Efficiency and Robustness Issues in Complex Statistical Designs for Two–Color Microarray Experiments

Dissertation
zur Erlangung des Doktorgrades
der Mathematisch–Naturwissenschaftlichen Fakultäten
der Georg–August–Universität zu Göttingen

vorgelegt von
Abu Hena M. Mahbub–ul Latif
aus
Bogra, Bangladesh

Göttingen 2005

D7

Referent: Prof. Dr. Edgar Brunner

Koreferent: Prof. Dr. Manfred Denker

Tag der mündlichen Prüfung:

To my family, friends, and mentors.

Abstract

Identifying differentially expressed genes is one of the common goals of microarray experiments. The use of an efficient design in microarray experiments can improve the power of the inferential procedure. Besides efficiency, robustness considerations should also be considered in selecting good microarray designs because missing observations often occur in the microarray experiments. In this dissertation, E–optimality criterion is used as the efficiency criterion and three robustness criteria are proposed to quantify the robustness of a microarray design.

For a given number of available arrays and number of treatment conditions, different microarray designs can be considered. The number of possible designs could be very large and thus a complete analysis of efficiency and robustness considerations could be computationally infeasible. A genetic algorithm based method is suggested for selecting good microarray designs for a set of given research questions. This method can be used to find good designs for both the one–way and two–factor factorial experiments. The use of both the efficiency and robustness criteria in the search procedure is also proposed. As an example, efficient and robust designs for the 3×2 factorial experiments are reported for different numbers of arrays.

Acknowledgements

First of all, I would like to thank my supervisor Prof. Dr. Edgar Brunner for his guidance and supervision throughout the development of this thesis. He has been an excellent mentor and was always available to discuss my work. I am also grateful to my co–supervisor Prof. Dr. Manfred Denker for his helpful suggestions during this research work, especially at the time of crisis.

I want to thank the Department of Medical Statistics and Center for Statistics, University of Göttingen for providing me the financial support during my stay at Göttingen. I am very grateful to Prof. Dr. Anita Schöbel, Dr. Jobst Landgrebe and Dr. Frank Bretz for their constructive suggestions on my research work. Many thanks to my colleagues at the department for supporting me in many ways, especially Karthi for helping proofreading the thesis.

I would like to thank my colleagues at the Institute of Statistical Research and Training, University of Dhaka for their patience and continuous support throughout my seemingly endless study leave.

I am very grateful to my parents and wife Irin for their love, patience, and encouragement during the stressful last three years. One of the best experiences that we lived through in this period was the birth of our daughter Addrita, who brings a lot of happiness to our life.

Abu Hena M. Mahbub-ul Latif

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Microarray technology is one of the most noteworthy innovations in molecular biology and genetics during the last decade or so. It can explore the transcriptional activity of a cell in a rapid and comprehensive way which could bring useful insight for assessing molecular contributors to biological processes. The applications of the microarrays are increasing in recent years and it is very likely that this technology will become a standard tool for clinical diagnostics in near future. Development of statistical methods for analyzing and interpreting microarray expression data is essential because high dimensional microarray data contain a large amount of variations from many sources and the performance of an microarray experiment solely depends on the methods that are used for the analysis.

1.1 Microarray Experiments [*]

Living organisms consist of cells that contain inheritable (genetic) information. The entire genetic content of a cell is termed as genome. This genetic information is used via a process which is called gene expression. The two main steps of gene expression are known as transcription and translation. Transcription is the utilisiation of the genes encoded by the cell's genome to produce messenger ribonucleic acids (*mRNA*). Only a tiny part of the genome is read by the cells during transcription to produce *mRNA* molecules. To a varying extent depending on the organism and the cell type in which transcription takes place, these *mRNA* molecules are used as patterns to produce protein in a process called translation. Proteins are the main carriers of cellular functionality at the molecular level. Because the type and quantity of gene expression at the *mRNA*

[*] The description of the transcriptome analysis methodology is based on a paragraph from Landgrebe and Lübke (2005).

and protein level is the main determinant of cellular identity, function and state, it is interesting to analyze gene expression on both levels. This thesis deals with the analysis of gene expression data acquired on the *mRNA* level. The graphical representation of the gene expression process is given in Figure (1.1).

$$\begin{tikzcd} DNA \arrow[r, "transcription"'] & RNA \arrow[r, "translation"'] & Protein \end{tikzcd}$$

Figure 1.1: Graphical illustration of the conversion of genetic information into proteins.

Techniques for analyzing gene expression on the transcriptional level have been used since the 1970s. Among them, hybridization techniques that use the base pairing property of complementary nucleic acid molecules evolved rapidly since the development of the Northern Blot (Alwine et al., 1977). In this technique a soluble, radioactively labelled *cDNA* probe is hybridized to a separated, membrane–bound (immobilized) *mRNA*–target to detect the size and abundance of one transcript binding to the probe. First steps to reverse the principle of the Northern Blot were undertaken soon: instead of immobilizing the *mRNA*–targets on a membrane and one labelled *cDNA*–probe is hybridized in solution (Northern Blot), multiple *cDNA*–probes were immobilized as spots on a membrane (macroarrays, early 1990s) or on glass (microarrays, late 1990s). Companies developed microarrays with oligo–nucleotide–probes of differing qualities and lengths. Affymetrix produces oligo–arrays with short oligos using in–situ photolithography, while Agilent manufactures long–oligo–arrays with an ink–jet–nucleotide linking technique.

The *mRNA*–targets were labelled with radioactivity (membrane arrays, (Southern et al., 1992)) or fluorescence (glass arrays, (DeRisi et al., 1996)) and hybridized in solution. Using labelled targets in solution and immobilized probes spotted as arrays, the expression of thousands of genes can be monitored at a time by measuring the radioactivity/fluorescence signal at every spot. If radioactive labelling or single–color–oligonucelotide–microarrays (Affymetrix) are used, only one color is available and direct comparisons of two different *mRNA*–targets on a single array can not be performed. With fluorescent labelling, two different fluorescent dyes can be used for different *mRNA*–targets enabling direct comparisons of the targets on one microarray. Because the experimental variance between different arrays is quite high due to varying experimental factors, e.g., labelling efficiency and hybridization quality, direct comparison approaches using the statistical block principle are to be preferred (Kerr and Churchill, 2001a). This thesis deals with only two–color *cDNA* microarray experiments.

A typical two–color microarray experiment has several steps, Eisen and Brown (1999) gave a detailed description of the experimental process of using microarrays. The first step is known as array fabrication in which a set of previously known *cDNA* sequences (probes) are printed onto the arrays using a robotic arrayer. The probes could be of full–length or partially sequenced *cDNA*s which are usually chosen from the available databases (e.g., GeneBank, dbEST, UniGene, etc.). The selection of probes set depends on the experiment, usually genes that are relevant to the biological questions under investigation are selected.

