7.6 7.8 8.0 8.2 8.4
Mean

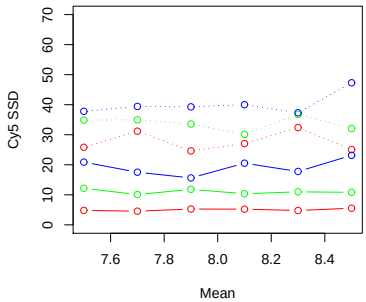

Cy5 SSD
0 10 20 30 40 50 60 70
7.6 7.8 8.0 8.2 8.4
Mean

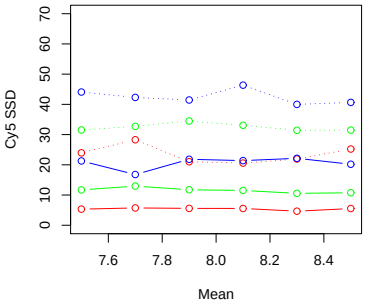

Cy5 SSD
0 10 20 30 40 50 60 70
7.6 7.8 8.0 8.2 8.4
Mean

(a) (b)

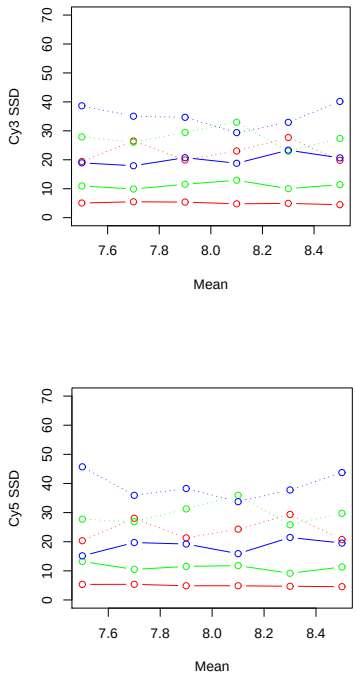

Cy3 SSD
0 10 20 30 40 50 60 70
7.6 7.8 8.0 8.2 8.4
Mean


Cy3 SSD
0 10 20 30 40 50 60 70
7.6 7.8 8.0 8.2 8.4
Mean


Cy5 SSD
0 10 20 30 40 50 60 70
7.6 7.8 8.0 8.2 8.4
Mean


Cy5 SSD
0 10 20 30 40 50 60 70
7.6 7.8 8.0 8.2 8.4
Mean

(c) (d)



(e)

Figure 3.10: Comparison of the performance of the new method and LOESS method for a variety of scenarios. Subfigure (a), (b), (c), (d) and (e) show the sum of square of the difference between the standardized reconstructed data (line for new method, dotted line for LOESS) and standardized original data for Cy3 and Cy5 when the variance for Cy5 dye response function is set to be 0.8, 0.9, 1.0, 1.1 and 1.2 respectively, and for each subfigure it considers the performance in the case of different mean value for Cy5 dye response (i.e. 7.5, 7.7, 7.9, 8.1, 8.3 and 8.5) and different standard deviation of the error added to the true gene expression (red, green and blue color corresponds to 0.1, 0.15 and 0.2 respectively).

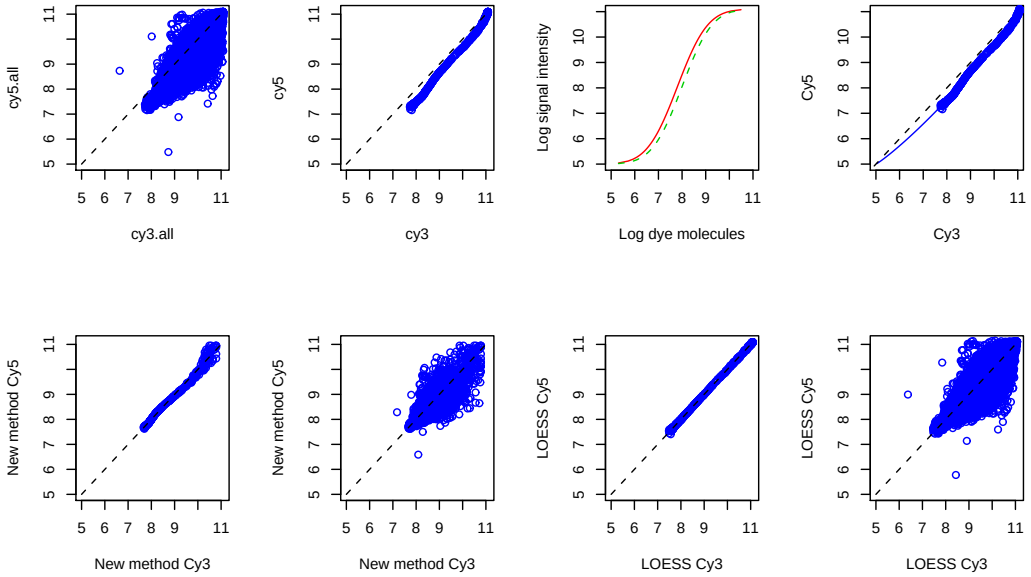


Figure 3.11: An example of dye effect normalization using real skin microarray gene expression data from experiment. Up-left subfigure shows the scatter-plot of all the gene expression in Cy3 and Cy5 channel; up-middle-left sub-figure shows the the scatter-plot of all the gene expression whose ranking is not changed more than 500 in Cy3 and Cy5 channel; up-middle-right and up-right subfigures show the resulting two-channel dye response models and also the dye effect curve respectively; down-left and down-middle-left subfigures show the reconstructed gene expression (not changed more than 500 and all the data) data by the new method respectively; down-middle-right and down-right subfigures show the reconstructed gene expression (not changed more than 500 and all the data) data by LOESS respectively.

3.4 Discussion

In this chapter, we first give a review of the background and recent development in dye effect normalization, especially the two-step intensity-dependent dye normalization method (i.e. LOESS method). In order to know the problem of dye effect normalization better, we suggest the concept of dye response function (model) and dye effect curve and study the causal relationship between these two issues. We propose two kinds of dye response model, one is linear and the other is nonlinear. We argue that the "S" shape nonlinear one is more reasonable, and propose to model it by the cdf of normal distribution (probit function). (Note that it is possible to use other mathematical models to represent the "S" shape curve, for example logistic distribution and inverse tangent function). Based on this model we develop our new method. The main idea of our method is to determine such a pair of dye response functions that the resulting dye effect curve matches the dye effect curve from the observed gene expression data. After specifying the pair of dye response functions we can use them to transform the observed gene expression intensity to true intensity. In fact it is equivalent to the calculation of the inverse of cdf of normal distribution. Finally, our method is compared to the LOESS method using simulated gene expression data and experimental gene expression data. In the case of simulated data, the performance of our method is better than that of LOESS method. It is anticipated because our method for dye normalization is just an inverse calculation of our method for data simulation so we do not expect LOESS method can outperform our method (although we add different levels of variation to the simulated data). In the case of experimental data, by comparing the original Cy3 vs Cy5 scatterplot and the dye-normalized Cy3 vs Cy5 scatterplots from the two methods visually, we can find that our

method is comparable to the LOESS method although it is unfortunate that we can not measure the performance quantitatively because we have no information of the true gene expression data. Further, compared to other popular method (e.g. LOESS) a strong merit of our method is that it is based on reasonable dye response model, not like LOESS method which has no good motivation and is purely an application of a general smoothing method.

Chapter 4

Estimating the proportion of true nulls

4.1 Introduction

With the rapid development of microarray technologies, scientists are able to take measurements of expression levels of thousands of genes in different conditions (e.g. treatment or control) simultaneously. Among the applications of microarray experiments, a very common one is to find out which differentially expressed genes to subject to further experimentation. Hypothesis testing is usually used for the identification of differentially expressed genes. For each statistical test performed, there is some probability that an erroneous inference will be made. If thousands of tests are performed, quite a number of incorrect inferences might occur just by chance alone. The need then arises to properly account for the occurrence of errors in applications that involve multiple testing. Until recently, statistical procedures devoted to this multiple testing problem mostly have focused on controlling or estimating false positive error criteria (Pounds, 2005).

For microarray experiments, the most used criterion nowadays is the false discovery rate (FDR) which is the expected proportion of false discoveries among all discoveries (Benjamini and Hochberg, 1995). Besides it, there are some other popular error rates such as positive false discovery rate (pFDR) (Storey, 2002) and local false discovery rate (lFDR) (Efron et al., 2001). In order to access or control these multiple error rates, we must estimate the proportion π_0 of true null hypotheses (i.e. the proportion of not differentially expressed genes) properly. Many statistical methods have been proposed to estimate the proportion of true nulls, among others Schweder and Spjøtvoll (1982), Allison et al. (2002), Storey and Tibshirani (2003), Pounds and Cheng (2004), Pounds and Cheng (2004), Liao et al. (2004), Dalmasso et al. (2005), Langaas and Lindqvist (2005), McLachlan et al. (2006) and Lai (2007). Other methods for selecting differentially expressed genes in microarray experiments produce an estimate of π_0 as a by-product, (Cox and Wong (2004); Lönnstedt and Speed (2002); Newton et al. (2001, 2004); Smyth (2004)). In this chapter, we focus on estimating π_0 on the basis of calculated p-values from hypothesis tests, by using mixture models with unknown number of components. We then apply the estimate of π_0 to the computation of pFDR and local FDR for real datasets.

The outline of this chapter is as follows. In the first section we introduce the general aim and structure. In the second section we review the definitions and controlling procedures of different error rates for multiple hypothesis testing in microarray experiments, and give an example to compare some of these error rates. In the third section we first review the two-component mixture model and the proportion of true null hypotheses and then review recently published methods of estimating the proportion of true null. In the fourth section we propose

three different mixture models with an unknown number of components for describing the distribution of p-values produced from the microarray experiments. In the fifth section we review the approach of Nobile and Fearnside (2007), a Markov chain Monte Carlo method for the Bayesian analysis of finite mixture distributions with an unknown number of components. In order to apply the MCMC method to our models, for each of the three mixture model we derive the corresponding explicit expression of the joint posterior distribution of the number of components and the allocation variables by integrating out the component parameters and mixture weights. In the sixth section, we illustrate our models with real and simulated gene expression data. The proportion of true null hypotheses, pFDR and lFDR are estimated and we show that lFDR gives more specific and relevant quantification of the evidence for differential expression that can be substantially different from pFDR. The final section contains a discussion.

4.2 Multiple hypothesis testing and error rates

4.2.1 Classical hypothesis testing

Statistical hypothesis testing is a formal means of distinguishing between one hypothesis concerning the parameters of the distribution of a population (the null hypothesis often denoted by H_0) against another (the alternative hypothesis often denoted by H_1). For instance, in a test concerning the value of an unknown parameter, the null hypothesis specifies a particular value for the parameter, whereas the alternative hypothesis specifies either an alternative value or a range of alternative values. The role of these two hypotheses is asymmetric, with the null hypothesis assumed as true until enough evidence to the contrary has been collected.

From the null hypothesis, a reference distribution of a test statistic (such as a t-statistic) can be derived and the resulting distribution is called 'the null distribution' which describes the variability of that statistic due to chance. By comparing the test statistic on the actual data to the null distribution, a p-value is computed to summarize the comparison. The p-value is the probability of observing a value for the test statistic that is at least as extreme as the observed test statistic under the assumption that null hypothesis is true. If the actual value of the statistic is too far from its expected value, which corresponds to a very small p-value, the test is deemed to be significant and the decision is to reject H_0 in favor of the alternative hypothesis. Otherwise, the test is deemed to be not significant and the decision is to not reject H_0. The set of values of the statistic that lead to the rejection of H_0 is called critical region or rejection region

and the set of values that do not lead to rejection of H_0 is called the acceptance region.

There are two cases when the test leads to a correct result. These occur when H_0 is true and the test leads to its acceptance and when H_1 is true and the test leads to rejection of H_0. On the other hand there are two cases when the test leads to an incorrect result. These occur when H_0 is true but the test leads to its rejection (a Type I error or false positive) and when H_1 is true but the test leads to the acceptance of H_0 (a Type II error or false negative). The probability of making a Type I error is denoted by α. It is also the significance level of the test, which determines the size of the critical region. The smaller the significance level, the smaller the critical region. The probability of making a Type II error is denoted by β. The power of the test, which is the probability of accepting the alternative hypothesis when it is in fact true, is $1 - \beta$.

In a microarray experiment, we usually like to know whether or not a gene is differentially expressed and we are interested in the parameter θ_j, which is the population mean difference in gene expression for gene j. Therefore, for m genes we have m pairs of mutually exclusive hypotheses:

$$H_{0j}: \ \theta_j = 0, \ \text{ gene } j \text{ is not differentially expressed}$$
$$H_{1j}: \ \theta_j \neq 0, \ \text{ gene } j \text{ is differentially expressed.}$$

When only a single pair of hypotheses is to be tested, the probability of each type of erroneous inference can be limited to desired levels by carefully planning the experiment and the statistical analysis. In this simple setting, the probability of a false positive can be limited by preselecting the significance level. The probability of a false negative can be limited by performing an experiment with adequate replication. Statistical power calculations can determine how much

replication is required to achieve a desired level of control on the probability of a false negative result. When multiple hypothesis tests are performed simultaneously, which is often the case in a microarray experiment, the situation is more complicated.

4.2.2 Multiple hypothesis testing

The problem of multiple testing can be described as the potential increase of false positive (i.e. Type I errors) that occurs when many statistical hypotheses are tested and each test has a specified Type I error probability. In the microarray setting, that means: " a p-value of 0.001 for one gene among a list of several thousands will no longer correspond to very few significant findings, as it is inevitable that such small p-value will occur by chance when considering a large enough set of genes. " (Dudoit et al., 2002)

To understand this problem better, consider m independent tests performed all at the per-comparison level α_C (i.e. the probability of making a Type I error). The corresponding family-wise significance level α_F (i.e. the probability of making at least one Type I error) is given by $\alpha_F = 1 - (1 - \alpha_C)^m$. The larger the number of tests, the closer α_F to 1. Thus, controlling α_F to a small value, say 0.05, will require an extremely small per-comparison level α_C. Therefore, in order to retain the desired overall rate of false positives (rather than a higher rate) in a experiment involving more than one test, the standard for each test must be more stringent. However, reducing the threshold of significance may substantially increase the number of false negatives. Therefore, choosing the p-value threshold used to determine statistical significance is a delicate problem that requires very careful attention. Additionally, the results must be appropriately interpreted after the significance threshold is chosen.

4.2.3 Error rates for multiple testing

Consider the situation of testing simultaneously m pairs of hypotheses H_{0j} and H_{1j}, $j = 1, \ldots, m$. The problem can be described by Table 4.1. The specific m hypotheses are assumed to be known in advance, the numbers m_0 and $m_1 = m - m_0$ of true and false null hypotheses are unknown parameters. The number of rejected hypotheses S is an observable random variable, and T_N, F_P, F_N and T_P are all unobservable random variables. F_P is the number of false positives, T_P is the number of true positives, T_N is the number of true negatives and F_N is the number of false negatives. In general, one would like to minimize the number F_P of false positives or Type I errors and the number F_N of false negatives or Type II errors. The standard way in a univariate setting is to pre-specify an acceptable Type I error rate α and seek tests which minimize the Type II error rate, i.e. maximize power, within the class of tests with Type I error rate α. In the multiple testing situation like a microarray experiment, it is no longer suitable to use the original Type I error rate any more. Therefore, statisticians have defined some other kinds of error rates, such as false positive rate (FPR), family-wise error rate (FWER), false discovery rate (FDR), positive false discovery rate (pFDR) and other FDRs like conditional FDR (cFDR), marginal FDR (mFDR) and local FDR (lFDR) to measure the occurrence of erroneous inferences when determining which results should be considered statistically significant. In the following sections we focus on the review of these error rates in the analysis of microarray experiment.

Table 4.1: Outcomes from m hypothesis tests. All the random quantities T_N, F_P, F_N and T_P depend on the data and the pre-specified level α.

	H_0 accepted	H_0 rejected	Total
H_0 true	T_N	F_P	m_0
H_0 false	F_N	T_P	m_1
Total	$m - S$	S	m

4.2.3.1 False positive rate (FPR)

The false positive rate (FPR) is the proportion of the number of true null hypotheses that were erroneously judged as being positive:

$$\text{FPR} = \frac{F_P}{m_0}. \tag{4.1}$$

Most traditional methods focus on controlling FPR. This is equivalent to saying the false positive rate is equal to the significance level.

4.2.3.2 Family-wise error rate (FWER)

The family-wise error rate is the probability that among all those genes that are truly not differential expressed at least one is incorrectly declared as differential expressed (e.g. making at least one Type I error among all hypotheses (Hochberg and Tamhane, 1987)), regardless of the number of genes tested,

$$\text{FWER} = \Pr(F_P \geq 1). \tag{4.2}$$

A similar, but less stringent, error rate is the generalized familywise error rate k-FWER, which is defined to be the probability of at least k Type I errors.

Generally, FWER is a very conservative error rate. Especially, with a large

number of hypotheses, it is typically impractical to insist that the probability of making even only one false rejection should be small. The FWER approach tends also to have low power as it tends to screen out all but a handful of genes that show extreme differential expressions.

4.2.3.3 FWER controlling procedures

Two main methods for controlling FWER are often used in practice. The simplest one is the well-known Bonferroni correction which guarantees that FWER $\leq \alpha$ by declaring all genes with p-values less than α/m differentially expressed. In other words, the procedure determines the actual Type I error rate for each hypothesis test as the ratio of the desired FWER level α and the number of tests. For example, to control the FWER at a 0.05 level with 5,000 hypothesis tests, a rejection cut-off of 0.00001 for each individual gene's p-value is required. Bonferroni correction is a single-step FWER control method which means that all of the p-values are tested against the same cut-off level.

The other method is Hochberg's procedure (Hochberg, 1988) which is a step-down method for controlling FWER. The procedure guarantees that the FWER is less than or equal to α.

1. Let $P_{(j)}$ be the j-th order statistic of the p-values, for $j = 1, \ldots, m$.

2. If $P_{(m)} < \alpha$, then reject all H_j, for $j = 1, 2, ..., m$, where H_j is the null hypothesis associated with the gene with the jth smallest p-value; If $P_{(m)} \geq \alpha$, then H_m can not be rejected and one has to go on to compare $P_{(m-1)}$ with $\alpha/2$.

3. If $P_{(m-1)} < \alpha/2$, then all H_j are rejected, for $j = 1, 2, ..., m-1$. If this is not the case, then $P_{(m-1)}$ can not be rejected.

4. Continue to compare $P_{(m-2)}$ with $\alpha/3$, and so on until the smallest i such that $P_{(m-i)} < \alpha/(1+i)$. Then reject all H_j, for $j = 1, 2, ..., m-i$.

Although both procedures control FWER, the size of the sets of genes rejected by the methods can vary greatly. The difference is that these methods have different level of power. In the microarray setting, power is the expected proportion of truly differential expressed genes that are correctly identified as being differential expressed, that is, power $= E\big[1 - \frac{F_N}{m_1}\big] = E[\frac{T_P}{m_1}]$. Though Bonferroni correction has the advantage of being simple to implement, it comes at the cost of reduced power as it will fail to reject many truly differentially expressed genes. Hochberg's procedure has greater power than Bonferroni's single step procedure, because it gains power by only subjecting the smallest p-value, P_1, to the single-step level test (e.g. α/m); larger P-values are subject to progressively less stringent bounds. However, this feature might not lead to any more genes being discovered in many practical microarray experiments, since when m is very large and j is very small, the cut-off by Hochberg's procedure is not very different from that of Bonferroni, α/m.

There are some other similar procedures for controlling FWER like Šidák's method which is a single-step method and Holm's method (Holm, 1979). More details can be found in Dudoit et al. (2002).

For controlling k-FWER, see Dudoit et al. (2004) who propose some procedures for it.