In the second step, total *RNA*s are separately isolated from the pair of competing biological samples (e.g., experimental and control cell type) under investigation. Total *RNA* is usually treated with DNase to remove genomic *DNA* that can inhibit the labelling reaction and lead to increased image background. Total RNA or mRNA is then subjected to reverse transcription in the presence of a fluorescent labelled deoxycytidine (or -uridine) triphosphate and a low-C (or -T)-dNTP mixture. The resulting cDNA contains nucleotides with a fluorescent label. On glass microarrays using fluorescent dyes, two separate labelling reactions with distinct fluorescent dyes (e.g., $Cy3$ and $Cy5$) are used per array.

The third step is known as hybridization in which first, two labelled target *cDNA*s are mixed in equal proportions and then are applied to the array which contains probe *cDNA*s in each spot. If the probe and target *cDNA*s are complementary of each other then they should be bound by their base pairs and the strength of the binding depends on the amount of the gene expression in the target samples. For example, if a gene (spotted on the array) is more expressed in the experimental cell (labelled with $Cy3$) than in the control cell then $Cy3$–molecules should bind more to that array spot compared to the $Cy5$–molecules. After sufficient time is allowed for this competitive hybridization, the array is carefully washed a number of times so that all the unbound target *cDNA*s are washed off. The next steps of a microarray experiment are image analysis and data extraction (Yang et al., 2002a).

In image analysis, a confocal laser microscope is used to scan the array at two channels or wavelengths, one for the $Cy3$ flourscent–tagged sample and another for the $Cy5$ flourscent–tagged sample. This procedure generates two 16–bit tagged image file format (tiff) images corresponding to two samples under investigation. These tiff images are considered as the 'raw' data for a microarray experiment. The measurement of the flourscent intensities for different probes can be obtained from the tiff images by using an image analysis software (e.g., QuantArray, Spot, etc.). The ratio of the fluorescence intensities for each spot indicates the relative abundance of the corresponding gene in the two samples under investigation. A graphical representation of different steps of microarray experiments are shown in Figure 1.2.

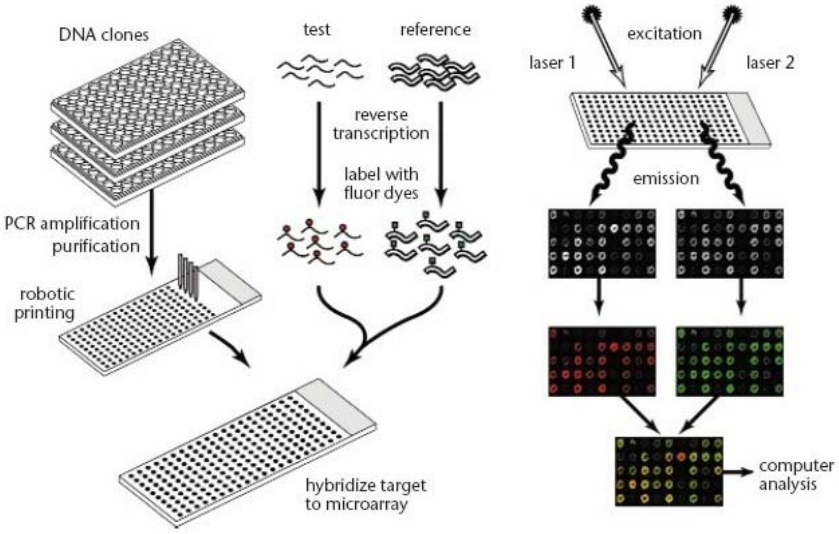


Figure 1.2: A graphical representation of the different steps of a two–color microarray experiment (Duggan et al., 1999).

Statistical methods can play vital roles in different stages of microarray experiment (Smyth et al., 2003). Techniques of design of statistical experiments can be used to decide which treatments are to be hybridized on the arrays and on how many arrays the hybridizations will be replicated (e.g., Kerr and Churchill, 2001b; Churchill, 2002). The raw intensity measurements must be normalized to adjust for any systematic biases that may arise due to the effects other than the treatment effects under investigation (e.g., Yang et al., 2002b; Huber et al., 2002; Smyth and Speed, 2003). The common goals of microarray data analysis include identifying differentially expressed genes (e.g., Dudoit et al., 2002b; Wolfinger et al., 2001; Newton et al., 2001), classifying genes into pre–existing or new meaningful classes (e.g., Eisen et al., 1998; Dudoit et al., 2002a), etc. Among the statistical methods that are used in different stages of microarray experiment, design of valid and efficient microarray experiments will be addressed in this dissertation.

1.2 Statistical Designs in Microarray Experiments

The objective of the experimental design is to make the analysis of the data and the interpretation of the results as simple as possible, given the purpose of the experiment and constraints of the experimental materials (Yang and Speed, 2002). A carefully designed experiment could efficiently use the available materials and estimate the effects of interest with high precision. On the other hand, a badly designed experiment could bring bias to the estimates or the effects may be non–estimable with the data that have been collected.

Experimental designs that are commonly used in microarray experiments can be classified into two broad categories on the basis of whether direct comparisons between the competing *RNA* samples (treatments hereafter) are made, i.e., whether the treatment comparisons are made within or between arrays. The common reference (*CR*) design (Callow et al., 2000) is the most commonly used microarray design where treatments are compared indirectly via a common reference sample. In *CR* design, the labeling strategy is often fixed for all the arrays, e.g., if the reference sample is labeled with a $Cy5$ dye then treatments are labeled with a $Cy3$ dye for all the arrays. Since all the treatments are labelled with a single dye, the *CR* can avoid the bias that usually arise due to the differences in the ability of the two dyes in binding to the spotted *cDNA* probes. There are several drawbacks of the *CR* design (Kerr and Churchill, 2001b). First, half of the information is not used to estimate the parameters of interest because the information from the reference sample is not of interest itself. Second, the indirect comparisons inflates the variance of the relevant parameter estimates. Third, the use of different reference samples places a strong constraint on the potential for comparing the data generated by different researchers (Jin et al., 2001).

Among the microarray designs that consider direct treatment comparisons, in dye–swap (*DS*) designs (Yang and Speed, 2002) each hybridization is done twice, with the dye assignments reversed in the second hybridization. The *DS* design is useful for reducing the systematic dye bias. This design is also known as saturated design because all possible pairwise treatment comparisons are made for this design. The main disadvantage of *DS* design is that the number of arrays could be very large if a large number of treatments is to be compared. To have a reasonable number of arrays per replication of a design, Kerr and Churchill (2001b) proposed another class of microarray designs which can also be used to make direct comparisons of the treatments. This class of designs is known as loop designs where the treatments are balanced in terms of the dye bias, i.e., each treatment is labeled once with a red and a green dye, respectively.