4.2.3.4 False discovery rate (FDR)

Benjamini and Hochberg (1995) introduce a different multiple hypothesis testing error measure called the false discovery rate (FDR). The quantity is the expected proportion of false positive findings among all the rejected hypotheses times the

probability of making at least one rejection,

$$\text{FDR} = E\left[\frac{F_P}{S}\middle| S > 0\right] \Pr(S > 0). \tag{4.3}$$

FDR offers a much less strict multiple testing criterion than FWER. Since FDR is more relevant than FWER in large-scale hypotheses generating studies, it is now widely recognized as a useful measure of the false positives in microarray experiments.

4.2.3.5 FDR controlling procedures

The aim of multiple testing procedures for control of FDR is to determine a threshold for significance in such a way that the false discovery rate is limited to being less than or equal to a prespecified level of tolerance. For example, by deciding to accept a FDR of 5% for a microarray experiment, a FDR procedure will find the largest subset of genes to be classed as differentially expressed that has an expected percentage of not differentially expressed genes of 5%.

After introducing the FDR as a useful error rate for multiple testing, Benjamini and Hochberg (1995) also propose a method (we call Benjamini and Hochberg's FDR procedure) that operates on p-values to control the FDR at a prespecified level. It is a step-up method and works as follows.

1. Let $P_{(j)}$ be the j-th order statistic of the p-values, for $j = 1, \ldots, m$.

2. Determine a threshold value for rejection by finding the largest integer j such that $P_{(j)} \leq j\alpha/m$, where α is the desired FDR level.

3. Reject any hypothesis whose p-value is smaller than or equal to $P_{(j)}$. Therefore j is the number of the total rejections from the hypothesis tests.

This approach has been mathematically proven to ensure that FDR $\leq \pi_0\alpha$ if the p-value under the true null hypotheses (i.e. genes that are truly not differentially expressed) are statistically independent and uniformly distributed over the interval $[0, 1]$. Note that π_0 is the proportion of true null hypotheses. In order to control FDR precisely, π_0 is required. However, this proportion is not really known, therefore π_0 is replaced by 1 to guarantee that the FDR is controlled conservatively. As this procedure is quite conservative, it is possible to develop a method that finds more significant results and still controls the FDR at the prespecified level.

Benjamini and Hochberg (2000) introduce another method for adapted FDR control. For a set of observed p-values, if the Benjamini and Hochberg's FDR procedure declares any results significant, then the null proportion π_0 is estimated to adjust the results that may lead to additional significant findings. However, this method offers limited power gain over the original Benjamini and Hochberg's FDR procedure, because the estimate of π_0 is very conservative (Hseuh et al., 2003).

4.2.3.6 A simple example for FPR, FWER and FDR

In this section, we use a well-known ALL/AML leukaemia dataset as an example to illustrate and compare the error rates and controlling procedures described so far.

Golub et al. (1999) are interested in identifying genes that are differentially expressed in patients with two types of leukaemia, acute lymphoblastic leukaemia (ALL) and acute myeloid leukaemia (AML). Gene expression levels are measured using Affymetrix high-density oligonucleotide chips. The learning set comprises 38 samples, 27 ALL cases and 11 AML cases (data available at `http://www.genome`

.wi.mit.edu/MPR). For the purpose of simplicity, we use the data as provided in the R package, multtest, which can be downloaded from http://www.bioconductor.org. The data has already been pre-processed and is summarized by a 3051×38 matrix $X = (x_{ji})$, where x_{ji} denotes the expression level for gene j in tumor mRNA sample i.

For these data, two-sample Welch t-statistics are calculated for each gene, along with their p-values. The different procedures at different α levels are then applied to these p-values, and the numbers of genes rejected by each combination of error rate (procedure) and α levels are compared in Table 4.2.

From Table 4.2, we find that using FPR will result in finding a very large number of active genes. If we choose to look at the results when we control the error rates at 5%, the FPR procedure declares 1164 out of 3051 genes tested are differentially expressed. Even if no gene is actually differentially expressed (i.e. $m_1 = 0$, $m_0 = m$), we would expect around 152 positive genes which are all false positive.

Using either of the two FWER controlling procedures (the Bonferroni and Hochberg) leads to very similar numbers of positive genes. The reason for it has been discussed in section 4.2.3.3. Since FWER is a very strict error rate, it is unlikely for us to falsely class any inactive genes as "differential expressed" by using it, but on the other hand, it would also make us overlook many truly differential genes.

Table 4.2 also shows the number of genes declared active for the Benjamini and Hochberg's FDR procedure controlled at different levels. For example, if we control FDR at 5%, then we expect that around $883 \times 0.05 \approx 44$ genes selected as differential will be actually inactive.

Table 4.2: Comparison of numbers of rejected genes by using different error rates in the leukaemia experiment.

Error rate and its controlling procedure	Desired α level			
	0.005	0.01	0.05	0.1
FPR	686	815	1164	1392
FWER (Bonferroni)	153	169	228	258
FWER (Hochberg)	154	170	233	261
FDR (Benjamini and Hochberg)	482	569	883	1063

4.2.3.7 Positive false discovery rate (pFDR)

As an alternative to FDR, positive false discovery rate (pFDR) is initially mentioned in Benjamini and Hochberg (1995) and later thoroughly studied by Storey (2002),

$$\text{pFDR} = E\left[\frac{F_P}{S}\middle| S > 0\right]. \tag{4.4}$$

The term "positive" reflects the fact that it conditions on the event that positive findings have occurred.

The definition of pFDR is motivated by concerns about what happens when $\Pr(S > 0)$ is much less than 1, in which case FDR might be misleading. Conceptually, pFDR is more sound than FDR. But for microarray data with a large m and many differentially expressed genes, the difference between pFDR and FDR is generally small as the extra factor in FDR, $\Pr(S > 0)$ is very close to 1.

Storey (2003) proposes that the pFDR at p is the probability that H_{0j} being true (i.e. gene j is not differentially expressed) conditional upon its p-value P_j being less than or equal to p, that is,

$$\text{pFDR}(p) \equiv \Pr(H_{0j} \text{ being true} | P_j \leq p) = \frac{\pi_0 p}{F(p)}. \tag{4.5}$$

where π_0 is the proportion of true nulls and $F(p)$ is the proportion of hypothesis testings with p-value less than p.

Storey (2002) defines the q-value as a pFDR analogue of p-value. The q-value gives a hypothesis testing error measure for each observed statistic with respect to pFDR just like the p-value to type I error and the adjusted p-value to FWER. Storey proposes q-value as

$$\mathrm{q}(P_{(i)}) = \min_{j \geq i} \left[\mathrm{pFDR}(P_{(j)}) \right]$$

for $i = 1, 2, ..., m$, where $P_{(1)} \leq P_{(2)} \leq ... \leq P_{(m)}$ are the ordered observed p-values. This definition ensures that $\mathrm{q}(P_{(1)}) \leq ... \leq \mathrm{q}(P_{(m)})$. The $\mathrm{q}(P_{(i)})$ gives us the minimum pFDR that we can achieve for rejection regions containing $[0, P_{(i)}]$ for $i = 1, ..., m$. In other words, for each p-value there is a rejection region with pFDR equal to $\mathrm{q}(P_{(i)})$ so that at least $P_{(1)}, ..., P_{(i)}$ are rejected.

The q-value is appealing because it gives a measure of significance that can be attached to each gene, but it must be stressed that it is not an estimate of the probability for the gene to be a false positive. The q-value is generally lower than the latter because it is computed using all the genes that are more significant than gene i. Obviously a gene whose p-value is near to the threshold $P_{(i)}$ does not have the same probability to be differentially expressed than a gene whose p-value is close to zero. Hence the q-value gives a too optimistic view of the probability for the gene to be a false positive. Therefore it is important to obtain an estimate of the FDR attached to each gene, called Local FDR. See Section 4.2.3.8 for more details of local FDR (lFDR).

4.2.3.8 Local FDR (lFDR)

Besides pFDR, Benjamini and Hochberg (1995) also mention other two alternative FDR error measures: Conditional FDR (cFDR) and marginal FDR (mFDR).

The cFDR is the FDR conditional on the observed number of rejections $S = s$,

$$\text{cFDR} = E\left[\frac{F_P}{S}\middle| S = s\right] = \frac{E\left[F_P \mid S = s\right]}{s}, \tag{4.6}$$

provided that $s > 0$, and cFDR $= 0$, for $s = 0$.

The marginal FDR (mFDR) is the ratio of the expectation of F_P to the expectation of S,

$$\text{mFDR} = \frac{E[F_P]}{E[S]}. \tag{4.7}$$

FDR, pFDR, cFDR and mFDR provide general information about a group of genes. But if we are actually interested in specific evidence for each gene, which kind of error rate should we use? The local false discovery rate (lFDR) is proposed by Efron et al. (2001) in a mixture model framework for this purpose. The lFDR at p is defined as the probability that gene j is not differentially expressed (i.e. H_{0j} is true) conditional upon its p-value P_j being equal to p, that is

$$\text{lFDR}(p) \equiv \Pr(H_{0j} \text{ being true} | P_j = p) = \frac{\pi_0}{f(p)}. \tag{4.8}$$

where $f(p)$ is the density of the p-values, which can be considered as a two component mixture with weights π_0 and $1 - \pi_0$: the $\text{Un}(0, 1)$ distribution under H_0 and an unknown distribution under the alternative H_1. From Equation (4.5) and Equation (4.8), we can find a simple relationship between pFDR and lFDR,

that is,

$$\text{pFDR}(p) = E_f(\text{lFDR}(P_j)|0 < P_j < p),$$

which is called the averaging theorem by Efron and Tibshirani (2002).

4.3 The mixture model and the estimate of the proportion of true nulls

4.3.1 The two-component mixture model for the distribution of the test statistic

Efron et al. (2001) first proposed a two-component mixture model for the distribution of the test statistic in the microarray multiple testing setting. The model is motivated as follows: let z_j be 1 if the jth gene expresses differentially and 0 if it does not. It is natural to model z_j, $j = 1, ..., m$, as Bernoulli trials with probability $1 - \pi_0$, where $\pi_0 = m_0/m$. Let T_j be a test statistic for testing hypothesis H_j, f_0 be the density of T_j distribution given $z_j = 0$ and f_1 be the density of T_j distribution given $z_j = 1$. The probability density function (pdf) of a test statistics T_i, $i = 1, ..., m$ is then a two-component mixture,

$$f(t) = \pi_0 f_0(t) + (1 - \pi_0) f_1(t), \tag{4.9}$$

where π_0, f_0 and f_1 are unknown.

4.3.2 Motivation for estimating π_0

The mixing parameter π_0 represents the proportion of non-differentially expressed genes in the microarray setting and it has attracted a lot of interest recently. The parameter π_0 is important for several reasons. Firstly, knowing the proportion of non-differentially expressed genes in a microarray experiment is of interest in its own right. It gives an important global measure of the extent of the changes studied. Secondly, knowing π_0 can help us not only control FDR, pFDR and

lFDR, but also estimate them better. Finally, this quantity is also crucial for sample-size calculations in a microarray experiment (Jung, 2005).

4.3.3 The two-component mixture model for the distribution of p-values

Many statistical methods have been proposed to estimate π_0, and most of the theoretical formulations are presented in terms of p-values rather than in terms of test statistics. Two basic assumptions are made concerning their distribution. First, it is assumed that test statistics corresponding to true null hypotheses will generate p-values that follow a uniform distribution on the unit interval. Thus, under the null distribution, the probability that a p-value falls below some threshold π_0 equals π_0. Second, p-values are, unless stated otherwise, assumed to be independent. Therefore the p-values $P_1, ..., P_m$ (not ordered) can be regarded as independent and identically distributed random variables with mixture density

$$f(p) = \pi_0 \mathrm{Un}_{[0,1]} + (1 - \pi_0)h(p), \tag{4.10}$$

where $h(p)$ is defined to be the density for P_i under the alternative distribution. One problem arising from the use of p-values is that we can't distinguish up- and down-regulated genes any more. However, one may look separately at the two tails of the distribution of the test statistic to assess differential expression corresponding to up- and down-regulation.

4.3.4 Some recent methods for estimating π_0

In the remainder of this section we give a general review of the recent papers on estimating the proportion of true null hypothesis, π_0. Note that although many of

these publications aim to the estimation of FDR, they actually focus on π_0 since a reliable estimate of this quantity is the most important step for the estimation of FDR.

Allison et al. (2002) is the first to apply the two-component mixture model to the observed p-values rather than the test statistics from multiple hypothesis testing. Besides the two basic assumptions discussed above, another assumption is made concerning the distribution of the p-values under the alternative hypothesis: the alternative distribution on the interval $[0, 1]$ can be modeled as a mixture of a few component distributions (Parker and Rothenberg, 1988). Each component is a two-parameter beta distribution with parameters α and β. Considering the fact that a uniform distribution on $[0, 1]$ can be regarded as a special form of the beta distribution when $\alpha = 1, \beta = 1$, the p-values can be modeled as independent and identically distributed random variables with mixture probability density:

$$f(p) = \pi_0 \mathrm{Un}(p|0,1) + \sum_{j=1}^{V} \pi_j \mathrm{Be}(p|\alpha_j, \beta_j) = \sum_{j=0}^{V} \pi_j \mathrm{Be}(p|\alpha_j, \beta_j) \qquad (4.11)$$

where π_0 is the proportion of true null hypotheses, V is the total number of components for the alternative distribution, π_j represents the proportion of the false null hypotheses from the jth component distribution, and α_0 and β_0 are set to be 1. A bootstrap test is used to first determine if the set of observed p-values differs significantly from a uniform distribution. If significant departure is detected, then a mixture model with a uniform component and a single two-parameter beta component is fit to the observed p-values. A bootstrap test is used to determine whether incorporation of another two-parameter beta component into the mixture model would significantly improve the model fit. This process is repeated

until it is determined that adding another beta component will not significantly improve model fit. The final fitted model is then used to compute an estimate of the FDR. This method implicitly assumes that all p-values are independent, but they show that the FDR estimates should be reasonably accurate when the p-values are mildly correlated, as long as the model fits well.

Pounds and Morris (2003) introduces a mixture model which is very similar to that of Allison et al. (2002). The mixture model consists of a continuous uniform component and a one-parameter beta component $\text{Be}(p|\alpha, 1)$ (beta-uniform mixture, BUM):

$$f(p) = \pi_0 \text{Un}(p|0, 1) + \pi_1 \text{Be}(p|\alpha, 1). \tag{4.12}$$

Maximum likelihood estimation is used to fit this model to the observed set of p-values. Given a threshold of significance, the resulting estimated distribution is partitioned into regions corresponding to the occurrences of false positives, false negative, true positives and true negatives. The geometric partition of the fitted model is used to compute estimates of the FDR and other multiple testing error rates. The method assumes that all p-values are statistically independent. The reliability of this method heavily depends on whether the BUM model can accurately represent the actual distribution of p-values.

Liao et al. (2004) develops a special mixture model (which contains a continuous uniform component and the other component derived from a flexible piecewise proportional hazards model) tailored to multiple testing by requiring the p-value distribution for the differentially expressed genes to be stochastically smaller than the p-value distribution for the non-differentially expressed genes. A smoothing mechanism is built in. A Bayesian inference is proposed for the mixture model

and a block-at-time Metropolis-Hastings algorithm (Chip and Greenberg, 1995) is used to fit the model. The fitted model gives robust estimates of local FDR.

Schweder and Spjøtvoll (1982) suggests an estimator $\hat{\pi}_0(\lambda)$ of π_0. Let $P_1, ..., P_m$ be the observed p-values. Let $W(\lambda) = \#\{P_j > \lambda\}$ be the number of p-values that are greater than some threshold value λ. Since the p-values associated with the false null hypotheses are likely to be small, a large majority of the p-values in the interval $[\lambda, 1]$, for λ not too small, should come from the uniform distribution on $[0, 1]$ (true null hypotheses). This means that,

$$E[W(\lambda)] \approx m\pi_0(1-\lambda).$$

Therefore, we have a estimator of π_0 for a given λ,

$$\hat{\pi}_0(\lambda) = \frac{W(\lambda)}{m(1-\lambda)} = \frac{\#\{P_j > \lambda\}}{m(1-\lambda)}.$$

The choice of λ is crucial for this estimator. Storey (2002) chooses

$$\hat{\pi}_0 = \min_{\lambda' \in \mathcal{R}}\{\hat{\pi}_0(\lambda')\},$$

where the minimum is computed on a grid $\mathcal{R} = \{0, 0.05, 0.10, ..., 0.95\}$. However Langaas and Lindqvist (2005) show that this estimator underestimates π_0 and propose a new estimator of π_0 which has better performance, proved by simulation studies,

$$\hat{\pi}_0 = \hat{\pi}_0(\hat{\lambda}),$$

where $\hat{\lambda} = \text{argmin}_{\lambda \in \mathcal{R}}\{\widehat{\text{MSE}}(\lambda)\}$, $\widehat{\text{MSE}}(\lambda)$ is the bootstrap estimator of $\text{MSE}\{\hat{\pi}_0(\lambda)\}$ suggested by Storey (2002) and Storey et al. (2004).

Storey and Tibshirani (2003) proposes a procedure for estimating π_0 based on spline smoothing of the function $\hat{\pi}_0(\lambda)$ (implemented in R function QVALUE). The smoothing approach is motivated by the fact that $\hat{\pi}_0(\lambda)$ usually fluctuates wildly for λ near 1 (when $\lambda \to 1$, bias decreases while variance increases). The method is applied as follows: First, $\hat{\pi}_0$ is calculated over a fine grid of λ (i.e. like the range $\{0, 0.01, 0.02, ..., 0.95\}$). Second, a natural cubic spline y with 3 degrees of freedom is fitted to $(\lambda, \hat{\pi}_0(\lambda))$. Finally, π_0 is estimated by $\lim_{\lambda \to 1} \hat{\pi}_0(\lambda)$.

Pounds and Cheng (2004) introduce a method that uses a special non-parametric density estimator called the spacings loess histogram (SPLOSH) to smooth the observed distribution of p-values and then estimate the upper bound of π_0. The SPLOSH density estimate is used to compute estimates of the cFDR and other multiple testing error rates.

Dalmasso et al. (2005) propose a family of estimators called LBE (Location Based Estimator) for an upper bound of π_0 based on the expectation of the transformed p-values and provide results on their asymptotic distribution under the assumption that the p-values are independent. In order to select one particular estimator among the proposed family, they give guidelines depending on the experimental setup and the accuracy needed.

Langaas and Lindqvist (2005) follow the two-component mixture model of the observed p-values to handle multiple testing and assume that the distribution under the alternative hypothesis, $f_1(p)$ is decreasing on $[0, 1]$ with $f_1(1) = 0$ which implies $\hat{\pi}_0 = f(1)$. Instead of parametric estimation, they derive estimators of π_0 based on nonparametric maximum likelihood estimation of the p-value density, restricting to decreasing and convex decreasing densities under the assumption of independent test statistics.

Lai (2007) proposes a moment-based method coupled with sample splitting for estimating the proportion of true null hypotheses. It is a very easy method and requires no independence assumptions. Explicit formula for the estimator of π_0 can generally be derived.

4.4 The proposed mixture models with an unknown number of components

In this section, we follow the spirit of the two-component mixture model (Efron et al., 2001) and propose that the alternative distribution $h(p)$ on $[0,1]$ might be approximated by a mixture of uniform distributions or by a mixture of one-parameter beta distributions.

4.4.1 Model 1: The uniform mixture distributions

Since a finite mixture of uniform distributions can approximate any distribution on $[0,1]$, we suggest that a possible model for the density of p-values from microarray experiment is given by:

$$f(p) = \sum_{j=0}^{k} \pi_j \mathrm{Un}(p|a_j, b_j), \quad k \geq 0, \quad 0 \leq a_j \leq b_j \leq 1, \quad a_0 = 0, \quad b_0 = 1, \quad (4.13)$$

where k is the unknown total number of components for the alternative hypothesis part of the mixture model, and the mixture weight of the jth component, π_j, satisfies that $\pi_j > 0$, $j = 0, ..., k$ and $\sum_{j=0}^{k} \pi_j = 1$. The parameters for the jth component are a_j and b_j. The first uniform distribution component is for the null hypothesis and it is completely specified so that $a_0 = 0$ and $b_0 = 1$. The remaining k components for the alternative hypothesis are not specified.

Assuming the density of the alternative distribution is a monotonic decreasing function of p-value, a more parsimonious model can be considered by letting $a_j = 0$ for all j in Equation (4.13):

$$f(p) = \sum_{j=0}^{k} \pi_j \text{Un}(p|0, b_j), \qquad k \geq 0 \text{ and } 0 \leq b_j \leq 1, \tag{4.14}$$

which only has half as many unspecified parameters as Equation (4.13) does. In order to distinguish it from the model of uniform mixtures, we call it the model of one-parameter uniform mixtures.

4.4.2 Model 2: The one-parameter beta mixture distributions

Inspired by Allison et al. (2002)'s mixture of two-parameter beta distributions for modeling the alternative distribution on [0,1], we propose a mixture of one-parameter beta distributions for the density of the alternative distribution.

The beta distribution is a family of continuous probability distribution defined on the interval $[0, 1]$ parameterized by two non-negative shape parameters. Assume a random variable X follows the beta distribution, then the probability density function of X is

$$\text{Be}(x|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}, \qquad 0 \leq x \leq 1, \tag{4.15}$$

where Γ is the gamma function and $a > 0$, $b > 0$. The beta density function can take on different shapes depending on the values of the two parameters. When $a = 1$, the beta density function in Equation (4.15) reduces to

$$\text{Be}(x|1, b) \;\; = \;\; b(1-x)^{b-1} \quad 0 \leq x \leq 1. \tag{4.16}$$

Since the uniform distribution on $[0, 1]$ is also the beta distribution $\text{Be}(1, 1)$, one can model the distribution of p-values as a finite mixture of beta distributions:

$$f(p) = \sum_{j=0}^{k} \pi_j \text{Be}(p|1, b_j), \qquad 0 \leq p \leq 1, \tag{4.17}$$

where $b_j = 1$ for $j = 0$, $\pi_j > 0$ for $j = 0, ..., k$ and $\sum_{j=0}^{k} \pi_j = 1$. This model is more flexible than BUM (Pounds and Morris, 2003) whose alternative distribution only considers one beta distribution.

4.4.3 The inference problem

Assuming that p-values capture the essence of the biological research problem, our aim is to make inference about π_0 based on a sample $p_1, ..., p_m$ from f given in Equation (4.10). However, π_0 would not be identifiable if we do not make some assumptions on the function $h(p)$.

Since p-values corresponding to false null hypotheses should presumably be small, it is natural to assume that the density $h(p)$ is very low for p near 1. It may even be natural to assume that $h(p)$ is a decreasing function of p. This motivates the assumption that $f(p)$ is decreasing with a minimum $f(1)$ at $p = 1$. This condition makes π_0 identifiable in the Equation (4.10) with $\pi_0 = f(1)$. A weaker sufficient condition for identifiability of π_0 is the existence of p_0, $0 \leq p_0 \leq 1$, with a minimum $f(p_0)$ at p_0. In practice, we may consider $f(p_0)$ as the upper bound of π_0.

4.5 A Bayesian approach for finite mixture model

4.5.1 Introduction

In the previous section we have proposed three different types of finite mixture models with an unknown number of components for the p-values arising from a microarray experiment. Due to their flexibility, mixture models can be used to model complex probability density distributions that are not easily described using standard models.

Suppose that random variables $x_1, ..., x_n$ are independent and identically distributed and have a parametric finite mixture density of form

$$f(x|k, \pi, \theta) = \sum_{j=0}^{k} \pi_j f(x|\theta_j). \tag{4.18}$$

Three types of parameters appear in the mixture model : $k+1$ is the number of components (Note that in the following we just call k the number of components for the purpose of simplicity), $\pi = (\pi_0, \pi_1, ..., \pi_k)$ denotes the mixture weights which satisfy that $\pi_j > 0$ for $j = 0, ..., k$ and $\sum_{j=0}^{k} \pi_j = 1$ and $\theta = (\theta_0, ..., \theta_k)$ denotes the parameters occurring in the mixture components. The mixture component densities $f(x|\theta_j)$ are assumed to be known and belong to the same parametric family, thus having the same functional form. We can understand the model in another way: each observed datum x has probability π_j of originating from the jth component, thus a latent vector $g = (g_1, ..., g_n)$ (also called the allocation vector) is behind the mixture model and g_i represents the index of the component that generates x_i.

Our interest is the estimates of the parameters k, π and θ from the n observations in the finite mixture model. A variety of estimation methods have been

available for this purpose, including the method of moments, maximum likelihood estimation, minimum distance and Bayesian approaches. See Titterington et al. (1985) and McLachlan and Peel (2000) for general reviews. Although maximum likelihood via EM algorithm (Dempster et al., 1977) or other numerical algorithms, such as Newton-Raphson and the method of scoring, has been the most widely applied method so far, Bayesian approaches have been getting popular due to the rapid progress of computing in the last two decades.

The basic idea of Bayesian theory is that prior beliefs about an unknown parameter vector $\psi \in \mathbf{\Psi}$ are transformed to posterior beliefs, given sample data $\mathbf{x} = (x_1, ..., x_n)$, by means of Bayes' theorem:

$$f(\psi|\mathbf{x}) = \frac{f(\psi, \mathbf{x})}{f(\mathbf{x})} = \frac{f(\psi)f(\mathbf{x}|\psi)}{\int_{\mathbf{\Psi}} f(\psi)f(\mathbf{x}|\psi)d\psi}, \tag{4.19}$$

where $f(\psi)$ is called the prior distribution, $f(\mathbf{x}|\psi)$ is the likelihood function and $f(\psi|\mathbf{x})$ is the posterior distribution. The role of the denominator is to make the posterior distribution integrate to 1. Note that in the mixture model (4.18) ψ corresponds to the parameters k, π and θ.

When ψ consists of more than three unknown parameters the numerical evaluation of the denominator of Equation (4.19) will be a demanding task. This is an important reason why the implementation of the Bayesian paradigm for mixture models is not at all straightforward.

The application of the Bayesian approach to the estimation of finite mixtures was accelerated only after the break-through papers by Tanner and Wong (1987) and Gelfand and Smith (1990). These publications creatively introduce two Markov chain Monte Carlo (McMC) algorithms called Data Augmentation and Gibbs sampling that allow simulation from complex posterior distributions

in a simple practical manner. A few years later, Diebolt and Robert (1994) apply both of these two MCMC methods in a mixture context, to estimate the posterior distributions for a mixture of normals with number of components k assumed known. But how to determine the number k of components in the mixture? It is a even more challenging problem. Although it has been researched for years there isn't a fully satisfactory solution available. Many different informal and formal approaches have been proposed. One method of choosing the number of components is to construct the posterior of k. A prior distribution is placed on k and then the marginal likelihoods for each k are estimated. The posterior of k is then found by simple implementation of Bayes' theorem. It is what Nobile (1994, 2005) and Roeder and Wasserman (1997) do. Green (1995) proposes a novel MCMC method known as Reversible Jump MCMC (RJMCMC) which allows the MCMC sampler to jump between different models. In the case of finite mixture models the sampler can jump between models with different number of components. Richardson and Green (1997) apply RJMCMC to sample from the joint posterior distribution of all the parameters, including the number k of components. They evaluate the posterior distribution of k by computing the relative frequency for each model visited throughout the simulation. However, if more and more parameters are included in the model, the dimension-jumping moves of RJMCMC will turn out to be very computational. A way of counteracting this significant increase in the number of parameters with an increase of k is to integrate some of the parameters out of the models, for example the component parameters and weights, only leaving in the model the number of components and the allocation vector. The integration is computable in a closed form if conjugate priors are used for the parameters. Then the only unknowns left in the model are the number of components k and the vector of allocations conveying from

which components each of the observations is coming. Quite a few works have been done in this framework such as Nobile (1994), Steele and Emond (2003), Fearnhead (2004). The latest work is Nobile and Fearnside (2007) and Fearnside (2007). They propose a new MCMC sampler, which has both the component parameters and weights integrated out. In the remainder of this section we make a summary of this MCMC sampler, and then apply it to our finite mixture models in Section 4.4.

4.5.2 The allocation sampler

Nobile and Fearnside (2007) propose an MCMC method called "the allocation sampler" for the Bayesian analysis of finite mixture distributions with an unknown number of components. The object of the allocation sampler is to draw samples from the joint posterior distribution of the number k of components and the allocation variables g under the assumption that the component parameters θ and mixture weights π can be integrated out of the model analytically:

$$f(k, g|x, \phi) \propto f(k, g, x|\phi) = f(k)f(g|k)f(x|k, g, \phi) \tag{4.20}$$

where ϕ is a vector of hyperparameters in the prior distribution over θ (parameters in θ). In order to get the explicit expression of $f(k, g|x, \phi)$, we should know the expressions of $f(k)$, $f(g|k)$ and $f(x|k, g, \phi)$. The first item is the prior distribution on k, and following Nobile (2005) we choose the $Poi(1)$ distribution as prior on k, restricted to $1 < k \leq k_{max}$; k_{max} is set to 50 in this thesis. The expressions of the remaining two items are discussed in Sections 4.5.2.1 and 4.5.2.2 which are based on Nobile and Fearnside (2007).

4.5.2.1 Calculating $f(g|k)$

We begin with the allocation vector $g = (g_1, ..., g_n)^t$, where g_i is the index of the component that generated x_i. It is assumed that the g_i are conditionally independent given k and π and that

$$\Pr(g_i = j|k, \pi) = \pi_j, \quad j = 0, 1, ..., k, \quad i = 1, .., n.$$

Therefore we have

$$f(g|k, \pi) = \prod_{j=0}^{k} \pi_j^{n_j}, \quad \sum_{j=0}^{k} n_j = n \tag{4.21}$$

where n_j is the number of observations generated from the jth component: $n_j = \text{card}\{A_j\}$ and A_j is the set of indices of the observations that g allocates to component j: $A_j = \{i : g_i = j\}$.

A popular choice for the prior on the mixture weights $\pi = (\pi_0, \ldots, \pi_k)$ is the Dirichlet distribution, $Dir(\alpha_0, ..., \alpha_k)$, where $\alpha_i > 0$ for $j = 0, 1, \ldots, k$:

$$f(\pi|k) = \frac{\Gamma(\alpha^*)}{\Gamma(\alpha_0)...\Gamma(\alpha_k)} \pi_0^{\alpha_0 - 1} ... \pi_k^{\alpha_k - 1}, \quad \pi_j \geq 0, \ \sum_{j=0}^{k} \pi_j = 1 \tag{4.22}$$

where $\alpha^* = \sum_{j=0}^{k} \alpha_j$. We have chosen to use a symmetric Dirichlet distribution in this setting, where the hyperparameters are $\alpha_j = \alpha_0 = 1$. Consequently, the prior can be thought of as a uniform distribution on the simplex of the weights. This distribution is a conjugate prior for the mixture weight and it is also used by Richardson and Green (1997) and Stephens (2000).

One can obtain $f(g|k)$ by integrating the density (4.21) with respect to the

density of the mixture weights:

$$\begin{aligned} f(g|k) &= \int f(g|k,\pi) f(\pi|k) d\pi \\ &= \frac{\Gamma(\alpha^*)}{\Gamma(\alpha^*+n)} \prod_{j=0}^{k} \frac{\Gamma(\alpha_j+n_j)}{\Gamma(\alpha_j)}. \end{aligned} \tag{4.23}$$

4.5.2.2 Calculating $f(x|k,g,\phi)$

In order to get the expression for $f(x|k,g,\phi)$, we need to know $f(x|k,\pi,\theta,g)$ and $f(\theta|k,\pi,g,\phi)$. Since the density of x_i is $f(x_i|\theta_{g_i})$ and the data $x_1,...,x_n$ are assumed conditionally independent given k, π, θ and g, we have

$$f(x|k,\pi,\theta,g) = \prod_{i=1}^{n} f(x_i|\theta_{g_i}). \tag{4.24}$$

Similarly, the component parameters θ_j are assumed independent of π and g, conditional on k. They are conditional independent with prior distributions $f(\theta_j|\phi_j)$, given hyperparameters $\phi = \{\phi_0,...,\phi_k\}$. Thus,

$$f(\theta|k,\pi,g,\phi) = \prod_{j=0}^{k} f(\theta_j|\phi_j). \tag{4.25}$$

We assume that the independent priors on the θ_j's, $f(\theta_j|\phi_j)$ are chosen so that the parameters θ_j's can be integrated out analytically from (4.24). After multiplying (4.24) and (4.25) and integrating out θ, we get

$$\begin{aligned} f(x|k,g,\phi) &= \int f(x|k,\pi,g,\theta,\phi) f(\theta|k,\pi,g,\phi) d\theta \\ &= \int \prod_{i=1}^{n} f(x_i|\theta_{g_i}) \prod_{j=0}^{k} f(\theta_j|\phi_j) d\theta_j \\ &= \prod_{j=0}^{k} \int \prod_{i \in A_j} f(x_i|\theta_j) f(\theta_j|\phi_j) d\theta_j \\ &= \prod_{j=0}^{k} f(x^j|\phi_j) \end{aligned} \tag{4.26}$$

where $x^j = \{x_i : i \in A_j\}$. We use the shorthand $f(x^j|\phi_j)$ to denote the integral in the third line of Equation (4.26):

$$f(x^j|\phi_j) = \int \prod_{i \in A_j} f(x_i|\theta_j) f(\theta_j|\phi_j) d\theta_j \tag{4.27}$$

Note that if $A_j = \emptyset$, $f(x^j|\phi_j) = 1$.

So far, we have presented the general expression of $f(x|k,g,\phi)$. In the next section we show how this specializes to the cases of mixtures of uniforms and mixtures of one-parameter betas proposed in Section 4.4.

4.5.2.3 Application to Model 1: The uniform mixture distributions

For the mixture of uniform distributions in Equation (4.13), the densities $f(x_i|\theta_{g_i})$ in Equation (4.24) are $\text{Un}(x_i|a_j,b_j)$, which is $1/(b_j - a_j)$ for $a_j < x_i < b_j$ and 0 for $x_i \geq b_j$ or $x_i \leq a_j$. For simplicity, we denote

$$\mathrm{Un}(x_i|a_j, b_j) = \frac{1}{b_j - a_j}\mathrm{I}_{(a_j,b_j)}(x_i), \tag{4.28}$$

where $\mathrm{I}_A(x)$ is an indicator function which takes on the value 1 if $x \in A$ and the value 0 otherwise. Thus,

$$\begin{aligned}
\prod_{i\in A_j} f(x_i|\theta_j) &= \prod_{i=1}^{n_j} \mathrm{Un}(x_i|a_j, b_j) \\
&= \prod_{i=1}^{n_j} \frac{1}{b_j - a_j}\mathrm{I}_{(a_j,b_j)}(x_i) \\
&= \frac{1}{(b_j - a_j)^{n_j}} \prod_{i=1}^{n_j} \mathrm{I}_{(a_j,\infty)}(x_i) \cdot \prod_{i=1}^{n_j} \mathrm{I}_{(-\infty,b_j)}(x_i),
\end{aligned}$$

and since $\prod_{i=1}^{n_j} \mathrm{I}_{(a_j,\infty)}(x_i)$ and $\prod_{i=1}^{n_j} \mathrm{I}_{(-\infty,b_j)}(x_i)$ are equivalent to $\mathrm{I}_{(-\infty,x_{(1)})}(a_j)$ and $\mathrm{I}_{(x_{(n_j)},\infty)}(b_j)$ respectively, where $x_{(i)}$ is the ith order statistic of x^j for $i = 1, \ldots, n_j$, we have

$$\prod_{i\in A_j} f(x_i|\theta_j) = \frac{1}{(b_j - a_j)^{n_j}}\mathrm{I}_{(-\infty,x_{(1)})}(a_j) \cdot \mathrm{I}_{(x_{(n_j)},\infty)}(b_j).$$

Independent priors are assigned to the parameters (a_j, b_j), $j = 1, ..., k$ and we let $f(a_j, b_j) = 2/(\phi_2 - \phi_1)^2$, $\phi_1 < a_j < b_j < \phi_2$. Therefore, using Equation (3.27)

$$\begin{aligned} f(x^j|\phi_j) &= \frac{2}{(\phi_2-\phi_1)^2}\int \frac{\mathrm{I}_{(-\infty,x_{(1)})}(a_j)\cdot \mathrm{I}_{(x_{(n_j)},\infty)}(b_j)}{(b_j-a_j)^{n_j}} da_j db_j \\ &= \frac{2}{(\phi_2-\phi_1)^2}\int_{x_{(n_j)}}^{\phi_2}\int_{\phi_1}^{x_{(1)}} \frac{1}{(b_j-a_j)^{n_j}} da_j db_j, \end{aligned}$$

which is a standard double integral problem. Fearnside (2007) considers this model with $\phi_2 = -\phi_1$ and computes $f(x^j|\phi_j)$. Here we allow for general $\phi_2 > \phi_1$ and following the same derivations as in Fearnside (2007, Appendix A.1) we get the marginal density of the data allocated to the jth component,

$$f(x^j|\phi_j) = \begin{cases} \frac{2}{(\phi_2-\phi_1)^2}\frac{\left[(x_{(n_j)}-x_{(1)})^{2-n_j}-(x_{(n_j)}-\phi_1)^{2-n_j}-(\phi_2-x_{(1)})^{2-n_j}+(\phi_2-\phi_1)^{2-n_j}\right]}{(n_j-1)(n_j-2)} & \\ & n_j > 2, \\ \frac{2}{(\phi_2-\phi_1)^2}\left[\log\frac{(\phi_2-x_{(1)})(x_{(n_j)}-\phi_1)}{(\phi_2-\phi_1)(x_{(n_j)}-x_{(1)})}\right] & \\ & n_j = 2, \\ \frac{2}{(\phi_2-\phi_1)^2}\left[x_{(1)}\log\frac{\phi_2-x_{(1)}}{x_{(1)}-\phi_1}+\phi_2\log\frac{\phi_2-\phi_1}{\phi_2-x_{(1)}}+\phi_1\log\frac{x_{(1)}-\phi_1}{\phi_2-\phi_1}\right] & \\ & n_j = 1. \end{cases} \tag{4.29}$$

If we choose $\phi_1 = 0$ and $\phi_2 = 1$, a seemingly sensible choice in Equation (4.13), we obtain a much simplified form:

$$f(x^j|\phi_j) = \begin{cases} \frac{2\left[(x_{(n_j)}-x_{(1)})^{2-n_j} - x_{(n_j)}^{2-n_j} - (1-x_{(1)})^{2-n_j}+1\right]}{(n_j-1)(n_j-2)} & n_j > 2, \\ 2\log\frac{x_{(n_j)}(1-x_{(1)})}{x_{(n_j)}-x_{(1)}} & n_j = 2, \\ 2\left(x_{(1)}\log\frac{1-x_{(1)}}{x_{(1)}} + \log\frac{1}{1-x_{(1)}}\right) & n_j = 1. \end{cases} \quad (4.30)$$

In a similar way, we derive the expression of $f(x^j|\phi_j)$ for the model of a mixture of one-parameter uniforms in (4.14) where a_j is fixed to be 0:

$$f(x^j|\phi_j) = \begin{cases} \frac{1-x_{(n_j)}^{1-n_j}}{1-n_j} & n_j > 1, \\ \log\frac{1}{x_{(n_j)}} & n_j = 1. \end{cases} \quad (4.31)$$

For the details of the derivation of (4.31), see Appendix B.1.

Note that the results (4.30) and (4.31) are only valid for the components $j = 1, \ldots, k$. For the first component ($j = 0$) in Model 1, it has been already specified as a standard uniform distribution ($a_0 = 0, b_0 = 1$), therefore $f(x^j|\phi_j) = 1$.