Using the same number of arrays as a *CR* design, the comparable loop design collects twice as much as data on the treatments under investigation and hence, provides more degrees of freedom for estimating error. Unlike the *DS* designs, not all the pairs of the treatments are hybridized for the loop designs, but each of the pair of treatments are connected sequentially. Landgrebe et al. (2004) studied some basic and composite designs for one–way and multi–factor factorial experiments.

The choice of an appropriate microarray design depends on both scientific and logistic issues (Yang and Speed, 2002). Among the scientific issues, the aim of the experiment needs to be addressed first, i.e., whether it is to identify differentially expressed genes, to search for a specific gene–expression pattern, or to identify a tumor subclass. The research questions need to be specified and it must be stated whether some questions are more important than the others. The amount of the available *RNA* is also important because the maximum number of possible hybridizations with a specific treatment depends on the corresponding amount of *RNA*. Moreover, details of sample isolation, *RNA* extraction, and labelling also affect the number of times the experiment has to be repeated. Kerr and Churchill (2001b) recommended to select microarray designs which are balanced with respect to the dye bias and can be used to estimate the effects of interest with less variance compared to the other competing designs, i.e., efficient design. Kerr (2003) discussed different design considerations for efficient and effective microarray studies.

The methods of analyzing microarray data is another scientific issue that could play an important role in selecting good microarray designs. We first describe a few statistical models that are used for modeling microarray expression data. One of the main objectives of a statistical model is to estimate the treatment effects after adjusting for all known systematic biases. So to assume a statistical model for microarray expression data, the sources of variations in the fluorescent measurements need to be studied first. The sources of variations in microarray data are yet to be completely understood which could be ranging from the hybridization to the ozone level of the laboratory. Schuchhardt et al. (2000) gave a detailed description of the possible sources of variation in microarray data. Kerr and Churchill (2001b) assumed the main sources of systematic variations in microarray expression data are due to the dyes, the arrays, the treatments, and the genes. They proposed a global *ANOVA* model for microarray expression data where all the main effects and interactions are assumed to be fixed. The primary effect of interest of such a model is the interaction between gene and treatment which indicates the effect of a treatment in different genes. Lee et al. (2002) described a two–stage approach to fit the global *ANOVA* model, where in the first stage, the gene independent parameters are estimated and the resulting residuals are then used as response for the second stage. The analysis in the first stage is similar to the normalization of the microarray data (Yang et al., 2002b; Huber et al., 2002). The second stage analysis is often done by gene basis, i.e., one model is assumed for each gene (Landgrebe et al., 2004). Wolfinger et al. (2001) considered a linear mixed effects model for analyzing microarray expression data where a main effect of array and interaction between array and treatment, and array and gene are assumed as random. Their model is similar to the global *ANOVA* model but does not include a dye effect.

Kerr and Churchill (2001b) made the connection between microarray experiments and classical incomplete block designs (Cochran and Cox, 1992). In microarray experiments, two differentially labelled *cDNA*s are hybridized together on each array. The arrays can be treated as the experimental blocks with block size two. If more than two treatments are to be compared not all the treatments can appear in the same array. Experimental problems of this nature have been studied in agricultural experiments since early in the last century and the designs that can be used in such cases where block size is smaller than the number of treatments are known as incomplete block designs. A microarray experiment can be considered as an incomplete block design if more than two treatments are to be compared and no gene–specific dye effect is assumed. One of the objectives of the usual normalization step of the microarray data analysis is to adjust the dye bias. In some recent studies (Landgrebe et al., 2004; Dobbin et al., 2003b; Kerr, 2003), it has been shown that the usual normalization methods (Yang et al., 2002b) can only adjust the overall dye effects, but not the gene–specific one. They advocated to include the dye effect in the gene–specific models, i.e., interaction between gene and dye should be taken into account. In this experimental situation, blocking factors are used from two directions (dye and array) and in statistical literature such experimental designs are known as row–column designs (Shah and Sinha, 1989). The difference between an incomplete block design and a row–column design is shown in Table 1.1 in the context of microarray experiments.

Array 1	Array 2	Array 3
A	B	A
B	C	C

(a) Incomplete block design

Dye	Array 1	Array 2	Array 3
$Cy3$	A	B	C
$Cy5$	B	C	A

(b) Row–column design

Table 1.1: Hybridization protocols of three arrays for comparing treatments A, B, and C. In (a), within an array a dye (say, $Cy3$) can be used to any of the treatments, but in (b) $Cy3$ can only be used to the treatments of the first row.

The problem of selecting an efficient incomplete block design for block size two has been studied extensively. In practice, efficiency criteria are used to assess the quality of a design in estimating the effects. Most of the common efficiency criteria (e.g., E–, A–, D–optimality) are defined as a function of the eigenvalues of the corresponding information or dispersion matrix (Pukelsheim, 1993). John and Mitchel (1977) defined regular graph designs and conjectured that efficient block designs can be found among the regular graph designs if they exist. Bagchi and Cheng (1993) proposed a class of highly efficient regular graph designs of block size two. The optimal designs that are suggested in the literature on incomplete block designs have little practical importance in the context of microarray experiments. This is because the underlying strategy for these studies is to define families of optimal designs. But in microarray experiments, experimenters are more interested in the designs by which the effects of interest can be estimated most efficiently with the available number of arrays. The effects of interest could be different for different studies and some effects could be more important than others.

So far there have not been many attempts on selecting good microarray designs, but inefficiency of the CR designs compared to the loop designs are mentioned in several studies. Kerr and Churchill (2001b) extensively studied the properties of the common reference and loop designs for the one–way factorial experiments. They suggested A–optimal designs for K, $K+2$, and $2K$ arrays when the number of treatments K is not too large. Yang et al. (2002b) used an A–optimality criterion to select efficient designs for the time–course and multi–factor factorial experiments. Landgrebe et al. (2004) showed a procedure for selecting good microarray designs from a set of basic designs by using an E–optimality criterion. Glonek and Solomon (2004) used the concept of admissibility in selecting good microarray designs. This approach can be used when more than one effect is of interest.

Microarray expression data often contain missing observations (Troyanskaya et al., 2001; Nguyen et al., 2004) due to various reasons including insufficient resolution, image corruption, dust or scratches on the array, excessive background noise, array fabrication error, etc. As the number of spots on the array increases to accommodate the entire gnome, the occurrence of such missing observations will tend to increase (Khan et al., 2003). Analysis of data with missing observations is particularly important in microarray context because repeating the experiments is not possible due to a limited quantity of materials and for budget constraints. Two main approaches have been used to deal with missing observations, which are : (i) analyze data after excluding missing observations, (ii) estimate missing values before the analysis. Approach (ii) is not the focus of this dissertation. The methods we are dealing with for analyzing microarray data can handle missing observations to some extent. We are interested in examining the loss of information due to missing observations. The designs for which this loss is small are known as robust (Dey, 1993). More specifically, robustness is a design consideration which indicates the ability of a design to estimate the effects of interest in the presence of missing observations. Robustness is a relatively new topic in microarray analysis and is briefly introduced in some recent papers (e.g., Kerr, 2003; Churchill, 2002; Simon et al., 2002). So far, no attempts have been made to quantify the robustness of a design in the microarray context. Besides efficiency, robustness could play an important role in selecting good designs for a given set of research questions.