4.5.2.4 Application to Model 2: The one-parameter beta mixture distributions

For the mixture of one-parameter beta distributions in Equation (4.17), the density $f(x_i|\theta_j)$ in Equation (4.24) is $\text{Be}(x_i|1, b_j)$, that is $b_j(1-x_i)^{b_j-1}$, see Equation

(4.16). Independent exponential priors are assigned to the parameters b_j for $j = 1, ..., k$ and that is $f(b_j) = \gamma e^{-\gamma b_j}$. Therefore,

$$\begin{aligned} f(x^j|\phi_j) &= \int \prod_{i \in A_j} f(x_i|\theta_j) f(\theta_j|\phi_j) d\theta \\ &= \int_0^\infty \prod_{i \in A_j} b_j (1 - x_i)^{b_j - 1} \gamma e^{-\gamma b_j} db_j \\ &= \int_0^\infty b_j^{n_j} \left[\prod_{i=1}^{n_j} (1 - x_i) \right]^{b_j - 1} \gamma e^{-\gamma b_j} db_j. \end{aligned}$$

After straightforward computations (for the details, see Appendix B.2), the marginal distribution of the data allocated to the jth component is:

$$f(x^j|\phi_j) = \frac{\gamma \, \Gamma(n_j + 1)}{\prod_{i=1}^{n_j} (1 - x_i) \Big[\gamma - \sum_{i=1}^{n_j} \log(1 - x_i) \Big]^{n_j + 1}}. \tag{4.32}$$

Note that the result (4.32) is only valid for the components $j = 1, \ldots, k$. For the first component ($j = 0$) in Model 2, it has been already specified as a standard uniform distribution ($b_0 = 1$), therefore $f(x^j|\phi_j) = 1$.

4.5.2.5 Posterior distributions

Finite mixture models can be summarized by looking at the posterior distributions of all the parameters in the model. Even though the parameters do not explicitly appear in the sampling procedure, the posterior distributions can be calculated using the MCMC output. This is in contrast to the usual RJMCMC

scheme, where all the parameters explicitly appear in the MCMC sampling procedure.

In this thesis, our interest is only in the posterior distributions of the first component's weight π_0, the number of components k and the posterior predictive distribution of a future observation x_{n+1}.

For the mixture weight π_i, its prior distribution is the beta distribution, $\pi_i \sim \text{Be}(\alpha_i, \alpha_* - \alpha_j)$, where $\alpha_* = \sum_{j=0}^{k} \alpha_j$. The posterior distribution of the mixture weight π_i conditional on k and g is also a beta distribution:

$$\pi_i|k, g, x \sim \text{Be}(\alpha_i + n_i, \alpha_* - \alpha_i + n - n_i). \tag{4.33}$$

Therefore, the marginal posterior distribution of the weights unconditional on g are found by averaging the right hand side of Equation (4.33) over the posterior distribution of g:

$$\pi_i|k, x \sim \sum_g f(g|k, x)\text{Be}(\alpha_i + n_i, \alpha_0 - \alpha_i + n - n_i). \tag{4.34}$$

It only makes sense to calculate the posterior distribution of the weight given a certain value of k, because the meaning of the weight for component j changes as k changes.

The posterior distribution of the number of components is a product of the MCMC sampler. We keep a record of the changing states of k throughout the simulation to estimate this posterior. The posterior probability for having k components in the model is found by taking the ratio of the sampler being in a state with k components and the total number of visited states:

$$\widehat{f(k|x)} = \frac{\sum_{i=1}^{N} I(k^{(i)} = k)}{N}, \tag{4.35}$$

where $k^{(i)}$ is the number of components in the ith simulation, N is the total number of allocation vectors simulated by the MCMC sampler and I is the indicator function.

The posterior predictive distribution is of great importance when using mixtures as a density estimation tool. Conditional on k, π, θ, g and x the future observation x_{n+1} is independent of the previous data x and has distribution of the same form as Equation (4.18):

$$f(x_{n+1}|k, \pi, \theta, g, x, \phi) = \sum_{j=0}^{k} \pi_j f(x_{n+1}|\theta_j). \tag{4.36}$$

Integrating this density with respect to the joint distribution of π and θ given k, g and x and then averaging it with respect to the joint distribution of k and g (see Chapter 2, page 36 of Fearnside (2007)) yields the posterior prediction of x_{n+1}:

$$f(x_{n+1}|x, \phi) = \sum_{k,g} f(k, g|x, \phi) \sum_{j=0}^{k} \frac{\alpha_j + n_j}{\alpha_* + n} f(x_{n+1}|x^j, \phi_j), \tag{4.37}$$

where

$$f(x_{n+1}|x^j, \phi_j) = \int f(x_{n+1}|\theta_j) f(\theta_j|x^j, \phi_j) d\theta_j \tag{4.38}$$

is the posterior predictive density of x_{n+1} according to component j. Note that this expression can be simplified, see Chapter 2, page 37 of Fearnside (2007) for

more details.

4.5.2.6 Implementation of the allocation sampler

For the implementation of the allocation sampler, we basically follow the way proposed by Nobile and Fearnside (2007) and Fearnside (2007) to sample from the joint posterior distribution of the number of components k and the allocation vector g given in Equation (4.20).

The allocation sampler is a hybrid approach and it makes use of both fixed k moves and variable k moves in order to try and move around the whole state space (i.e. all the possible allocation vectors) when approximating the posterior distribution. The sampler starts a move by firstly randomly selecting between these two types of move with equal probability.

The first type of moves updates g while keeping k at its current value and consists of:

- Gibbs sampling on the components of g,
- three different Metropolis-Hastings moves on g.

The Gibbs sampler on the components of g, from g_1 to g_n, only changes one component of g at each step and it guarantees that theoretically the whole state space would be swept given enough simulation time. In contrast, the three different Metropolis-Hastings moves on g can change several components of g at the same time. The combined use of them aims at moving around the state space more efficiently, especially for the case of large sample size n.

The second type of moves changes the number of components k and the allocation vector g simultaneously and consists of a pair of Metropolis-Hastings moves: a move creates a new component (ejection move) and a reverse move

deletes a existing component under the constraint of $1 \leq k \leq k_{max}$ (absorption move). For more technical details of the two types of moves and the performance evaluation of the sampler, see Section 3 of Fearnside (2007).

Besides the above two type of moves, an extra "post-processing" step (a relabeling method) is used to reassign labels in the allocation to perform parameter inference (Fearnside, 2007; Nobile and Fearnside, 2007). This issue arises from the lack of identifiability from finite mixture distributions. Finite mixture distributions are not identifiable because the likelihood function for a mixture model is invariant to a permutation of the labels of the components in the model. For example, in a mixture of two components, whether the components are labeled $\{1, 2\}$ or $\{2, 1\}$ has no influence on the value of likelihood of the mixture model. This lack of identifiability should be addressed if parameter estimation is of interest. In this chapter we are interested in estimating π_0 only, and since the first component is completely specified to be a standard uniform distribution, the inference of π_0 may proceed without the need of a "post-processing" step to reassign the labels.

4.6 Applications and results

In this section a selection of different datasets will be analysed using the allocation sampler. We shall not focus on the type of data analysis, preprocessing and test for differential expression that have been performed to produce the p-values, but instead on reporting and comparing to previous analyses carried out by other methods.

4.6.1 Allocation sampler procedure

The allocation sampler is implemented to produce the posterior results in the same way for all the following datasets. Initially, the sampler is started from $k = 1$ and had a burn-in of 10000 iterations preceding another 1000000 iterations. A thinning parameter, 100, is used to produce a sample of 10000 draws from the iterations. For the thinning parameter of the next run of the allocation sampler, we simply double the value of the latest thinning parameter. We keep running the sampler by updating the thinning parameter until we find that the Markov chain has converged. In contrast, Nobile and Fearnside (2007) and Fearnside (2007) chose a more complex way to determine the thinning parameter, see Chapter 4, page 89 of Fearnside (2007).

There are several ways to judge the convergence of the Markov chain. One way is to study the cumulative occupancy fraction of k. If the pattern is stable, then it seems that the Markov chain has converged. Another way is to calculate the AR estimate of effective sample size, see Fearnside (2007, Appendix B).

Since the samples drawn from the posterior distribution are not independent, the effective sample size is the number of independent samples required to produce an estimate with the same precision as that given by a number of dependent

samples. Normally several thousand effective samples is a good indication of the convergence of Markov chain.

The implementation of MCMC is always very computational intensive. In order to be more efficient, the code of the allocation sampler is written in combination of Fortran and R (Fearnside, 2007; Nobile and Fearnside, 2007). A workstation with a AMD Opteron CPU and 4 GB RAM is used to execute the allocation sampler for the datasets in this section. The amount of processor time required for running the allocation sampler depends on a lot of factors such as the size of the dataset, the number of the iterations, the number of the components of the mixture and the type of the model used. Therefore we do not think it is very meaningful to record and compare the exact processor running time for each dataset in this section, since the datasets are different from each other. However, to give readers a rough idea of the implementation speed, as an example, we report that the allocation sampler would cost about 2.5 hours to finish 1 million iterations for 10000 data observations using beta mixture model with 3 or 4 components.

4.6.2 Breast cancer data

For our first example, we consider the data from the study of Hedenfalk et al. (2001), which examined gene expressions in breast cancer tissues from women who were carriers of the hereditary BRCA1 or BRCA2 gene mutations, predisposing to breast cancer. The dataset comprised the measurement of 3226 genes using cDNA arrays, for 7 BRCA1 tumours and 8 BRCA2 tumours. It is publicly available at `http://research.nhgri.nih.gov/microarray/NEJM_Supplement`. A total of 56 genes were filtered out, because they had one or more expression measurements exceeding 20, which were considered not trustworthy (Storey and

Tibshirani, 2003). Therefore, 3170 gene expression measurements for 15 samples are used here. The p-values are calculated on the basis of permutation tests, as described in Storey and Tibshirani (2003).

We first apply the allocation sampler to Hedenfalk's data using the beta mixtures. The corresponding result is shown in Figure 4.1, which contains several subfigures. The top subfigure shows the trace of 10000 samples of k from 2 million iterations. The middle left subfigure shows that the AR estimate of effective sample size is 6780, which is large enough for a reasonable precision of the estimates. The middle right subfigure shows the cumulative occupancy fraction of k is very stable, which also means the convergence of Markov chain. Since the most frequent number of components k is 3 (the probability is more than 0.75), the histogram of estimates of π_0 conditional on $k = 3$ is shown in the bottom right subfigure. The bottom middle subfigure is the plot of the posterior predictive distribution imposed on the histogram of the original Hedenfalk's data (i.e. f density estimate).

We also apply the allocation sampler to Hedenfalk's data using uniform mixtures and one-parameter uniform mixtures respectively. The corresponding results are displayed in Figure 4.2 and Figure 4.3.

It shows that for the efficiency of the implementation of the allocation sampler the beta mixtures is better than the one-parameter uniform mixtures and the one-parameter uniform mixtures is better than the uniform mixtures. The effective sample size for the beta mixtures achieves 6780 from just 2 million iterations, in contrast, the effective sample size for the the one-parameter uniform mixtures and the uniform mixtures is only 305 from 64 million iterations and 1106 from 8 million iterations. It means that for the one-parameter uniform mixtures and the uniform mixtures the MCMC chain is not mixing well. One possible reason

is that they involve more components than the beta mixtures.

In this chapter, our key focus is the estimation of π_0. There are actually two different ways to estimate the upper bound of π_0. One is to use the estimate of π_0 from our method directly. The other is to use the minimum value of f as the estimate of π_0 (see Section 4.4.3). As a simple way to do it, we can compute f values on a fine grid of the $[0, 1]$ interval and select the smallest one from them to approximate the minimum. We expect both of the ways to work well and give very similar estimates. In reality, it is not true with the first one: sometimes it would have difficulty in identifying π_0. This problem originates from our method which does not impose any condition for π_0 identification to the mixture model. In our method, we only fix the first component to be $\text{Un}(0, 1)$ and then just let the method automatically specify the remaining components and decide which data observation (p-value) is from which component. So, sometimes we might have the following situation (note that the problem isn't found in the case of Hedenfalk's data): for some datasets, our method can find another component very similar to the first standard uniform component, and then it would be very difficult to assign the observations to which of the two components in a proper way so that the estimate of π_0 would be quite variable. Therefore, in order to avoid the effect of any potential "lack of identification" problem, we prefer the way of estimating π_0 from minimum f.

Table 4.3 summarizes the posterior distribution of π_0 for each of the three models. Although the effective sample size is not ideally large enough for the case of the one-parameter uniform mixtures and the uniform mixtures, all the results are quite consistent: the median of posterior π_0 only varies in a small interval from 0.656 to 0.694. In contrast, among other analyses of this dataset, π_0 was estimated to be 0.669 by QVALUE, 0.586 by BUM, 0.622 by SPLOSH,

Table 4.3: Hedenfalk's breast cancer data: the estimation of π_0 using three different mixture models.

Models	Percentiles of π_0 posterior				
	2.5th	25th	50th	75th	97.5th
Beta mixtures	0.613	0.643	0.656	0.669	0.693
One-parameter uniform mixtures	0.610	0.666	0.679	0.692	0.722
Uniform mixtures	0.648	0.677	0.694	0.713	0.731

0.688 by LBE, 0.675 by Langaas and Lindqvist (2005) and 0.673 by Liao et al. (2004). There is a very high degree of agreement between our method and these published methods except BUM.

The research for multiple hypotheses testing has so far mainly focused on FDR and pFDR. The methods for lFDR are much less developed. One reason is that it is more difficult to estimate the lFDR than FDR or pFDR. The FDR or pFDR can be formulated in terms of F, the cumulative distribution of f, for which the empirical distribution of the p-values is a consistent and stable estimator (Storey, 2003). To estimate the lFDR, however, it is necessary to estimate the density f. Here our method provides such a estimate via calculating the posterior predictive distribution according to Equation (4.37). After knowing the estimate of π_0, F and f, we are able to calculate lFDR and pFDR according to Equation (4.8) and (4.5) respectively.

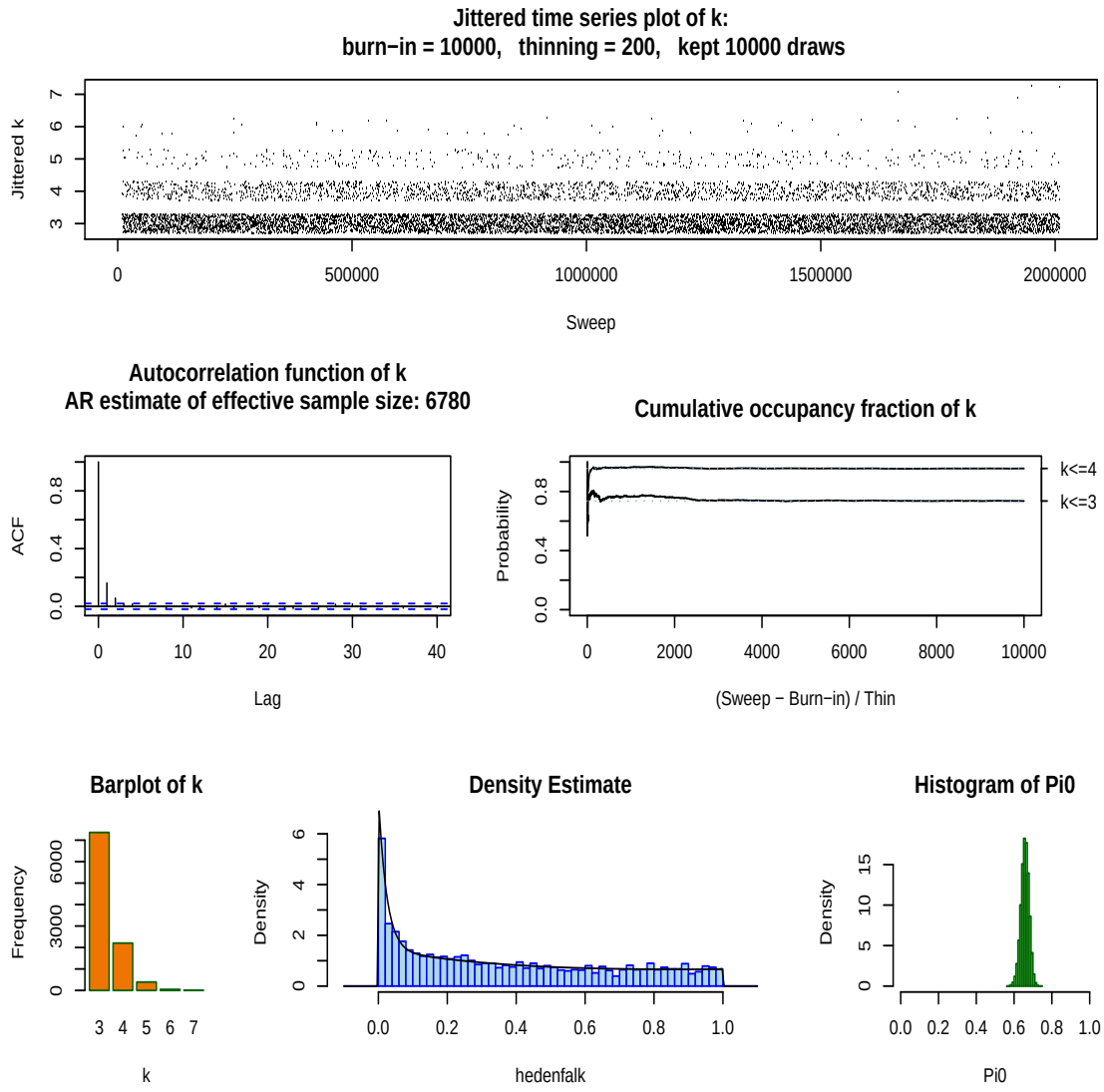


Figure 4.1: Analysis of Hedenfalk's breast cancer data using the beta mixture distributions. From top to bottom and from left to right, it shows jittered time series plot of k, autocorrelation function of k, cumulative occupancy fraction of k, the plots of posterior predictive distribution imposed on histogram of p-values, the posterior of number of components and the histogram of the posterior π_0 conditional on the number of the most frequent component (i.e. 3 in this case).

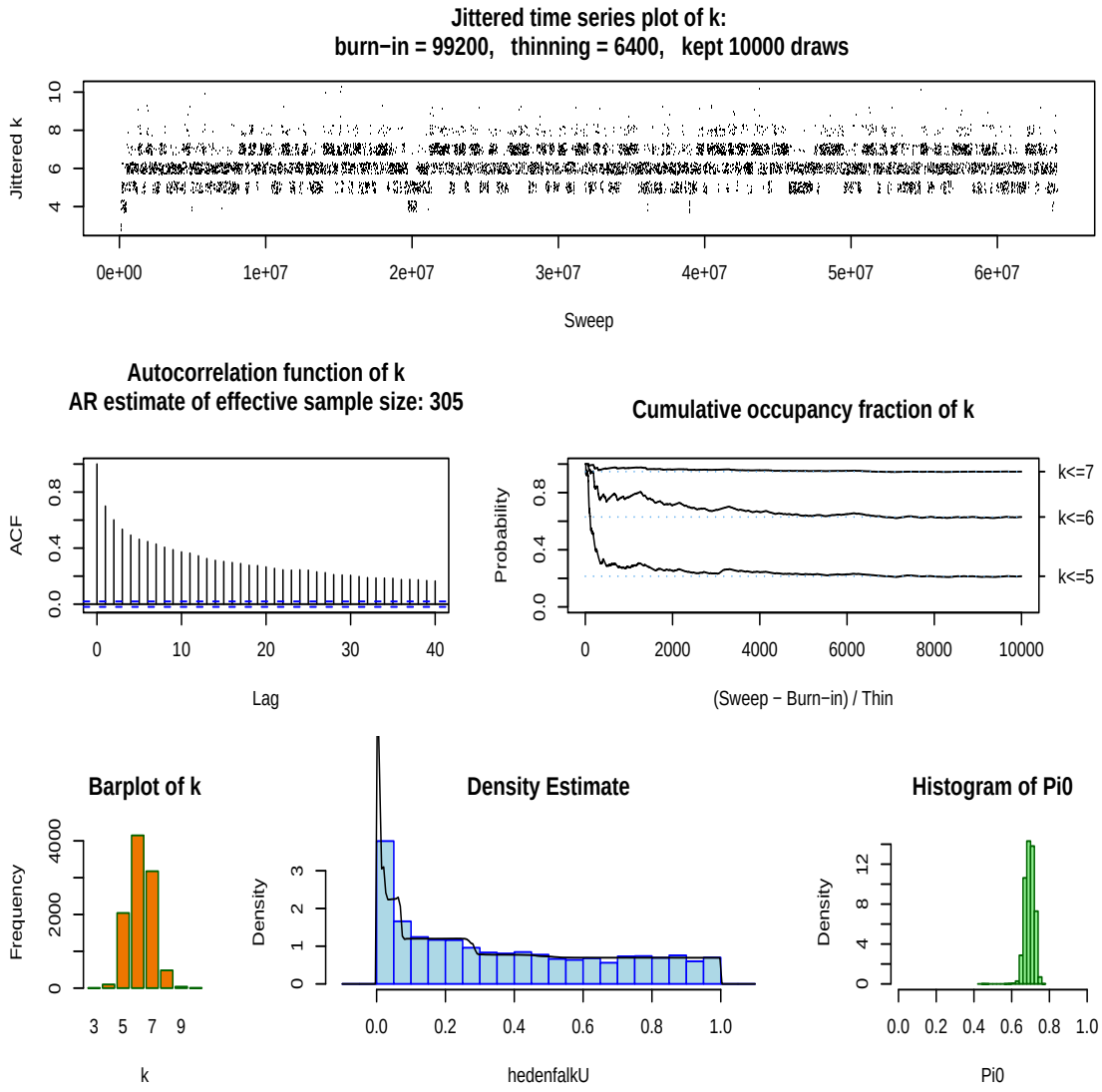


Figure 4.2: Analysis of Hedenfalk's breast cancer data using the one-parameter uniform mixture distributions. From top to bottom and from left to right, it shows jittered time series plot of k, autocorrelation function of k, cumulative occupancy fraction of k, the plots of posterior predictive distribution imposed on histogram of p-values, the posterior of number of components and the histogram of the posterior π_0 conditional on the number of the most frequent component (i.e. 6 in this case).

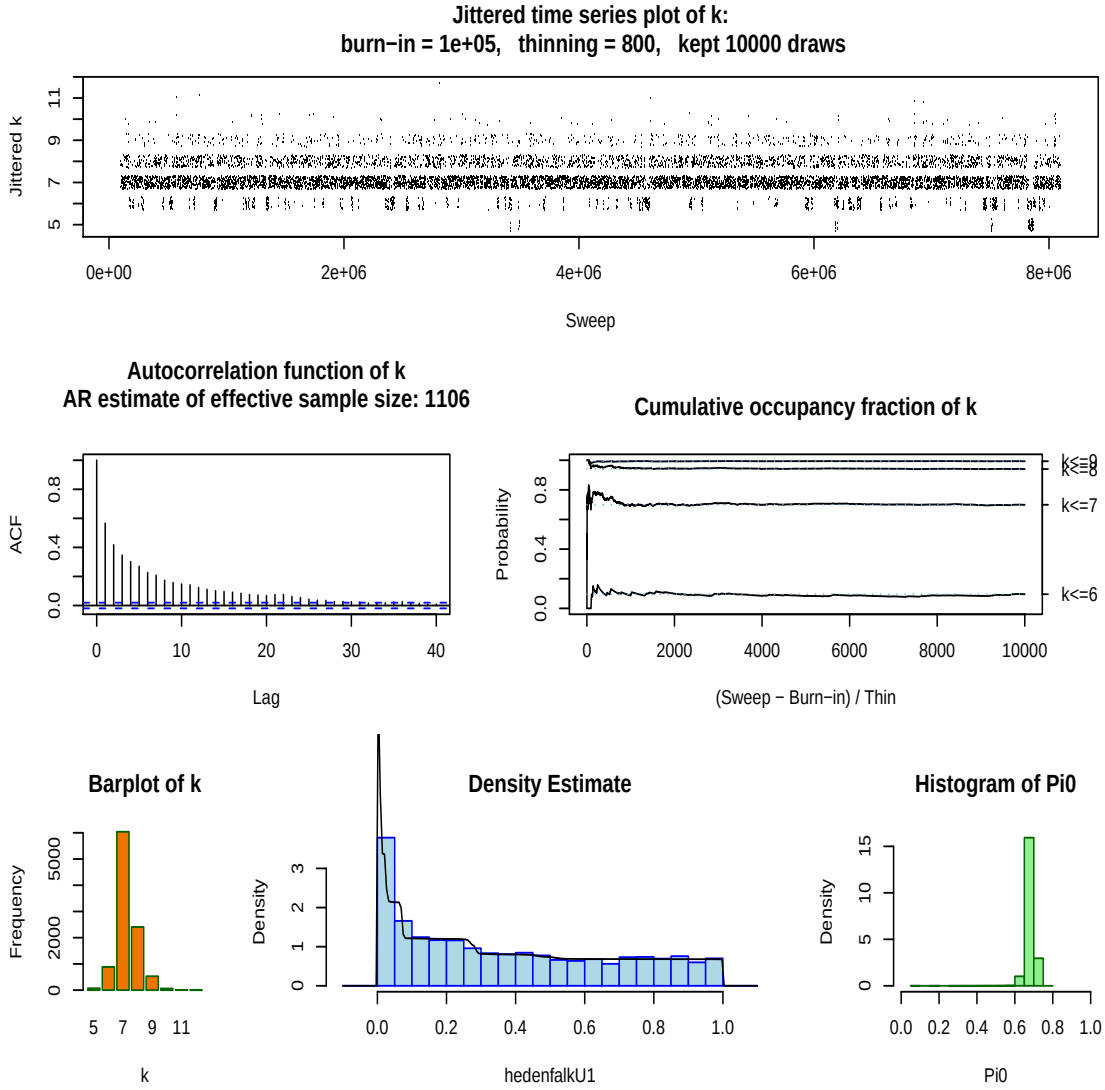


Figure 4.3: Analysis of Hedenfalk's breast cancer data using the uniform mixture distributions. From top to bottom and from left to right, it shows jittered time series plot of k, autocorrelation function of k, cumulative occupancy fraction of k, the plots of posterior predictive distribution imposed on histogram of p-values, the posterior of number of components and the histogram of the posterior π_0 conditional on the number of the most frequent component (i.e. 7 in this case).

Since the estimate of π_0 is represented in a distribution form, the estimates of lFDR and pFDR is also in a distribution form. Figure 4.6 (a) and (b) plot low and high bounds of 95% and 50% credible interval and median against raw p-value for the estimated lFDR and pFDR respectively (beta mixtures model is used here). For the purpose of comparison, we also plot the lFDR and pFDR estimates using Liao's method (The R code is available at `http://www.geocities.com/jg liao/software`) and Storey's QVALUE method (The R code is available at `http://faculty.washington.edu/~jstorey`) in these two figures respectively. It shows that the estimates by our method are quite close to those by Liao's method and Storey's QVALUE in the whole range of p-values.

For the purpose of model checking, we also estimate $\hat{F}$ and the cumulative distribution F_m according to the following two equations:

$$\hat{F}(p) = \hat{\pi}_0 p + (1 - \hat{\pi}_0)\hat{F}_1(p), \tag{4.39}$$

where F_1 is the cumulative distribution for density f_1 for p-values under the alternative distribution, and

$$F_m(p) = \#\{P_i \leq p\}/m, \tag{4.40}$$

which is the empirical cumulative distribution of raw p-values $P_1, \ldots, P_m$ and converges to $F(p)$ uniformly over $p \in [0, 1]$. The estimated $\hat{F}$ (scaled to unit) and the empirical cumulative distribution F_m is plotted in Figure 4.6(c). It shows that they are almost identical, indicating excellent model fitting.

4.6.3 Lipid metabolism data

The second data is from a study of lipid metabolism by Callow et al. (2000). The apolipoprotein AI (ApoAI) gene is known to play a pivotal role in high density lipoprotein (HDL) metabolism. Mice which have the ApoAI gene knocked out have very low HDL cholesterol levels. The purpose of this experiment is to determine how ApoAI deficiency affects the action of other genes in the liver, with the idea that this will help determine the molecular pathways through which ApoAI operates. The experiment compared 8 ApoAI knockout mice with 8 normal C57BL/6 ("black six") mice, the control mice. For each of these 16 mice, target mRNA was obtained from liver tissue and labelled using a Cy5 dye. The RNA from each mouse was hybridized to a separate microarray. Common reference RNA was labelled with Cy3 dye and used for all the arrays. The reference RNA was obtained by pooling RNA extracted from the 8 control mice. In total, the experiment involves 8 microarrays and for each microarray 6384 genes were measured. The raw experiment data is available at `http://bioinf.wehi.edu.au/limmaGUI/DataSets.html` and is analysed as described in Smyth, Thorne and Wettenhall (2005), on the basis of the theory presented in Smyth (2004). The resulting p-values from the comparison of knockout mice with normal mice were the input to the estimation of π_0.

We apply the allocation sampler to Callow's data using the model of beta mixtures, one-parameter uniform mixtures and uniform mixtures respectively. Similar to the Hedenfalk's data, for the one-parameter uniform mixtures and uniform mixtures the MCMC chain is not mixing well. Even after huge number of iterations, the effective sample size is still not large enough. So here for Callow's data we just show the output result from the beta mixtures model in Figure 4.4.

Table 4.4 summarizes the posterior distribution of π_0 conditional on the most

Table 4.4: The Callow's lipid metabolism data: the estimation of π_0 using three different mixture models.

Models	Percentiles of π_0 posterior				
	2.5th	25th	50th	75th	97.5th
Beta mixtures	0.817	0.854	0.868	0.881	0.902
One-parameter uniform mixtures	0.815	0.865	0.889	0.904	0.923
Uniform mixtures	0.858	0.900	0.912	0.922	0.939

frequent number of components in the posterior distribution of the number of components for each of the three models. It shows that results from the three models are quite consistent: the median of posterior π_0 only varies in a small interval from 0.868 to 0.912. In contrast, among other analyses of this dataset, π_0 was estimated to be 0.901 by QVALUE, 0.837 by BUM, 0.830 by SPLOSH, 0.895 by LBE , 0.866 by Langaas and Lindqvist (2005) and 0.830 by Liao et al. (2004). Again, we see a very high degree of agreement between our method, QVALUE, LBE and Langaas and Lindqvist (2005)'s method.

Like the study of Hedenfalk's data, we analyze Callow's data using our method (using model of beta mixtures), Liao's method and Storey's QVALUE respectively. The estimates of lFDR by our method and Liao's method are plotted in Figure 4.7 (a), and the estimates of pFDR by our method and Storey's QVALUE are plotted in Figure 4.7 (b). Again, it shows that the estimates by our method are very similar to these by Liao's method or Storey's QVALUE. The estimated $\hat{F}$ (scaled to unit) from the beta mixtures model of our method and the empirical cumulative distribution F_m is compared in Figure 4.7 (c). They are almost identical, indicating excellent model fitting.

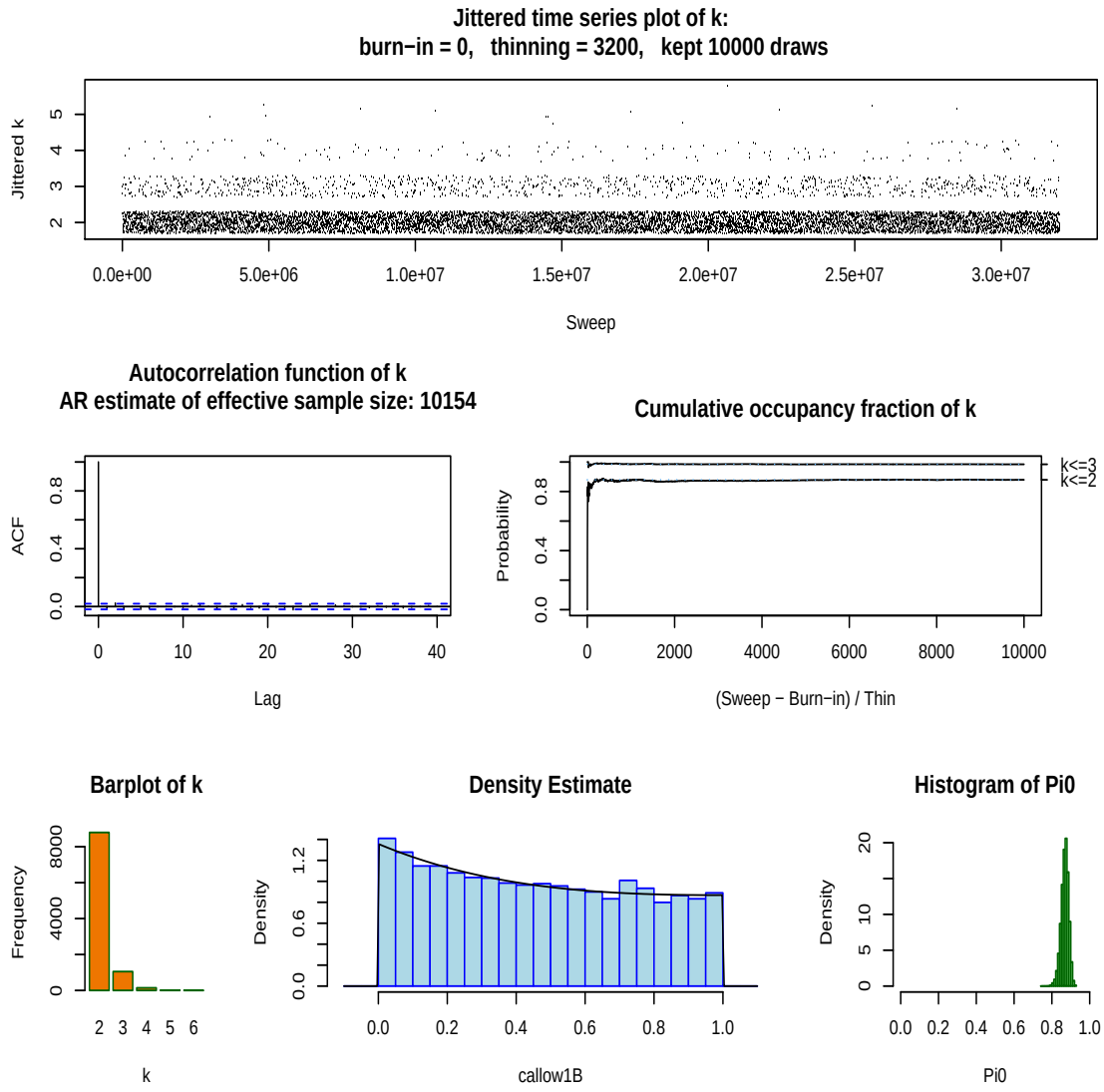


Figure 4.4: Analysis of Callow's lipid metabolism data using the beta mixture distributions. From top to bottom and from left to right, it shows jittered time series plot of k, autocorrelation function of k, cumulative occupancy fraction of k, the plots of posterior predictive distribution imposed on histogram of p-values, the posterior of number of components and the histogram of the posterior π_0 conditional on the number of the most frequent component (i.e. 2 in this case).

4.6.4 A small simulation study

Since our proposed method is very computational intensive, it is infeasible for us to evaluate the performance by running a large simulation study. As an alternative, we apply our proposed beta mixture model to a small number of simulated datasets. The experiment is designed to have $m = 10000$ genes in total, with m_0 of them non-differentially expressed and $m - m_0$ of them differentially expressed. Each of the two comparison groups (e.g. cancer versus normal) has 15 subjects. We generate, for $j = 1, \ldots, 15$,

$$x_{ij}^{[1]} \sim N(0,1), \quad i = 1, \ldots, m,$$

$$x_{ij}^{[2]} \sim N(0,1), \quad i = 1, \ldots, m_0,$$

$$x_{ij}^{[2]} \sim N(\delta,1), \quad i = m_0+1, \ldots, m.$$

The corresponding p-value p_i, $i = 1, \ldots, m$, is computed from the one sided t-test comparing $x_{ij}^{[1]}$, $j = 1, \ldots, 15$ with $x_{ij}^{[2]}$, $j = 1, \ldots, 15$. Sixteen sets of p-values are generated with different combinations of π_0 (i.e. m_0/m) and δ ($\pi_0 = 0.5, 0.6, 0.7, 0.8$; $\delta = 0.4, 0.7, 1.0, 1.3$). Figure 4.5 displays the histograms of the p-values of these 16 datasets.

For each set of p-values, we apply our method and Storey's QVALUE to obtain the posterior distribution of π_0 and the estimate of π_0 respectively. Table 4.5 shows that the results of the two methods are very similar across nearly all the settings. When compared to the true π_0, we find that not only our method but also the popular Storey's QVALUE tend to give estimates very close to the true π_0.

Figure 4.8 shows the plots of the estimates of lFDR against p-values by our

method and Liao's method respectively. Figure 4.9 shows the plots of the estimates of pFDR against p-values by our method and Storey's QVALUE respectively. For the purpose of comparison, we also impose the curves of true lFDR and pFDR in the two figures respectively. Note that the true lFDR and pFDR can be calculated from the dataset given the way of simulation, see Appendix C for details.

For the case of pFDR, our method and QVALUE give very similar estimates throughout all the situations and the estimates from our method seem to be closer to the true pFDR than those from QVALUE. For the case of lFDR, our method also gives quite similar estimates as Liao's method does although there is some small disparity between Liao's estimate and 95% credible interval of our estimate in the 6th, 7th and 15th dataset. In the former two datasets, Liao's estimates are more close to the true lFDR, and in the latter one our estimate is more close to the true lFDR.

From this small simulation study, we show that the performance of our method is satisfactory, and in general our method is able to give nice estimates of pFDR or lFDR similar to other popular methods like QVALUE or Liao's method. Moreover, comparing Figure 4.8 and Figure 4.9, we see the obvious difference between pFDR and lFDR: given a p-value, its corresponding lFDR can be much larger than corresponding pFDR.

Table 4.5: The estimation of π_0 by our method (using model of beta mixtures) and Storey's QVALUE for the 16 simulated data.