1.3 Objectives

The objective of this dissertation is to provide an improved method to find efficient and robust microarray designs. The main points of the work include the following:

- To quantify the loss of information due to missing observations, three robustness criteria are proposed in the context of microarray experiments.

- A procedure to find good microarray designs from a set of candidate designs is suggested. The method uses both the efficiency and robustness criteria in evaluating the designs.

- A computer program is written in **R** (R Development Core Team, 2004) which can be used to find good designs for given research questions and a pre–specified number of available arrays.

This dissertation is organized as follows. In §2, the assumed model for analyzing microarray expression data is described and also the efficiency and robustness criteria are specified. The importance of using efficient design in microarray experiments are shown by using a simulation study. As an example, methods for selecting efficient and robust designs for a given experimental layout and set of research questions are shown in §3. In §4, a genetic algorithm based search procedure is developed which can be used for selecting efficient and robust microarray designs for both one–way and multi–factor factorial experiments. In §5, the efficient and robust designs for the 3×2 experimental layout are reported for different numbers of arrays. The performance of the proposed method is validated by simulation studies.

Chapter 2

Efficiency and Robustness Criteria for Microarray Designs

2.1 Introduction

Statistical design of microarray experiments plays a vital role in allocating *mRNA* samples under investigation to the arrays. The application of classical experimental designs to microarrays was first investigated by Kerr et al. (2000). Microarray experiments can be considered as incomplete block experiments of block size two when more than two treatments are of interest (Kerr and Churchill, 2001b). Among the experimental designs used in microarrays, the common reference (*CR*) design (Callow et al., 2000) is the most commonly used one where the treatments under investigation are compared indirectly via a common reference sample. Kerr and Churchill (2001b) proposed loop designs which compare the treatments of interest directly by connecting every pair of treatments sequentially. In this dissertation, we consider different types of the loop designs and we call the simple loop design as circular loop (*CL*) design. Another important design is the dye–swap (*DS*) design which compares each pair of the treatments twice with a forward and a reverse dye labelling. Landgrebe et al. (2004) suggested some basic and composite microarray designs for two–factor factorial experiments.

Several designs can be considered for a specific microarray experiment. The choice of the design depends, among other things, on its performance in estimating the effects of interest. It is desirable to use the design which can estimate the effects with maximum efficiency. Efficiency criteria are used to assess the quality of a design with respect to the estimates of the effects of interest. So far, different efficiency criteria have been proposed in the microarray literature to select designs for microarray experiments. Kerr and Churchill (2001b) were the first to discuss the procedure for comparing microarray designs for one–way experimental layouts. Yang and Speed (2002) considered 2×2 factorial experiments to compare the efficiency of the loop designs with the common reference designs. They did not assume the gene×dye interaction in their model. Glonek and Solomon (2004) considered a similar model to Yang and Speed (2002) and suggested to select efficient designs from the class of admissible designs. Landgrebe et al. (2004) included the gene×dye interaction in their model and used a minimax approach to select efficient microarray designs for both one–way and multi–factor factorial experiments.

All of the above investigations used complete observations to estimate the efficiency of the designs to be compared. However, microarray expression data often contain missing observations due to various reasons including image resolution, image corruption, dust or scratches on the array, etc. (Troyanskaya et al., 2001). For a given experimental question, an efficient design could break down due to missing observations. So, besides efficiency, considerations of the robustness properties of the candidate designs could be useful in selecting good microarray designs. By robustness, we mean the property of a design that shows its ability to estimate the effect of interest in the presence of missing observations. The importance of the robustness issues has been stressed in recent papers in the context of microarray experiments (Kerr, 2003; Churchill, 2002; Simon et al., 2002), but till date no attempts have been made to use the robustness in selecting designs for microarray experiments.

The main objective of this chapter is to formalize different efficiency and robustness criteria in the context of microarray experiments. The linear statistical model that we assume for analyzing microarray data is described in §2.2. Three robustness criteria, namely, breakdown number, average efficiency, and proportion of the effective designs are suggested in §2.4. In §2.5, a simulation study is performed to show the consequences of using an inefficient design instead of efficient ones for finding differentially expressed genes.

2.2 Linear Models for Microarray Expression Data

Let n denote the number of available arrays, G denote the number of genes that are spotted on each array, and K be the number of treatments under investigation. Let y_{ijkg} be the log–transformed intensity measurement corresponding to the array i, dye j, treatment k, and gene g. Kerr and Churchill (2001b) extensively studied the relevant sources of variations in the microarray expression data and identified the variations due to the arrays, dyes, treatments, and genes as the major sources. They proposed the following global *ANOVA* model for the log–transformed intensity measurement y_{ijkg}:

$$y_{ijkg} = \mu + \alpha_i + \theta_j + \beta_k + \gamma_g + (\alpha\gamma)_{ig} + (\theta\gamma)_{jg} + (\beta\gamma)_{kg} + \epsilon'_{ijkg}, \tag{2.1}$$

where μ denotes the overall mean, α, θ, β, and γ correspond to the main effects of array, dye, treatment, and gene, respectively and $(\alpha\gamma)$, $(\theta\gamma)$, and $(\beta\gamma)$ represent the two–factor interaction corresponding to gene with array, dye, and treatment, respectively. In microarray studies, the effects of interest are the interactions between the gene and treatment which measure the differentials in the gene expressions across different treatments. All the main effects and interaction are assumed to be fixed and the random error term ϵ'_{ijkg} is assumed to be independently distributed with mean 0 and variance σ'^2. Throughout of this thesis, we assume that each gene is spotted only once on each array.

In principle, the least squares estimates of the parameters of the model (2.1) should be obtained by using existing common statistical packages. In microarray data, the number of genes is often very large (typically in thousands). Hence, the number of parameters of the model (2.1) and the dimension of the corresponding model matrix could be very large. The space constraints of the common statistical packages may cause problem for using the usual least square routine to estimate the parameters of the model like (2.1). To overcome this problem, Lee and Whitmore (2002) suggested a two–stage approach to fit the model (2.1) which is simple and effective. In the first–stage model, the gene–specific terms of the model (2.1) are absorbed in the error term η_{ijkg}, i.e.,

$$y_{ijkg} = \mu + \alpha_i + \theta_j + \beta_k + \eta_{ijkg}. \tag{2.2}$$

This model is simpler compared to the global *ANOVA* model (2.1) because it has a small ($= 1 + n + 2 + K$) number of parameters and the corresponding least squares estimates can easily be obtained by using existing statistical packages. Estimation of the parameters of the first–stage model (2.2) can be viewed as a normalization step of microarray data analysis where the systematic biases due to other than the treatment effects are adjusted. The estimated residuals of the first–stage model (2.2) are used as the response of the second–stage model. Instead of residuals of the model (2.2), normalized log–intensities corresponding to the two channels (e.g, Huber et al., 2002) can also be used as the response of the second–stage model. The second–stage model can be written as

$$\hat{\eta}_{ijkg} = \gamma_g + (\alpha\gamma)_{ig} + (\theta\gamma)_{jg} + (\beta\gamma)_{kg} + \epsilon'_{ijkg}, \tag{2.3}$$

which contains all the gene–specific parameters of the global *ANOVA* model (2.1). Under the assumption that gene expressions are independent of each other, the parameters of the second–stage model (2.3) can be estimated independently for each gene. In microarray analysis, the difference of the log–intensities corresponding to two dyes (i.e., treatments) is the measurement of interest. Assume the treatments k and k' are hybridized to the probes on the array i where the former is labelled with a green ($Cy3$) dye ($j = 1$) and the latter with a red ($Cy5$) dye ($j = 2$). For a specific gene g, the relative expression of the array i can be expressed as

$$\begin{aligned} z_i &= \hat{\eta}_{i1k} - \hat{\eta}_{i2k'} \\ &= (\theta\gamma)_1 - (\theta\gamma)_2 + (\beta\gamma)_k - (\beta\gamma)_{k'} + \epsilon'_{i1k} - \epsilon'_{i2k'} \\ &= \delta_1 - \delta_2 + \tau_k - \tau_{k'} + \epsilon_i, \end{aligned} \tag{2.4}$$

where $\delta_j = (\theta\gamma)_j$, $\tau_k = (\beta\gamma)_k$, and $\epsilon_i = \epsilon'_{i1k} - \epsilon'_{i2k'}$ are defined to simplify the notation. Without loss of generality, the gene–specific subscripts are excluded from the model (2.4) because genes are separately modeled, i.e., the model of this type can be assumed for each gene $g = 1, 2, \ldots, G$. The model (2.4) has a smaller number of parameters compared to the model (2.3) because array–specific parameters are canceled out during the computation of the relative expression. The output of the lowess regression based normalization methods (Yang et al., 2002b) can also be used as the response in the model (2.4).

Let $\mathbf{Z} = (z_1, z_2, \ldots, z_n)'$ be the vector of the relative expressions corresponding to a specific gene. Each array contributes one measurement to the vector $\mathbf{Z}$. In matrix notation, the model (2.4) can be written as

$$\mathbf{Z} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.5}$$

where $\mathbf{X}$ denotes the $n \times (K + 2)$ design matrix with rank($\mathbf{X}$) $< \min(n, K + 2)$, $\boldsymbol{\beta} = \big(\delta_1, \delta_2, \tau_1, \ldots, \tau_K\big)'$ denotes the $(K+2)$–dimensional vector of parameters, and $\boldsymbol{\epsilon}$ denotes the n–dimensional vector of independent random errors with mean 0 and variance σ^2 ($= 2\sigma'^2$). The parameter vector $\boldsymbol{\beta}$ contains the fixed dye effects δ_1, δ_2 and the treatment means $\tau_1, \ldots, \tau_K$. The model of the type (2.5) is known as a non–full rank model in the classical linear models literature (Searle, 1971, §5) because the corresponding moment matrix $\mathbf{X}'\mathbf{X}$ is not of full rank.

The dye effects are included in our gene–specific *ANOVA* model (2.4) because the standard normalization procedures (e.g., Yang et al., 2002b; Lee et al., 2002) can only adjust the overall dye–effects, but not the gene–specific dye effects. The gene–specific dye bias is displayed by the genes that do not fall into the overall pattern of the dye effect that characterizes the majority of the genes (Dobbin et al., 2003a). In some recent papers (e.g., Landgrebe et al., 2004; Dobbin et al., 2003a; Kerr, 2003), it was pointed out that even using the normalized data, the dye effects could be significant for some of the genes. Thus, we have included dye effects in the gene–specific *ANOVA* model (2.4).

2.2.1 Contrast Matrix

The types of research questions could be different for different experimental layouts, e.g, pair wise or many–to–one treatment comparisons could be of interest for the one–way factorial experiments, whereas in multi–factor factorial experiments, combinations of the simple effects, main effects, or interaction are often seen as the effects of interest. In practice, the experimental question of interest can be expressed in terms of a vector of linear functions of the regression parameters $\boldsymbol{\beta}$, e.g., $\mathbf{C}'\boldsymbol{\beta}$, where $\mathbf{C}$ denotes a $(K+2)\times d$ contrast matrix and the value of $d(\geq 1)$ depends on the type of experimental question. A matrix $\mathbf{C}$ is said to be a contrast matrix if and only if $\mathbf{C}'\mathbf{1}_d = \mathbf{0}_d$, where $\mathbf{1}_d$ and $\mathbf{0}_d$ are the d–dimensional vectors with all elements equal to 1 and 0, respectively.

As an example, consider a microarray design for a 1×3 experimental layout where the treatment of interest is investigated under three different conditions, i.e., $K = 3$ for this example. Assume the gene–specific *ANOVA* model (2.5) for the analysis. The corresponding vector of the parameters can be written as

$$\boldsymbol{\beta} = \big(\delta_1, \delta_2, \tau_1, \tau_2, \tau_3\big)',$$

where δ_j denotes the j^{th} dye effect and τ_k denotes the k^{th} treatment effect, $j = 1, 2$, $k = 1, 2, 3$. Different contrast matrices can be considered for defining different treatment effects, e.g., the function

$$\mathbf{C}_1'\boldsymbol{\beta} = \big(0, 0, 1, -1, 0\big)\boldsymbol{\beta} = \tau_1 - \tau_2$$

compares the first treatment with the second treatment where $d = 1$. Similarly, if one is interested only in the dye effect then the corresponding linear function would be:

$$\mathbf{C}'\boldsymbol{\beta} = \big(1, -1, 0, 0, 0\big)\boldsymbol{\beta} = \delta_1 - \delta_2.$$

In general, the zeros of a contrast vector are used to exclude the effects of the regression vector which are not of interest and the non–zero elements of it define the comparison of interest.