Dataset	π_0	δ	π_0 posterior percentiles					π_0 (QVALUE)
			2.5th	25th	50th	75th	97.5th	
1	0.5	0.4	0.445	0.523	0.551	0.567	0.588	0.535
2	0.5	0.7	0.398	0.446	0.476	0.496	0.532	0.495
3	0.5	1.0	0.460	0.480	0.487	0.494	0.505	0.500
4	0.5	1.3	0.479	0.506	0.507	0.509	0.512	0.496
5	0.6	0.4	0.601	0.638	0.651	0.662	0.682	0.646
6	0.6	0.7	0.521	0.596	0.622	0.629	0.640	0.591
7	0.6	1.0	0.513	0.566	0.612	0.617	0.623	0.573
8	0.6	1.3	0.540	0.596	0.598	0.600	0.604	0.565
9	0.7	0.4	0.628	0.683	0.704	0.723	0.751	0.721
10	0.7	0.7	0.683	0.714	0.720	0.725	0.735	0.715
11	0.7	1.0	0.694	0.703	0.706	0.709	0.714	0.708
12	0.7	1.3	0.666	0.696	0.698	0.700	0.703	0.673
13	0.8	0.4	0.699	0.752	0.793	0.844	0.891	0.844
14	0.8	0.7	0.772	0.791	0.799	0.805	0.817	0.793
15	0.8	1.0	0.768	0.783	0.789	0.794	0.803	0.837
16	0.8	1.3	0.779	0.788	0.791	0.793	0.798	0.807

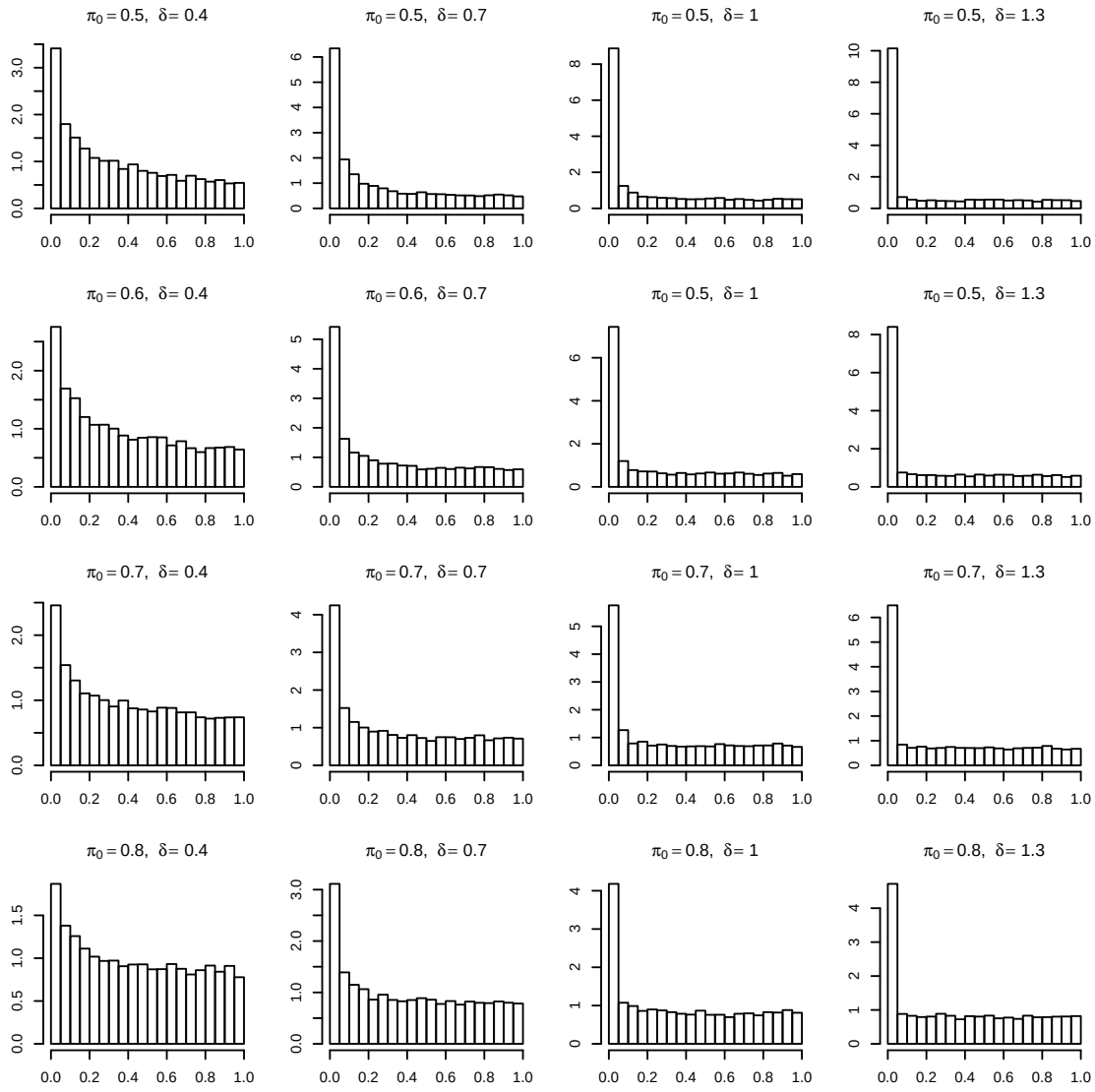
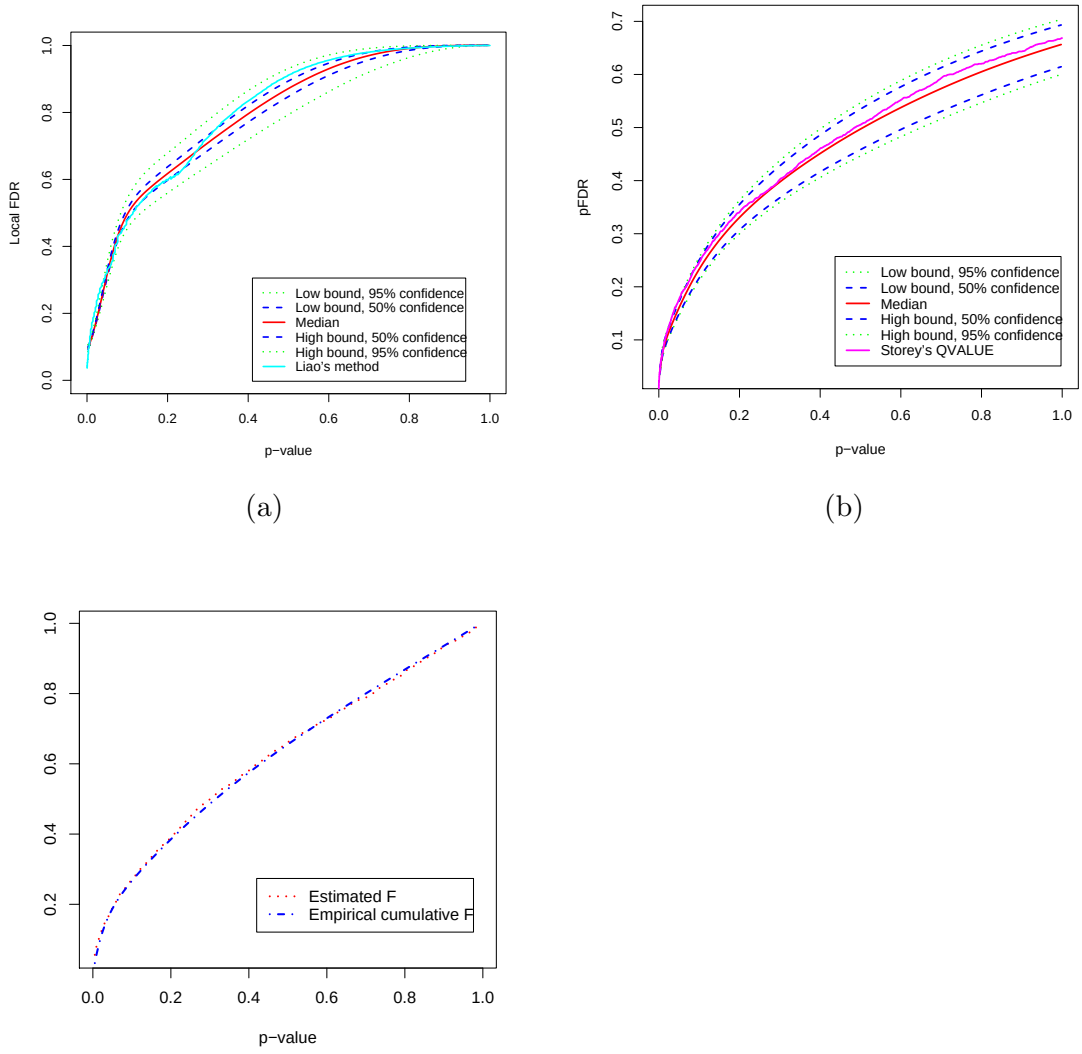


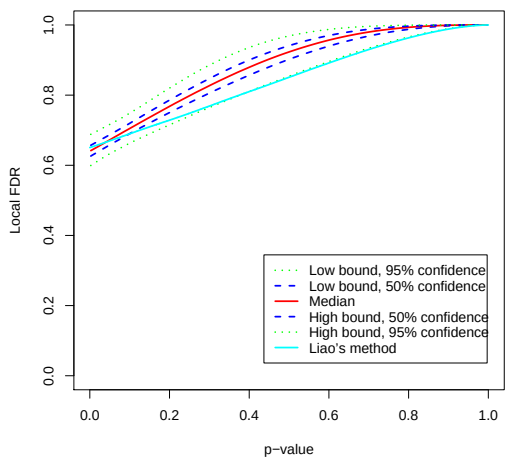
Figure 4.5: The histograms of p-values for 16 simulated datasets. The title of each subfigure indicates the two parameters π_0 and δ used for generating the dataset.



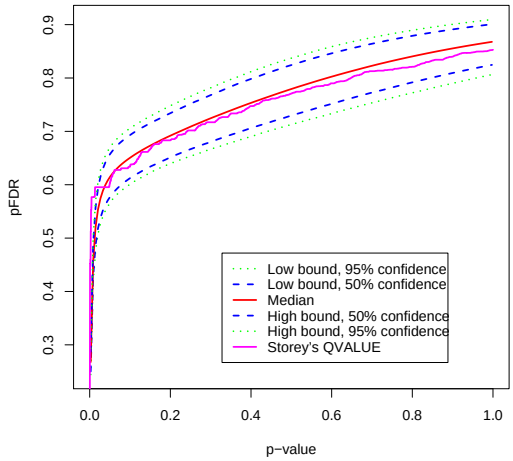
(a) (b)

(c)

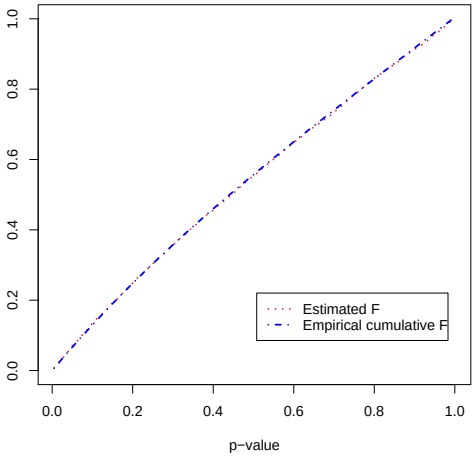
Figure 4.6: Analysis of the Hedenfalk's breast cancer data using the model of beta mixtures. (a) The estimated lFDR (low and high bound of 95% and 50% credible interval and median) and Liao's method. (b) The estimated pFDR (low and high bound of 95% and 50% credible interval and median) and Storey's QVALUE method. (c) The estimated cumulative F from the beta mixture distributions model and the empirical cumulative distribution F from the p-values.



(a)



(b)



(c)

Figure 4.7: Analysis of the Callow's lipid metabolism data using the model of beta mixtures. (a) The estimated lFDR (low and high bound of 95% and 50% credible interval and median) and Liao's method. (b) The estimated pFDR (low and high bounds of 95% and 50% credible interval and median) and Storey's QVALUE method. (c) The estimated cumulative F from the beta mixture distributions model and the empirical cumulative distribution F from the p-values.

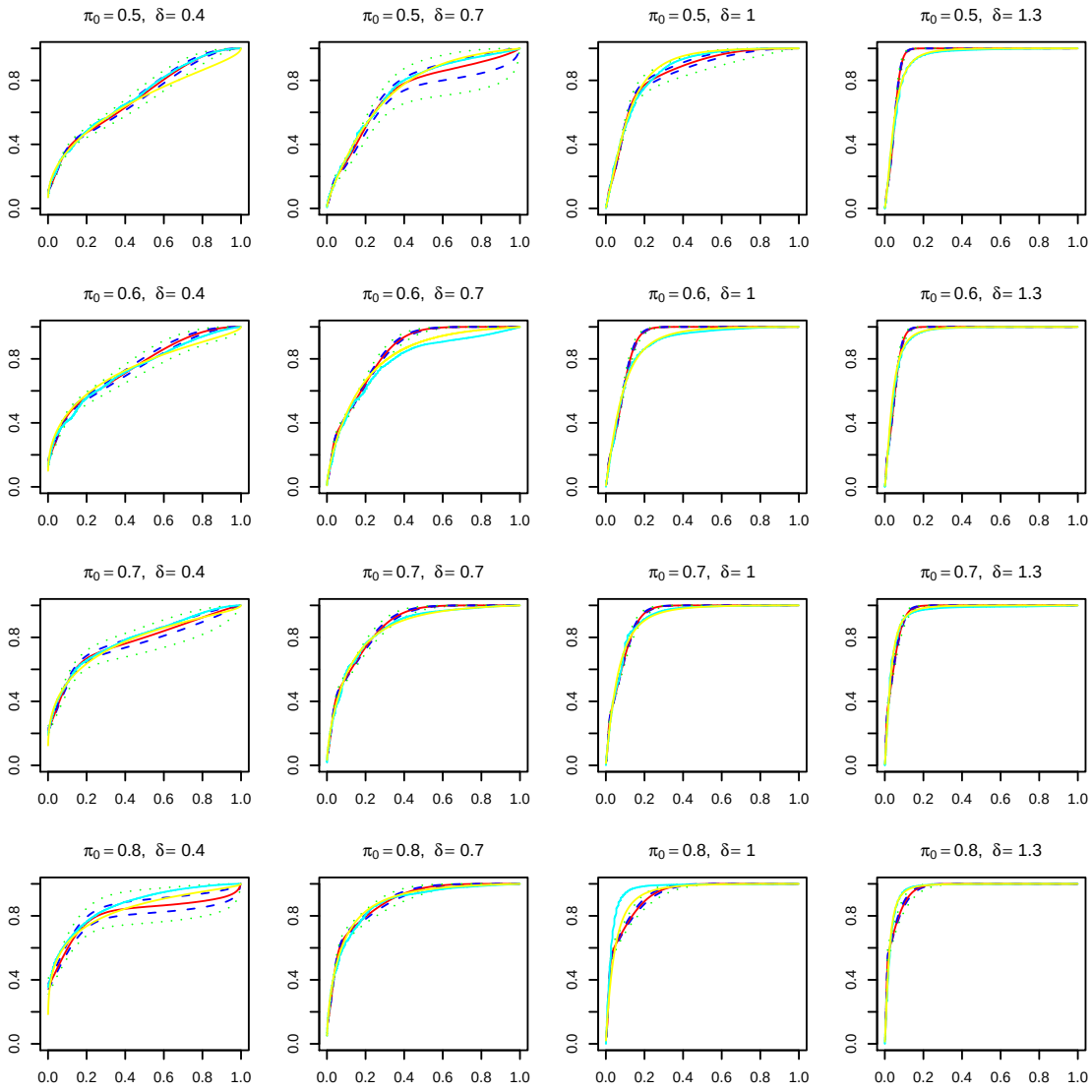


Figure 4.8: The lFDR estimates by our method (using model of beta mixtures) and Liao's method for 16 simulated datasets in Section 4.6.4. The datasets 1-16 are from left to right and from top to bottom. For our method, low and high bound of 95% and 50% credible interval and median of the lFDR estimates are drawn with dash green line, dash blue line and red line. For Liao's method, the lFDR estimates are drawn with light blue line. For the true lFDR, it is drawn with yellow line. The title of each figure indicates the two parameters π_0 and δ used for generating the dataset.

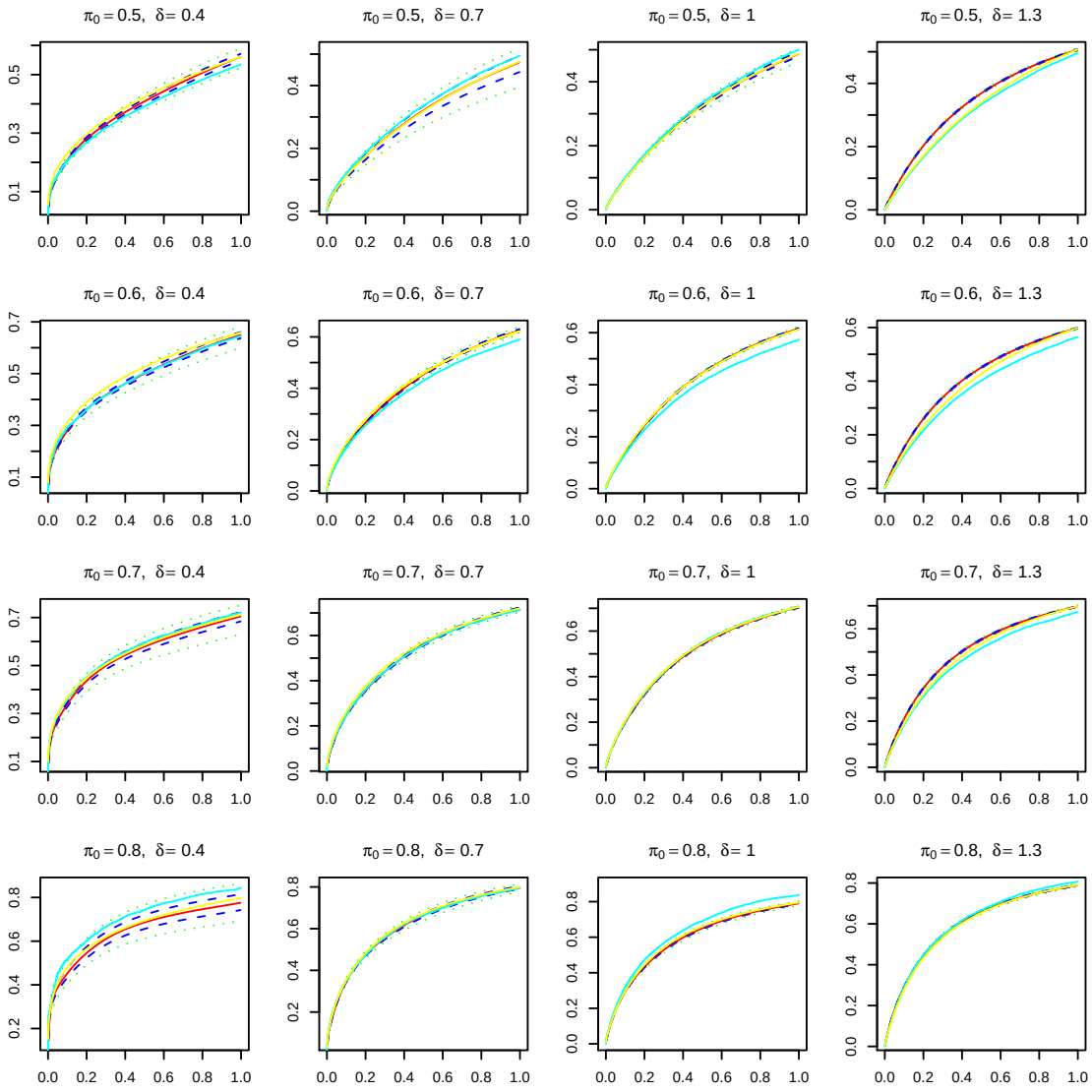


Figure 4.9: The pFDR estimates by our method (using model of beta mixtures) and Storey's QVALUE for 16 simulated datasets in Section 4.6.4. The datasets 1-16 are from left to right and from top to bottom. For our method, low and high bound of 95% and 50% credible interval and median of the pFDR estimates are drawn with dash green line, dash blue line and red line. For Storey's QVALUE, the pFDR estimates are drawn with light blue line. For the true pFDR, it is drawn with yellow line. The title of each figure indicates the two parameters π_0 and δ used for generating the dataset.

4.7 Discussion

The main motivation of this chapter is to provide a tool for accurate estimation of proportion of true null hypotheses π_0 which is a key input value for the calculation of a variety of important error rates for multiple hypothesis testing in microarray experiments. For this purpose, the whole chapter is arranged to have two main parts. In the first part we give a background introduction to the different error rates for multiple hypothesis testing. In the second part we propose three different type of finite mixtures (beta, one-parameter uniform and uniform) with unknown number of components and the first component known to be a uniform distribution to model the distribution of p-values from microarray experiments.

A newly developed MCMC method called the allocation sampler is applied to estimate π_0, lFDR and pFDR in the context of these finite mixture models for both real and simulated microarray gene expression data.

We find that the beta distribution is a more suitable building block than the one-parameter uniform or uniform distribution to approximate the distribution of p-values, because the mixture would involve fewer components and subsequently need less computation time. Also, for the beta mixture model the allocation sampler performs more efficiently and it can achieve much more effective samples given a fixed number of MCMC iterations. Therefore we suggest to use beta mixtures in the Bayesian analysis framework. Modelling the distribution of p-values as finite mixture of beta distributions has been proposed in earlier work by Allison et al. (2002) and Pounds and Morris (2003). The former use a beta mixture model with unknown components while the latter only consider a simple but less flexible two-component beta mixture model. Although our proposed beta

mixture model is the same as that of Allison et al. (2002), our method has its own merit: it can indicate the degree of variation of the estimates by giving their posterior distribution.