For the multi–factor factorial experiments, general forms of the contrast matrices are available for the simple effects, main effects, and interaction. As an example, consider a $n_a \times n_b$ experimental layout where n_a and n_b are the number of conditions of the two factors of interest, say, A and B, respectively. The general form of the contrast matrices corresponding to the main effects (C_A, C_B) and interaction (C_{AB}) are:

$$\mathbf{C}'_A = \left[\mathbf{g}_a \vdots (\mathbf{P}_a \otimes \mathbf{1}'_b)\right],\ \mathbf{C}'_B = \left[\mathbf{g}_b \vdots (\mathbf{1}'_a \otimes \mathbf{P}_b)\right],\ \text{ and }\ \mathbf{C}'_{AB} = \left[\mathbf{g}_{a\cdot b} \vdots (\mathbf{P}_a \otimes \mathbf{P}_b)\right],$$

respectively, where $\mathbf{g}_a = [\mathbf{0}_a \vdots \mathbf{0}_a]$, $\mathbf{P}_a = \mathbf{I}_a - (1/a)\mathbf{J}_a$ is the centering matrix, $\mathbf{I}_a$ is the identity matrix, and $\mathbf{J}_a = \mathbf{1}_a\mathbf{1}'_a$ is the sum matrix of order a. In this example, $d = n_a$ for the main effect of A, $d = n_a \cdot n_b$ for the interaction effect, etc.

2.2.2 Estimability

The inclusion of the dye effects in the gene–specific *ANOVA* model (2.5) and the fact that the treatment and dye effects are confounded in a single array (Kerr and Churchill, 2001b), estimability of the effect of interest becomes an issue. The least squares estimate of the regression parameter $\boldsymbol{\beta}$, which is a solution of the consistent system of linear equations $\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{Z}$, is not unique for non–full rank models. However, the estimate of a parametric function $\mathbf{C}'\boldsymbol{\beta}$, say, is unique if it is an estimable function. A linear combination of the parameters $\mathbf{C}'\boldsymbol{\beta}$ is said to be estimable if there exist a linear combination of the response $\mathbf{t}'\mathbf{Z}$, say, which can be used as an unbiased estimate of $\mathbf{C}'\boldsymbol{\beta}$, i.e., $E(\mathbf{t}'\mathbf{Z}) = \mathbf{C}'\boldsymbol{\beta}$. A necessary and sufficient condition for the estimability of the effect $\mathbf{C}'\boldsymbol{\beta}$ is

$$\mathbf{C}'(\mathbf{X}'\mathbf{X})^{-}(\mathbf{X}'\mathbf{X}) = \mathbf{C}', \tag{2.6}$$

where $(\mathbf{X}'\mathbf{X})^{-}$ is a generalized inverse of the moment matrix $\mathbf{X}'\mathbf{X}$ (Searle, 1971, §5.4). The concept of estimability is crucial: if a linear function $\mathbf{C}'\boldsymbol{\beta}$ is not estimable, the associated experimental question can not be answered unbiasedly. That is, any estimate of $\mathbf{C}'\boldsymbol{\beta}$ deviates from the true value by a systematic, unknown quantity. Note that, estimability of an effect does not depend on the response.

The best linear unbiased estimator (BLUE) of an estimable function $\mathbf{C}'\boldsymbol{\beta}$ is

$$\mathbf{C}'\hat{\boldsymbol{\beta}} = \mathbf{C}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Z},$$

which is unique, i.e., does not depend on the choice of the generalized inverse of $\mathbf{X}'\mathbf{X}$ (Searle, 1971, page 181). The variance of the estimator $\mathbf{C}'\hat{\boldsymbol{\beta}}$ is

$$Var(\mathbf{C}'\hat{\boldsymbol{\beta}}) = \sigma^2\mathbf{C}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}, \tag{2.7}$$

where $\mathbf{C}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}$ is called a variance factor if $d = 1$. For $d > 1$, $\mathbf{C}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}$ is a nonnegative definite square matrix of order d which is known as a dispersion matrix. For a given contrast matrix $\mathbf{C}$, by considering the variance factor or dispersion matrix as a function of the design matrix $\mathbf{X}$, the quality of the associated design can be quantified. The role of the variance factor or the dispersion matrix in the test of the respective hypothesis is described in the following section.

2.2.3 Methods of Inference

Though the methods of analyzing microarray data are not the main focus of this dissertation, the inference procedure for testing a hypothesis $H_0 : \mathbf{C}'\boldsymbol{\beta} = \mathbf{0}$ is described in this section for the sake of completeness. Depending on whether $\mathbf{C}$ is a vector or matrix, two test statistics can be considered to test the null hypothesis $H_0 : \mathbf{C}'\boldsymbol{\beta} = \mathbf{0}$.

If $\mathbf{C}$ is a vector, the following test statistic can be used:

$$T_0 = \frac{\mathbf{C}'\hat{\boldsymbol{\beta}}}{\hat{\sigma}\sqrt{\mathbf{C}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}}}.$$

Under the null hypothesis $H_0 : \mathbf{C}'\boldsymbol{\beta} = \mathbf{0}$, the statistic T_0 has a central t–distribution with $f = n - rank(\mathbf{X})$ degrees of freedom, provided the null hypothesis $H_0 : \mathbf{C}'\boldsymbol{\beta} = \mathbf{0}$ is testable. In practice, the technical variance σ^2 is estimated unbiasedly by

$$\hat{\sigma}^2 = \frac{1}{f}\mathbf{Z}'(\mathbf{I}_n - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}')\mathbf{Z}.$$

On the other hand, if $\mathbf{C}$ is a matrix and the null hypothesis $H_0 : \mathbf{C}'\boldsymbol{\beta} = \mathbf{0}$ is testable, the following test statistic can be used:

$$F_0 = \frac{1}{\hat{\sigma}^2 f_1}\mathbf{Z}'\mathbf{T}\mathbf{V}^{-}\mathbf{T}'\mathbf{Z},$$

where $\mathbf{V} = \mathbf{C}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}$, $\mathbf{T} = \mathbf{C}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{Z}'$, and $f_1 = rank(\mathbf{V})$. Under the null hypothesis $H_0 : \mathbf{C}'\boldsymbol{\beta} = \mathbf{0}$, the statistic F_0 has a central F–distribution with f_1 and f degrees of freedom. A large value of the test statistic indicates that the data show evidence against the null hypothesis.

In microarray experiments, the hypothesis under investigation is tested simultaneously for a large number of genes and at the end, a small number of genes is selected as differentially expressed. For making a decision by using a large number of tests, the probability of rejecting a test erroneously must be controlled for all the tests. To control the inferential error which is commonly known as the problem of multiple testing, a number of approaches have been proposed in the context of microarray experiments, e.g., false discovery rate (Benjamini and Hochberg, 1995), significance analysis of microarray (Tusher et al., 2001), etc. Multiple testing procedures can be used to adjust the raw p–values. The resulting adjusted p–values can control the inferential error rate at a specific level for all the tests and are used to select differentially expressed genes. The topic of multiple testing problem is not the focus of this dissertation, see, e.g., Dudoit et al. (2003) for a review.

The test statistic T_0 or F_0 is a function of the data $\mathbf{Z}$, contrast matrix $\mathbf{C}$, and design matrix $\mathbf{X}$. The research question under investigation defines the contrast matrix, but the design matrix depends on the selections of the pair of the treatments that are hybridized to the probes on the arrays. In practice, the experimenter decides which pair of the treatments are hybridized on the arrays, which treatment is labelled with red/green dye, and on the number of times each of the arrays will be replicated. That means, the experimenter can decide on the design matrix before conducting the experiment. Thus, a carefully chosen design matrix (i.e., treatment pairs) could influence more to the inferential procedure than the commonly used ones. In the following sections, a procedure of selecting good designs from a set of candidate designs will be described.

2.3 Efficient Microarray Designs

The criterion by which the quality of a design can be assessed with respect to the estimate of the effect of interest is called an efficiency (optimality) criterion, which we denote by ϕ. Efficiency criteria play a useful role in selecting efficient designs from a set of candidate designs. Efficient designs can provide the estimate of the effect of interest with a smaller variance. If the effect of interest can be expressed in terms of a vector $\mathbf{C}$, the variance factor $\mathbf{C}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}$ itself can be used as an efficiency criterion. If $\mathbf{C}$ is a matrix, however, the efficiency criterion is a function that maps a square matrix into a scalar, i.e., $\phi : \mathbb{R}^{d\times d} \to \mathbb{R}$, where d denotes the number of columns of $\mathbf{C}$. In literature, there exist several efficiency criteria, see, e.g., Pukelsheim (1993, §6).

2.3.1 Efficiency Criteria

The common efficiency criteria can be defined as a function of the eigenvalues of the corresponding dispersion or information matrix. Kerr and Churchill (2001b) considered the A–optimality criterion in the context of microarray experiments which can be defined as the sum of the eigenvalues of the dispersion matrix, i.e., the trace of the dispersion matrix. Another common efficiency criterion is the D–optimality which uses the product of the non–zero eigenvalues of the dispersion matrix, i.e., the determinant of the dispersion matrix. And the E–optimality criterion uses the largest eigenvalue of the dispersion matrix. From the estimation point of view, the A–, D–, and E–optimality criterion deal with the average, generalized, and maximum variances of the estimates, respectively. In this study, we will use only the E–optimality as the efficiency criterion because of its straightforward interpretation. Unlike the A– or D–optimality criterion, the E–optimality criterion does not depend on the dimension of the information or dispersion matrix. A detailed discussion on efficiency criteria and their properties can be found in Pukelsheim (1993, §6). In the following, we will describe a procedure to find efficient microarray designs from a set of candidate designs with respect to the E–optimality criterion.

Let $\mathcal{D} = \{\xi_1, \xi_2, \ldots, \xi_T\}$ be the set of candidate designs and $\mathbf{X}_t$ be the design matrix corresponding to the design ξ_t, $t = 1, 2, \ldots, T$. Let $\mathbf{C}$ be the contrast matrix corresponding to the research question of interest and assume that the effect of interest $\mathbf{C}'\boldsymbol{\beta}$ is estimable for all the candidate designs. Landgrebe et al. (2004) suggested the following expression of the E–optimality criterion corresponding to the design ξ_t,

$$\phi(\xi_t, \mathbf{C}'\boldsymbol{\beta}) = \frac{\operatorname{tr}(\mathbf{C}'\mathbf{C})}{\lambda_{\max}\big(\mathbf{C}'(\mathbf{X}_t'\mathbf{X}_t)^{-}\mathbf{C}\big)}, \tag{2.8}$$

where $\lambda_{\max}(\mathbf{V})$ and $\operatorname{tr}(\mathbf{V})$ denote the largest eigenvalue and trace of the square matrix $\mathbf{V}$, respectively. The numerator of the expression (2.8) is used as a normalizing constant which ensures invariance of the E–optimality criterion $\phi(\cdot,\cdot)$ under scalar multiplication of the contrast matrix, i.e., for a scalar r, $\phi(\xi, \mathbf{C}'\boldsymbol{\beta}) = \phi(\xi, r\mathbf{C}'\boldsymbol{\beta})$, $\forall\, \xi \in \mathcal{D}$. The E–optimality criterion cannot be defined if the effect of interest is non–estimable corresponding to the design under investigation.

The design which corresponds to the largest E–optimality criterion value is the most efficient design (ξ_∞) and it can be formally expressed as

$$\xi_\infty = \arg\max_\xi\{\phi(\xi, \mathbf{C}'\boldsymbol{\beta}), \forall\, \xi \in \mathcal{D}\}. \tag{2.9}$$

The E–optimality criterion is a minimax approach with respect to the dispersion matrix and thus can guard against worst cases. A design ξ_1 is said to be more efficient compared to the design ξ_2 if and only if $\phi(\xi_1, \mathbf{C}'\boldsymbol{\beta}) > \phi(\xi_2, \mathbf{C}'\boldsymbol{\beta})$, provided the effect of interest $\mathbf{C}'\boldsymbol{\beta}$ is estimable with respect to both the designs ξ_1 and ξ_2.

The ratio of the efficiency criterion of two competing designs is commonly known as the relative efficiency which is very useful in interpreting results. If more than two designs are to be compared, the relative efficiency of a design $\xi_t \in \mathcal{D}$ can be defined in terms of the E–optimality criterion as

$$\phi_{\text{rel}}(\xi_t, \mathbf{C}'\boldsymbol{\beta}) = \frac{\phi(\xi_t, \mathbf{C}'\boldsymbol{\beta})}{\max_{\xi\in\mathcal{D}}\,\{\phi(\xi, \mathbf{C}'\boldsymbol{\beta})\}}.$$

For a given set of candidate designs, the relative efficiency of a design gives an idea about its efficiency compared to the other designs of the candidate set.

When more than one experimental question is of interest, the average of the efficiency criterion over different questions is often used as an efficiency criterion (Yang and Speed, 2002; Landgrebe et al., 2004), we call it overall efficiency. However, researchers could be interested in estimating some of the effects more efficiently than others. To accommodate such cases, we suggest to use an weighted average of the efficiency criterion for the computation of an "overall efficiency". Let $\mathbf{C}_q$ be the contrast matrix corresponding to the q^{th} question, $q = 1, 2, \ldots, Q$. The overall efficiency can then be defined in terms of the E–optimality criterion as

$$\bar{\phi}(\xi_t, \mathbf{C}_0'\boldsymbol{\beta}) = \frac{\sum_{q=1}^{Q} w_q\, \phi(\xi_t, \mathbf{C}_q'\boldsymbol{\beta})}{\sum_{q=1}^{Q} w_q},$$

where w_q is the weight corresponding to the q^{th} question, $\mathbf{C}_0' = \big(\mathbf{C}_1 \,\vdots\, \mathbf{C}_2 \,\vdots\, \cdots \,\vdots\, \mathbf{C}_q\big)'$ is the combined contrast matrix, and $\phi(\xi_t, \mathbf{C}_q'\boldsymbol{\beta})$ is the E–optimality criterion corresponding to ξ_t for the effect $\mathbf{C}_q'\boldsymbol{\beta}$. For $q > 1$, the most efficient design can be obtained by using the overall efficiency criterion $\bar{\phi}(\cdot, \cdot)$ in (2.9) instead of the E–optimality criterion $\phi(\cdot, \cdot)$.