Besides the proportion of true nulls (non-differentially expressed genes), the proposed method can also be applied to estimate lFDR and pFDR. Several authors have recently raised the important issue that pFDR (FDR) can give misleading inference when particular genes are of interest. Finner and Roters (2002) discuss cheating with FDR. Suppose that one wishes to reject a particular hypothesis, one can simply group this hypothesis with 99 other hypotheses that are false and will certainly be rejected. The FDR for the family of 100 hypotheses is then no greater than 1/100. Glonek and Soloman (2003) give more realistic examples. In their example one, the pFDR is 0.17 if we reject all the hypotheses with test statistic $Z \geq 2$. Given the test statistic Z in the small proximity of 2, however, the lFDR is a huge 0.99972. All these examples show that the averaging mechanism in pFDR may not be desirable. Suppose that we want to identify genes that show some evidence of differential expression for further biological study. The lFDR quantifies the gene-specific evidence for each gene. The pFDR or FDR, however, averages over other genes with stronger evidence. The lFDR should thus be preferred in such situations.

Take the Hedenfalk's breast cancer data in Section 4.6 as an example. For comparing between BRCA1 or BRCA2 tissues, a total of 319 genes are declared differentially expressed if it is based on pFDR $< 10\%$. The 156 genes among them, however, have lFDR $> 10\%$ and the smaller pFDR values are the result of averaging over genes with stronger evidence for differential expression. Only 163 genes will be declared differentially expressed if it is based on lFDR $< 10\%$. We believe that whether a specific gene should be selected for further biological

investigation should depend on the evidence for that specific gene, not other genes with stronger evidence. Therefore the inference based on lFDR should be preferred, furthermore, the concept of lFDR is easier to understand.

Our method assumes that the genes (p-values) are independent. More sophisticated analysis that takes into account the dependence structure of different groups of genes may be carried out in the future as our knowledge of microarray data accumulates. Nevertheless, our proposed method will remain a useful tool for basic analysis before more complicated modelling is attempted.

Chapter 5

Conclusion and future research

This chapter will summarize the conclusions that can be drawn from the work presented in this thesis. Also, some possible future work will be considered.

As a new powerful tool for generating thousands of gene profiles simultaneously, the cDNA microarray has been a hot research topic in statistical bioinformatics circles. So far most of the research efforts have made to the statistical analysis of gene expression data from the experiments. However, only a small amount of work has been done on the design of cDNA microarray experiments despite the fact that a good experimental design is a must for efficient estimation of the parameters of interest and best use of the limited number of arrays and samples. In this thesis, Chapter 2 deals with four problems related with the optimal designs of cDNA microarray experiments. Section 2.1 gives a general introduction to the issues of cDNA microarray experimental designs, including experimental effects, technical and biological replications, pooling and experimental designs.

Section 2.2 describes an approach for designing optimal microarray experiments considering both technical and biological replicates. For a specific treatment (condition), the gene expression is modelled as the sum of true gene expression plus biological and technical variations. For a whole cDNA microarray experiment involving multiple treatments and arrays, a design matrix can be obtained. An optimality score can be computed from the design matrix given an optimality criterion. Like Wit and McClure (2004), a simulated annealing method is applied to search for optimal or near-optimal designs. We illustrate the approach with two examples. It shows that it is L-optimal for microarray experiments to use as many biological replicates as possible. Also, a dye-swap design is not always L- or D-optimal if both technical and biological replicates are used for each treatment in the experiment.

Section 2.3 argues that factorial experiment design should be considered if the aim of microarray experiments is to study the gene expressions from multiple factors. It suggests using the Q-optimality criterion rather than the L- or D-optimality criterion is a proper choice for the optimal design of factorial microarray experiments. In this section, the gene expression is modelled in a multi-factorial way and a simple example is used to demonstrate how to find the Q-optimal design.

In section 2.4 we discuss the difference between technical and biological variation and explain how pooling samples reduces biological variation of gene expression. We propose an approach to pool samples optimally so that the variance of gene expression value can be minimized given a fixed budget. An practical example is used to study the relationship among the variance of gene expression value, the number of samples in a pool and the ratio of biological variance and technical variance.

In section 2.5 we review the distant pair design which is introduced for the case of the combined study of cDNA microarray gene expression and molecular marker data by Fu and Jansen (2005). We find that the A-optimality criterion for the case of multiple markers proposed in their original paper is not very proper. Therefore, we introduce the gene expression model and suggest an alternative A-optimality criterion for the case of multiple markers. A simple example is used to show that the alterative one is a better choice.

Typically, a cDNA microarray is subject to several artifacts, each of which can compromise the quality of the data. Therefore, these artifacts should be removed before analyzing the data, or else the results would be biased. A dye effect is a major artifact which is non-linear or intensity dependent. To deal with it, (Yang and Speed, 2003) suggest a two-step intensity-dependent normalization method (i.e. LOESS method) by fitting a smooth curve to a scatter plot of Cy5 and Cy3 values in a transformed scale (i.e. MA scatter plot). As an alternative, we propose a new method in Chapter 3. The method is based on an assumption that the dye response function is a "S" curve and can be modelled by functions like the probit function. Since the dye response function describes the relationship between the observed gene expression data and true gene expression data, our method tries to find such a pair of dye response functions (Cy3 and Cy5) so that the resulting dye effect curve matches the dye effect curve from the observed gene expression data. Once a pair of dye response functions is specified, the observed gene expression data can be transformed back to true data. In essence, our method is also a kind of intensity-dependent normalization by fitting a "smooth curve" to a scatter plot of Cy5 versus Cy3 although the "smooth curve" we use is the difference of two dye response functions. The performance studies with simulated and experimental gene expression data show that our method is comparable to the LOESS method.

In a microarray, often thousands of genes are tested simultaneously, against a null hypothesis (expressed or not). When confronted with such a vast number of hypothesis tests and the potential for numerous false positives, the traditional statistical approach is to impose a penalty to account for multiple testing, such as the Bonferroni correction. However that penalty can be far too severe, especially so when it is likely that many of the alternatives are true. To address this problem, quite a few error rates of multiple testing such as false discovery rate (FDR), positive false discovery rate (pFDR) and local false discovery rate (lFDR) have been proposed. To assess or control these multiple error rates, a reliable estimate of the proportion of true null hypotheses π_0 (the proportion of genes that are not differentially expressed) is very important. In Chapter 4 of this thesis, we assume that the p-values from the multiple testing are, unconditionally, independent and identically distributed random variables with mixture density which has a unknown number of components. Three kinds of mixture distributions (beta, uniform and one-parameter uniform) are proposed for approximating the distribution of p-values. A MCMC method called the allocation sampler is applied to estimate π_0 and the mixture density. With these estimates, pFDR and local FDR can be subsequently computed. We demonstrate that the estimates from our method is similar to that from Storey's QVALUE method, and we also claim that when particular genes are of interest local FDR is more specific and relevant than pFDR.

This thesis deals with three areas in statistical analysis of cDNA microarray experiments: optimal experimental design, dye effect normalization and estimation of the proportion of true null hypotheses, pFDR and lFDR. Some future researches could be done in these areas.

- In Chapter 2, we discuss how to find L- or D-optimal design for microarray

experiment with both biological and technical replicates. Readers might criticise these designs and ask: What if an array fails in my experiment? Is a reference design not more robust? To answer these questions we should have a clear definition of a robust optimal design. Consider that if an array fails the corresponding row in the design matrix X is eliminated, then we can think of the design matrix as a random variable X^* by sampling the rows of X with some fixed success rate. Therefore we can try to give a definition of robust: A design X is robust optimal, if X^* maximizes $E(score(X^*))$. In practice, we can estimate the expected value by drawing X^* and calculating the mean of scores for different draws X^*. Following this conceptual extension, some real practical applications can be made. Bailey (2007) defines robust design in a different way: "the measure of robustness is the number of blocks which can be lost". It would be interesting to compare our definition with Bailey's.

- In Chapter 2, we use several simple examples to show that it is optimal to use as many biological replicates as possible. However we are not able to give a mathematical proof to this claim. Is it possible to check the conjecture that "Biological replicates always result in more optimal designs" strictly? Another related and more detailed conjecture is that to prove that

$$\text{Trace}\{X^t(\sigma_t^2 I + \sigma_b^2 ZZ^t)X\}^{-1} \geq \text{Trace}\{X^t(\sigma_t^2 + 2\sigma_b^2)IX\}^{-1},$$

where X is the design matrix, Z is the assignment matrix with exactly one 1 and one -1 in each row.

- In Section 5 of Chapter 2, we consider relatively small values for the number of markers k and arrays n. This allowed us to calculate the optimal

distant pair design by an exhaustive search. However, if k and n grow, the number of possible designs become intractable. This means that we have to resort to an other type of optimization techniques, such as for example, simulated annealing. We plan to implement this in the future. Also, when the number of markers k becomes larger than the number of arrays, our main effect model becomes unidentifiable. In other words, every design is unable to estimate all the effects. In this case, it becomes interesting to consider alternative models, such as for example penalized models. This would involve adding a term $\lambda||\beta||_q$ to the likelihood, where λ is a tuning parameter and $||.||$ is the q-norm of a vector. For different values of $\lambda > 0$ the solution for β becomes tractable again and allow us to find an optimal design. Interesting designs are those designs that for a reasonable range of λ are close to optimal.

- As we mention in the end of Chapter 4, our method depends on the assumption of independence between test statistics. Since this assumption could hardly stand in practice, it is necessary for us to develop more sophisticated methods to deal with the dependence structure between the test statistics. Also, it will be interesting to model different dependence between test statistics when planning the simulation experiment studies.

Appendix A

Computing Σ

We are able to calculate the covariance matrix Σ with some important information of a microarray experiment, such as the random effect design matrix that contains information about the assignment of biological and technical replicates to array.

Let us assume that there is a microarray experiment with m arrays and n conditions, $(m, n > 2)$. We focus on two different arrays, let's say the ith and jth array, each of which involves two samples under different treatments. The ith array corresponds to the a and b sample replicates under the K and L treatments respectively while the jth array corresponds to the c and d sample replicates under the O and P treatments respectively. In the experiment, a and c are labeled with one type of dye while b and d are labeled with the other type of dye.

Based on the gene expression model and the information in the last paragraph, the covariance of gene expressions of the ith and jth arrays is given:

$$\mathrm{Cov}(y_i, y_j) \;=\; \mathrm{Cov}(\delta_{KL} + \epsilon_{Ka} - \epsilon_{Lb} + \eta_i, \delta_{OP} + \epsilon_{Oc} - \epsilon_{Pd} + \eta_j),$$

After ignoring the constant δ and expanding the right side, we have

$$\begin{aligned}\mathrm{Cov}(y_i, y_j) &= \mathrm{Cov}(\epsilon_{Ka} - \epsilon_{Lb} + \eta_i, \epsilon_{Oc} - \epsilon_{Pd} + \eta_j) \\ &= \mathrm{Cov}(\epsilon_{Ka}, \epsilon_{Oc}) - \mathrm{Cov}(\epsilon_{Ka}, \epsilon_{Pd}) + \mathrm{Cov}(\epsilon_{Ka}, \eta_j) - \mathrm{Cov}(\epsilon_{Lb}, \epsilon_{Oc}) \\ &\quad +\mathrm{Cov}(\epsilon_{Lb}, \epsilon_{Pd}) - \mathrm{Cov}(\epsilon_{Lb}, \eta_j) + \mathrm{Cov}(\eta_i, \epsilon_{Oc}) - \mathrm{Cov}(\eta_i, \epsilon_{Pd}) \\ &\quad +\mathrm{Cov}(\eta_i, \eta_j). \end{aligned} \tag{A.1}$$

If we assume that the biological error and technical error are independent from each other, then the covariance of biological errors and technical errors are zero, which reduces the expression of $\mathrm{Cov}(y_i, y_j)$ to a group of covariances of biological errors and covariances of technical errors as follows,

$$\begin{aligned}\mathrm{Cov}(y_i, y_j) &= \mathrm{Cov}(\epsilon_{Ka}, \epsilon_{Oc}) - \mathrm{Cov}(\epsilon_{Ka}, \epsilon_{Pd}) - \mathrm{Cov}(\epsilon_{Lb}, \epsilon_{Oc}) \\ &\quad +\mathrm{Cov}(\epsilon_{Lb}, \epsilon_{Pd}) + \mathrm{Cov}(\eta_i, \eta_j). \end{aligned} \tag{A.2}$$

Since we assume that

$$\mathrm{Cov}(\epsilon_{C_1 k_1}, \epsilon_{C_2 k_2}) = \begin{cases} \sigma_b^2 & \text{if} \quad C_1 = C_2 \quad \text{and} \quad k_1 = k_2 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\mathrm{Cov}(\eta_i, \eta_j) = \begin{cases} \sigma_t^2 & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{cases},$$

we are able to compute the values of $\mathrm{Cov}(y_i, y_j)$ in different situations, some of which are shown in Figure A.1.

Figure A.1(a) describes that the ith and jth arrays involve a common treatment (e.g. treatment K) and under that treatment each of them has the same technical replicate labeled with the same type of dye (e.g. replicate a). In this situation the expression of $\text{Cov}(y_i, y_j)$ in Equation (A.1) is reduced to $\text{Cov}(\epsilon_{Ka}, \epsilon_{Ka})$ which is σ_b^2. Figure A.1(b) describes a similar situation except that the two technical replicates are labeled with different type of dye, in such case $\text{Cov}(y_i, y_j)$ is equal to $-\sigma_b^2$.

Figure A.1(c) describes that the two arrays have two treatments in common. Under one of the treatment both of the arrays has the same technical replicate labeled with the same type of dye, under the other treatment each of the arrays has different technical replicate. In this situation the expression of $\text{Cov}(y_i, y_j)$ is also reduced to $\text{Cov}(\epsilon_{Ka}, \epsilon_{Ka})$ which is σ_b^2. Figure A.1(d) describes a similar situation except that the two technical replicates are labeled with a different type of dye, in such case $\text{Cov}(y_i, y_j)$ is equal to $-\sigma_b^2$.

Figure A.1(e) describes another situation that the two arrays have two treatments in common but have the same technical replicate labeled with the same type of dye under each of the two treatments. In this case the expression of $\text{Cov}(y_i, y_j)$ is reduced to $\text{Cov}(\epsilon_{Ka}, \epsilon_{Ka}) + \text{Cov}(\epsilon_{Lb}, \epsilon_{Lb})$ which is $2\sigma_b^2$. Figure A.1(f) describes a similar situation except that for each of the treatments, the technical replicates are labeled with a different type of dye, in such case $\text{Cov}(y_i, y_j)$ is equal to $-2\sigma_b^2$.

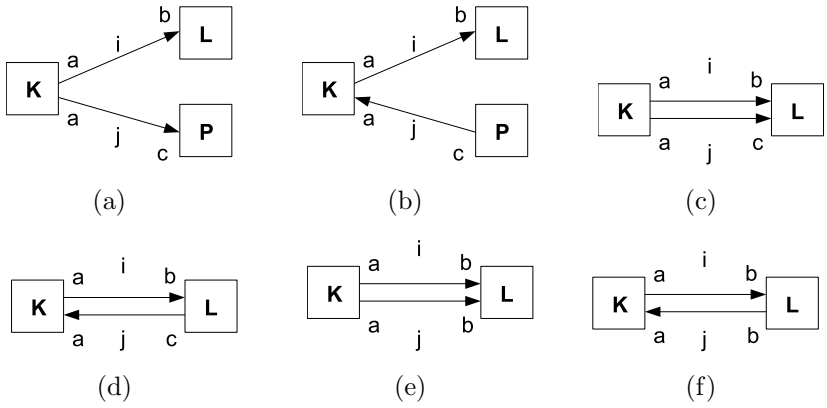


Figure A.1: Directed graphs describe six typical situations involved in computing covariance of gene expressions of the ith and jth microarrays. Each array is represented by an arrow. The head of the arrow indicates that the sample was labeled with Cy5, while the tail represents a sample that was labeled with Cy3. The two experimental treatments of an array are indicated by capital letters, like K, L and P. The sample replicate allocated for each treatment is represented by lowercase letters, like a, b and c. (a) two arrays have one common treatment, under that treatment both of the arrays has the same technical replicate; (b) the same as (a) except the jth array has a different dye assignment; (c) two arrays have two common treatment, under one treatment both of the arrays has the same technical replicate while under the other treatment they have different technical replicate; (d) the same as (c) except that the jth array has a different dye assignment; (e) two arrays have two common treatments and under both treatments the arrays has the same technical replicates; (f) the same as (e) except the jth array has a different dye assignment.

Appendix B

Integrating parameters from the model

This appendix contains the details of the integral

$$f(x^j|\phi_j) = \int \prod_{i \in A_j} f(x_i|\theta_j) f(\theta_j|\phi_j) d\theta_j$$

which defines the marginal distribution of the data x where the prior on the parameters $f(\theta_j|\phi_j)$ is non-conjugate.

B.1 Uniform distribution

Let $f(x_i|\theta_j) = \text{Un}(x_i|0, b_j)$, which is $1/b_j$ for $0 < x_i < b_j$ and 0 for $x_i < 0$ or $x_i > b_j$, then

$$\begin{aligned}\prod_{i\in A_j} f(x_i|\theta_j) &= \frac{1}{b_j^{n_j}} \prod_{i=1}^{n_j} \mathrm{I}_{(0,\infty)}(x_i) \cdot \prod_{i=1}^{n_j} \mathrm{I}_{(-\infty,b_j)}(x_i) \\ &= \frac{1}{b_j^{n_j}} \prod_{i=1}^{n_j} \mathrm{I}_{(x_i,\infty)}(b_j) \\ &= \frac{1}{b_j^{n_j}} \mathrm{I}_{(x_{(n)},\infty)}(b_j),\end{aligned}$$

where I is the indicator function. Then

$$f(x^j|\phi_j) = \int \frac{1}{b_j^{n_j}} \mathrm{I}_{(x_{(n)},\infty)}(b_j) f(\theta_j|\phi_j) d\theta_j.$$

The prior on the parameter $\theta_j = (0, b_j)$ is defined as

$$f(\theta_j|\phi_j) = \frac{1}{\phi_2 - \phi_1}, \quad \phi_1 < b_j < \phi_2,$$

where $\phi = (\phi_1, \phi_2)$ is an known hyperparameter.

Then we calculate $f(x^j|\phi_j)$ as follows,

$$f(x^j|\phi_j) = \frac{1}{\phi_2 - \phi_1} \int_{x_{(n_j)}}^{\phi_2} \frac{1}{b_j^{n_j}} db_j.$$

This integral with respect to b_j has two different cases which need to be examined separately.

When $n_j > 1$,

$$\begin{aligned}f(x^j|\phi_j) &= \frac{1}{\phi_2 - \phi_1} \int_{x_{(n_j)}}^{\phi_2} \frac{db_j^{1-n_j}}{1-n_j} \\ &= \frac{1}{\phi_2 - \phi_1} \frac{\phi_2^{1-n_j} - x_{(n_j)}^{1-n_j}}{1-n_j}.\end{aligned}$$

When $n_j = 1$, (note that $x_{(n)} = x_{(1)}$ here.)

$$\begin{aligned} f(x^j|\phi_j) &= \frac{1}{\phi_2 - \phi_1} \int_{x_{(n_j)}}^{\phi_2} \frac{db_j}{b_j} \\ &= \frac{1}{\phi_2 - \phi_1} \log \frac{\phi_2}{x_{(n_j)}}. \end{aligned}$$

By setting $\phi_1 = 0$ and $\phi_2 = 1$ and collating the above two equations, we have the final result as follows,

$$f(x^j|\phi_j) = \begin{cases} \frac{1-x_{(n_j)}^{1-n_j}}{1-n_j} & n_j > 1, \\ \\ \log \frac{1}{x_{(n_j)}} & n_j = 1. \end{cases} \tag{B.1}$$

B.2 One-parameter Beta distribution

Let $f(x_i|\theta_j) = \beta(x_i|1, b_j)$, whose expression is $b_j(1 - x_i)^{b_j - 1}$ for $0 < b_j < \infty$ and $0 < x_i < 1$. Then

$$\begin{aligned} \prod_{i \in A_j} f(x_i|\theta_j) &= \prod_{i \in A_j} b_j (1 - x_i)^{b_j - 1} \\ &= b_j^{n_j} \left[\prod_{i=1}^{n_j} (1 - x_i) \right]^{b_j - 1} \end{aligned}$$

Independent gamma prior for the parameters b_j is

$$f(b_j) = \frac{\gamma^\alpha}{\Gamma(\alpha)} b_j^{\alpha-1} \exp\{-\gamma b_j\}, \quad j = 1, ..., k.$$

where α is the shape parameter, γ is the rate parameter and Γ is Gamma function. Therefore,

$$\begin{aligned} f(x^j|\phi_j) &= \int \prod_{i \in A_j} f(x_i|\theta_j) f(\theta_j|\phi_j) d\theta \\ &= \int_0^\infty b_j^{n_j} \left[\prod_{i=1}^{n_j} (1 - x_i) \right]^{b_j - 1} \frac{\gamma^\alpha}{\Gamma(\alpha)} b_j^{\alpha-1} \exp\{-\gamma b_j\} db_j \\ &= \frac{\gamma^\alpha}{\Gamma(\alpha) \prod_{i=1}^{n_j} (1 - x_i)} \int_0^\infty b_j^{n_j+\alpha-1} \exp\left\{ - b_j \Big[\gamma - \sum_{i=1}^{n_j} \log(1 - x_i) \Big] \right\} db_j. \end{aligned}$$

Since

$$\int_0^\infty \frac{\Big[\gamma - \sum_{i=1}^{n_j} \log(1 - x_i)\Big]^{n_j+\alpha}}{\Gamma(n_j + \alpha)} b_j^{n_j+\alpha-1} \exp\left\{ - b_j \Big[\gamma - \sum_{i=1}^{n_j} \log(1 - x_i) \Big] \right\} db_j = 1,$$

then we have

$$f(x^j|\phi_j) = \frac{\gamma^\alpha}{\Gamma(\alpha) \prod_{i=1}^{n_j} (1 - x_i)} \frac{\Gamma(n_j + \alpha)}{\Big[\gamma - \sum_{i=1}^{n_j} \log(1 - x_i)\Big]^{n_j+\alpha}}.$$

Let $\alpha = 1$, then the final result is

$$f(x^j|\phi_j) = \frac{\gamma}{\prod_{i=1}^{n_j}(1-x_i)} \frac{\Gamma(n_j+1)}{\left[\gamma - \sum_{i=1}^{n_j}\log(1-x_i)\right]^{n_j+1}}. \tag{B.2}$$

Appendix C

Calculate true pFDR and lFDR

In this appendix we demonstrate how to calculate true pFDR and lFDR from simulated datasets.

Assume that we have the simulated data $X_{ij} \sim N(0, \sigma^2)$ and $Y_{ij} \sim N(\delta, \sigma^2)$ for $i = 1, \ldots, m$, and $j = 1, \ldots, n$, we can compute corresponding p-value, p_j by using one sided t-test. The probability of p_j coming from true null hypotheses is $\pi_0 = m_0/m$ and the probability of p_j coming from false null hypotheses is $1 - \pi_0$. We can write the probability of having a p-value under true null hypotheses smaller than p as follows:

$$\begin{aligned} \Pr_0[P \leq p] &= \Pr_0[1 - F_{0,2(n-1)}(T) \leq p] \\ &= \Pr_0[T \geq F^{-1}_{0,2(n-1)}(1-p)] \\ &= 1 - \Pr_0[T \leq F^{-1}_{0,2(n-1)}(1-p)] \\ &= 1 - F_{0,2(n-1)}[F^{-1}_{0,2(n-1)}(1-p)] \\ &= p, \end{aligned} \tag{C.1}$$

where T is the t statistic and the degrees of freedom for this test is $2(n-1)$. $F_{0,2(n-1)}$ is the cdf of t distribution with $2(n-1)$ degrees of freedom and non-centrality parameter zero. In the same way, we can write out the probability of having a p-value under false null hypotheses smaller than p:

$$\Pr_{\delta^*}[P \leq p] = 1 - F_{\delta^*,2(n-1)}[F^{-1}_{0,2(n-1)}(1-p)], \tag{C.2}$$

where $F_{\delta^*,2(n-1)}$ is the cdf of t distribution with $2(n-1)$ degrees of freedom and non-centrality parameter

$$\delta^* = \sqrt{\frac{n}{2}}\frac{\delta}{\sigma}.$$

Therefore we have

$$\begin{aligned}
\Pr[P \leq p] &= F(p) \\
&= \pi_0 \Pr_0 + (1-\pi_0)\Pr_{\delta^*} \\
&= \pi_0 p + (1-\pi_0)(1 - F_{\delta,2(n-1)}[F^{-1}_{0,2(n-1)}(1-p)]).
\end{aligned} \tag{C.3}$$

If we want to get the density of p, $f(p)$, then we can take derivative on Equation (C.3):

$$f(p) = \pi_0 + (1-\pi_0)\frac{F'_{\delta^*,2(n-1)}[F^{-1}_{0,2(n-1)}(1-p)]}{F'_{0,2(n-1)}[F^{-1}_{0,2(n-1)}(1-p)]}. \tag{C.4}$$

Finally we can plug the expressions of $F(p)$ and $f(p)$ into the following equations to get true value of pFDR and lFDR given a p-value.

$$\text{pFDR}(p) = \frac{\pi_0 p}{F(p)},$$

and

$$\text{lFDR}(p) = \frac{\pi_0}{f(p)}.$$

Bibliography

Allison, D. B., Gadbury, G. L., Heo, M., Fernandez, J. R., Lee, C. L., Prolla, T. A. and Weindruch, R. (2002), 'A mixture model approach for the analysis of microarray gene expression data', *Computational Statistics & Data Analysis* **39**, 1–20.

Bailey, R. A. (2007), 'Designs for two-color microarray experiments, final draft'.

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

Benjamini, Y. and Hochberg, Y. (2000), 'On the adaptive control of the false discovery rate in multiple testing with independent statistics', *J. Educ. Behav. Stat.* **25**, 60–83.

Brown, P. O. and Botstein, D. (1999), 'Exploring the new world of the genome with dna microarrays', *Nat.Genet.* **21(Suppl)**, 33–7.

Byrne, K. A., Wang, Y. H., Lehnert, S. A., Harper, G. S., McWilliam, S. M., Bruce, H. L. and Reverter, A. (2005), 'Gene expression profiling of muscle tissue in brahman steers during nutritional restriction', *J. Anim. Sci.* (83), 1–12.

Caetano, A. R., Johnson, R. K., Ford, J. J. and Pomp, D. (2004), 'Microarray

profiling for differential gene expression in ovaries and ovarian follicles of pigs selected for increased ovulation rate', *Genetics* **168**, 1529–1537.

Callow, M. J., Dudoit, S., Gong, E. L., Speed, T. P. and Rubin, E. M. (2000), 'Microarray expression profiling identifies genes with altered expression in hdl-deficient mice', *Genome Res.* **10**, 2022–2029.

Chen, Y., Dougherty, E. R. and Bittner, M. L. (1997), 'Ratio-based decisions and the quantitative analysis of cdna microarray images', *J. of Biomedical Optics* **4**(2), 364–74.

Chip, S. and Greenberg, E. (1995), 'Understanding the metropolis-hastings algorithm', *Am. Stat.* **49**, 327–335.

Chris, C. A. and Ghazal, P. (2003), 'Combinatorial image analysis of dna microarray features', *Bioinformatics* **19**(2), 194–203.

Churchill, G. A. (2002), 'Fundamentals of experimental design for cdna microarrays', *Nature Gen.* **32**, 490–495.

Cleveland, W. (1979), 'Robust locally weighted regression and smoothing scatterplots', *JASS* **74**, 829–836.

Cleveland, W. and Devlin, S. J. (1988), 'Locally weighted regression: an approach to regression analysis by local fitting', *JASS* **83**, 596–610.

Cox, D. R. and Wong, M. Y. (2004), 'A simple procedure for the selection of significant effects', *J. R. Statist. Soc. B* **66**, 395–400.

Cui, X., Hwang, J., Qiu, J., Blades, N. J. and Churchill, G. A. (2005), 'Improved statistical tests for differential gene expression by shrinking variance components estimates', *Biostatistics* **6**, 59–75.

Dalmasso, C., Broet, P. and Moreau, T. (2005), 'A simple procedure for estimating the false discovery rate', *Bioinformatics* **21**(5), 660–668.

Darvasi, A. (2003), 'Genomics: Gene expression meets genetics', *Nature* **422**, 269–270.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the em algorithm', *JRSS B* **39**, 1–38.

Diebolt, J. and Robert, C. P. (1994), 'Estimation of finite mixture distributions through bayesian sampling', *Journal of the Royal Statistical Society B* **56**, 363–375.

Dudoit, S., Shaffer, J. P. and Boldrick, J. C. (2002), Multiple hypothesis testing in microarray experiments. U. C. Berkeley Division of Biostatistics Working Paper Series. Paper 110.

Dudoit, S., van der Laan, M. J. and Pollard, K. S. (2004), 'Multiple testing. part i. single-step procedures for control of general type i error rates', *Statistical Applications in Genetics and Molecular Biology* **3**, Article 13.

Efron, B. and Tibshirani, R. (2002), 'Empirical bayes methods and false discovery rates for microarrays', *Genetic Epidemiology* **23**, 70–86.

Efron, B., Tibshirani, R., Storey, J. D. and Tusher, V. (2001), 'Empirical bayes analysis of a microarray experiment', *JASS* **96**(456), 1151–1160.

Fearnhead, P. (2004), 'Particle filters for mixture models with an unknown number of components', *Statistics and Computing* **14**, 11–21.

Fearnside, A. T. (2007), Bayesian analysis of finite mixture distributions using the

allocation sampler, PhD thesis, University of Glasgow, Department of Statistics.

Finner, H. and Roters, M. (2002), 'Multiple hypotheses testing and expected number of type i errors', *Ann. Stat.* **30**, 220–238.

Fu, J. and Jansen, R. C. (2005), 'Optimal design and analysis of genetic studies on gene expression', *Genetics* .

Gelfand, A. and Smith, A. (1990), 'Sampling-based approaches to calculating marginal densities', *JASS* **85**, 972–985.

Glonek, G. F. V. and Solomon, P. J. (2004), 'Factorial and time course designs for cdna microarray experiments', *Biostatistics* **5**, 89–111.

Glonek, G. and Soloman, P. (2003), 'Discussion of resampling-based multiple tsting for microarray data analysis by ge, dudoit and speed', *Test* **12**, 1–77.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. and Lander, E. S. (1999), 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring', *Science* **286**, 531–537.

Green, P. J. (1995), 'Reversible jump markov chain monte carlo computation and bayesian model determination', *Biometrika* **82**, 711–732.

Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Kallioniemi, O., Borg, A. and Trent, J. (2001), 'Gene expression profiles of hereditary breast cancer', *N Engl J Med* **344**, 549.

Hochberg, Y. (1988), 'A sharper bonferroni procedure for multiple test of significance', *Biometrika* **75**(4), 800–802.

Hochberg, Y. and Tamhane, A. (1987), *Multiple Comparison Procedures*, Wiley.

Holm, S. (1979), 'A simple sequential rejective multiple test procedure', *Scandinavian Journal of Statistics* **6**, 65–70.

Hseuh, H., Chen, J. J. and Kodell, R. L. (2003), 'Comparison of methods for estimating number of true null hypotheses in multiplicity testing', *J. Biopharm Stat* **13**, 675–689.

Jansen, R. C. and Nap, J. P. (2001), 'Genetical genomics: the added value from segregation', *Trends Genet.* **17**, 388–391.

Jansen, R. C. and Nap, J. P. (2004), 'Regulating gene expression: surprises still in store', *Trends Genet.* **20**, 223–225.

John, J. A. and Mitchell, T. J. (1977), 'Optimal incomplete block designs', *J. R. Statist. Soc. B,* (39), 39–43.

John, J. A. and Williams, E. R. (1995), *Cyclic and Computer Generated Designs, 2nd edn.*, Chapman and Hall-CRC.

Jung, S.-H. (2005), 'Sample size for fdr-control in microarray data analysis', *Bioinformatics* **21**, 3097–3104.

Kerr, M. K. and Churchill, G. A. (2001*a*), 'Experimental design for gene expression microarrays', *Biostatistics* **2**, 183–201.

Kerr, M. K. and Churchill, G. A. (2001*b*), 'Statistical design and the analysis of gene expression microarray data', *Genetical Res.* **77**, 123–128.

Kerr, M. K., Martin, M. and Churchill, G. A. (2000), 'Analysis of variance for gene expression microarray data', *J. Computational Biology* **7**(6), 819–837.

Khanin, R. and Wit, E. (2004), 'Design of large time-course microarray experiments with two channels', *Statistics Department, University of Glasgow, Technical report* **04**(08).

Kraft, P. and Horvath, S. (2003), 'The genetics of gene expression and gene mapping', *Trends Biotechnol.* **21**, 377–378.

Lai, Y. (2007), 'A moment-based method for estimating the proportion of true null hypotheses and its application to microarray gene expression data', *Biostatistics* .

Langaas, M. and Lindqvist, B. H. (2005), 'Estimating the proportion of true null hypotheses, with application to dna microarray data', *J. R. Statist. Soc. B* **67**(4), 555–572.

Liao, J. G., Lin, Y., Selvanayagam, Z. E. and Shih, W. J. (2004), 'A mixture model for estimating the local false discovery rate in dna microarray analysis', *Bioinformatics* **20**(16), 2694–2701.

Lönnstedt, I. and Speed, T. (2002), 'Replicated microarray data', *Statist. Sin.* **12**(31-46).

McLachlan, G. J., Bean, R. W. and Jones, L. B. (2006), 'A simple implementation of a normal mixture approach to differential gene expression in multiclass microarrays', *Bioinformatics* **22**(13), 1608–1615.

McLachlan, G. and Peel, D. (2000), *Finite Mixture Models*, John Wiley & Sons, New York, USA.

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

Newton, M. A., Kendziorski, C. M., Richmond, C. S., Blattner, F. R. and Tsui, K. W. (2001), 'On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data', *J. Computnl Biol.* **8**, 37–52.

Newton, M. A., Noueiry, A., Sarkar, D. and Ahlquist, P. (2004), 'Detecting differential gene expression with a semiparametric hierarchical mixture model', *Biostatistics* **5**, 155–176.

Nobile, A. (1994), 'Bayesian analysis of finite mixture distributions', *Ph.D. dissertation, Department of Statistics, Carnegie Mellon University, Pittsburgh. Available at http://www.stats.gla.ac.uk/˜agostino* .

Nobile, A. (2005), 'Bayesian finite mixtures: a note on prior specification and posterior computation', *Technical Report 05-3, Department of Statistics, University of Glasgow* .

Nobile, A. and Fearnside, A. T. (2007), 'Bayesian finite mixture with an unknown number of components: the allocation sampler', *Statistics and Computing* **17**, 147–162.

Parker, R. A. and Rothenberg, R. B. (1988), 'Identifying important results from multiple statistical tests', *Statist. Med.* **7**, 1031–1043.

Pounds, S. B. (2005), 'Estimation and control of multiple testing error rates for microarray studies', *Briefings in Bioinformatics* **7**(1), 25–36.

Pounds, S. B. and Cheng, C. (2004), 'Improving false discovery rate estimation', *Bioinformatics* **20**(11), 1737–1745.

Pounds, S. B. and Morris, S. W. (2003), 'Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distibution of p-values', *Bioinformatics* **19**(10), 1236–1242.

Richardson, S. and Green, P. J. (1997), 'On bayesian analysis of mixtures with an unknown number of components (with discussion)', *Journal of the Royal Statistical Society B* **59**, 731–792.

Roeder, K. and Wasserman, L. (1997), 'Practical bayesian density estimation using mixtures of normals', *Journal of the American Statistical Association* **92**, 894–902.

Schena, M., Shalon, D. and et al (1995), 'Quantitative monitoring of gene expression patterns with a complementary dna microarray', *Computer Journal* **270**, 467–470.

Schweder, T. and Spjøtvoll, E. (1982), 'Plots of p-values to evaluate many tests simultaneously', *Biometrika* **69**, 493–502.

Smyth, G. K. (2004), 'Linear models and empirical bayes methods for assessing differential expression in microarray experiments', *Statist. Appl. Genet. Molec. Biol.* **3**(1).

Smyth, G. K., Michaud, J. and Scott, H. S. (2005), 'Use of within-array replicate spots for assessing differential expression in microarray experiments', *Bioinformatics* **21**(2067-2075).

Smyth, G. K., Thorne, N. and Wettenhall, J. (2005), 'Limma: Linear models for microarray data, users guide', *Walter and Eliza Hall Institute of Medical Research* .

Steele, R. J., R. A. E. and Emond, M. J. (2003), 'Computing normalizing constants for finite mixture models via incremental mixture importance sampling (imis)', *Tech Report 436, Dept of Statistics, U. of Washington* .

Stephens, M. (2000), 'Bayesian analysis of mixture models with an unknown number of components - an alternative to reversible jump methods', *The Annals of Statistics* **28**, 40–74.

Storey, J. D. (2002), 'A direct approach to false discovery rates', *J. R. Statist. Soc. B* **64**(3), 479–498.

Storey, J. D. (2003), 'The positive false discovery rate: a bayesian interpretation and the q-value', *Anna. Stat.* **31**, 2013–2035.

Storey, J. D., Taylor, J. E. and Siegmund, D. (2004), 'Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: a unified approach', *J. Roy. Stat. B* **66**, 187–205.

Storey, J. D. and Tibshirani, R. (2003), 'Statistical significance for genomewide studies', *PNAS* **100**(16), 9440–9445.

Tanner, M. and Wong, W. (1987), 'The calculation of posterior distributions by data augmentation', *JASS* **82**, 528–540.

Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985), *Statistics Analysis of Finite Mixture Distributions*, John Wiley & Sons, Chichester, UK.

Tsai, P. W., Gilmour, S. G. and Mead, R. (2000), 'Projective three-level main effects designs robust to model uncertainty', *Biometrika* **87**, 467–475.

Tseng, G. C., Oh, M. K., Rohlin, L., Liao, J. C. and Wong, W. H. (2001), 'Issues in cdna microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects', *Nucleic Acids Research* **29**, 2549–2557.

van Laarhoven, P. J. M. and Aarts, E. H. (1987), *Simulated Annealing: Theory and Applications*, Dordrecht: Reidel.

Wernisch, L. (2002), 'Can replication save noisy microarray data?', *Comparative and functional genomics* **3**, 372–374.

Wit, E. and McClure, J. (2004), *Statistics for Microarrays: Design, Analysis and Inference*, John Wiley & Sons, Hoboken, NJ.

Wit, E., Nobile, A. and Khanin, R. (2005), 'Near-optimal designs for dual channel microarray studies', *Appl. Statist.* **5**(54), 817–830.

Wolkenhauer, O., Moller-Levet, C. and Sanchez-Cabo, F. (2002), 'The curse of normalization', *Comp Funct Genomics* **3**, 375C379.

Yang, Y. H., Dudoit, S., Luu, P. and Speed, T. P. (2002*a*), 'Normalization for cdna microarray data', *Technical Report 589, Department of Statistics, University of California at Berkeley* .

Yang, Y. H., Dudoit, S., Luu, P. and Speed, T. P. (2002*b*), 'Normalization for cdna microarray data: a robust composite method addressing single and multiple slide systematic variation', *Nucleic Acids Research* **30**(4).

Yang, Y. H. and Speed, T. P. (2002), 'Design issues for cdna microarray experiments', *Nature Review: Genetics* **3**, 579–588.

Yang, Y. H. and Speed, T. P. (2003), *Design and analysis of comparative microarray experiments, In statistical analysis of gene expression microarray data*, Chapman & Hall/CRC.