Note: In microarray experiments, replications of a basic design are often used to construct designs with a larger number of arrays, i.e., composite designs (e.g., Landgrebe et al., 2004). In this section, we will show the relation between the E–optimality criterion of a basic design and the corresponding composite design. Let $\phi(\xi_1, \mathbf{C}'\boldsymbol{\beta})$ be the E–optimality criterion corresponding to a basic design ξ_1 which has n arrays and $\mathbf{X}_1$ be the corresponding design matrix. Let ξ be the composite design which is composed of two replications of ξ_1, i.e., ξ has $2n$ number of arrays. The design matrix of ξ can be written in terms of the design matrix of ξ_1 as $\mathbf{X} = (\mathbf{X}_1' \vdots \mathbf{X}_1')'$.

By using the relationship

$$(\mathbf{X}'\mathbf{X})^{-} = (\mathbf{X}_1'\mathbf{X}_1 + \mathbf{X}_1'\mathbf{X}_1)^{-} = \frac{1}{2} \cdot (\mathbf{X}_1'\mathbf{X}_1)^{-},$$

we can show,

$$\phi(\xi, \mathbf{C}'\boldsymbol{\beta}) = \frac{\operatorname{tr}(\mathbf{C}'\mathbf{C})}{\lambda_{\max}\big(\mathbf{C}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}\big)} = \frac{2 \cdot \operatorname{tr}(\mathbf{C}'\mathbf{C})}{\lambda_{\max}\big(\mathbf{C}'(\mathbf{X}_1'\mathbf{X}_1)^{-}\mathbf{C}\big)} = 2\,\phi(\xi_1, \mathbf{C}'\boldsymbol{\beta}). \tag{2.10}$$

That means, the E–optimality criterion of a composite design is the product of the number of replications and the E–optimality criterion of the related basic design. This property is also satisfied for the A– and D–optimality criterion.

2.3.2 Non–inferior Designs

Besides the overall efficiency criterion, a filtering procedure can also be used for comparing designs when more than one question is of interest. This filtering procedure classifies the set of candidate designs into inferior and non–inferior designs in such a way that none of the inferior designs can be used to estimate any of the effects of interest more efficiently compared to the non–inferior designs. Formally, a design $\xi^\star$ is said to be a non–inferior design if there exist no design $\xi \in \mathcal{D}$, such that,

$$\phi(\xi^\star, \mathbf{C}_q'\boldsymbol{\beta}) \leq \phi(\xi, \mathbf{C}_q'\boldsymbol{\beta}), \ \ \forall\, q = 1, 2, \ldots, Q,$$

with strict inequality for at least one q. Glonek and Solomon (2004) called the class of non–inferior designs “admissible” and suggested that good microarray designs can be found from the corresponding set of admissible designs. The concept of admissibility is commonly used in statistical decision theory to compare decision rules (Casella and Berger, 1990, §10.4). In the context of microarray experiment, Landgrebe et al. (2004) showed, with an example, that admissible designs are not always the most efficient ones. In §3.1, we will show some examples of inferior and non–inferior designs.

2.4 Robust Microarray Designs

Robustness considerations are necessary in microarray analysis because expression data often contain missing observations due to unreliable spot measurements. Here, we assume that each gene is spotted only once on an array and the expression values are missing completely at random, e.g., missing due to technical reasons. This means that the probability of observing a missing expression measurement is equal across all spots of an array and are constant over different arrays. In this study, the gene–specific *ANOVA* model (2.5) is assumed and for a specific gene, each array contributes only one data point to the analysis. In the following text, the expression "missing an array" is often used to indicate that a data point is missing for the gene of interest.

The major problem with missing values is that they may lead to less efficient or even non–estimable estimates of the effects of interest. As an example, consider two microarray designs for a 1×3 experimental layout, namely, *2CR* and *DS*, each of which has six arrays. Three treatments *1*, *2*, and *3* are to be compared in this experiment. The graphical representations of these two designs are shown in Figure 2.1. The *2CR* designs denotes a design that is consisted of two replications of the basic *CR* design for the 1×3 experimental layout. In *CR* designs, the treatments under investigation are compared indirectly via a common reference sample which we denote by *R*. The *DS* design compares each pair of the treatments twice by reversing the dye label. Here, we follow the common practice of microarray literature for graphically representing an array, i.e., an array is represented by a pair of treatment labels and an arrow. The treatment labels, which can either be numbers or letters, represent *mRNA* samples corresponding to the treatments. The arrow that connects two treatment labels (samples) indicates the dye labelling protocol, e.g., samples at the arrow head and arrow tail are labeled with a red and a green dye, respectively. For example, $1 \rightarrow 3$ represents an array associated with the treatments *1* and *3* that are labeled with a green and a red dye, respectively.

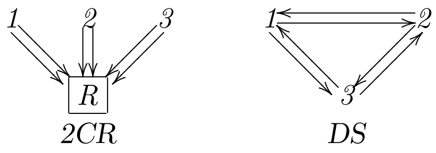


Figure 2.1: Graphical representations of the *2CR* and *DS* designs for 1×3 experimental layout.

In this example, for a specific gene, the following model is assumed for the normalized expression measurement corresponding to the i^{th} array of the *DS* design

$$z_i = \delta_g - \delta_r + \tau_k - \tau_{k'} + \epsilon_i, \begin{cases} i = 1, 2, \ldots, 6 \\ k \neq k' = 1, 2, 3, \end{cases}$$

and in matrix notation, this model can be written as

$$\mathbf{Z} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{\beta} = (\delta_1, \delta_2, \tau_1, \tau_2, \tau_3)'$ (see §2.2 for details of these models). In case of the *2CR* design, the reference sample R is labelled with a red dye for all the arrays and $\boldsymbol{\beta} = (\delta_1, \delta_2, \tau_1, \tau_2, \tau_3, \tau_R)'$. The design matrices corresponding to the designs *2CR* and *DS* are shown in Table 2.1. In the context of microarray experiments, each array contributes one row to the design matrix, e.g., the first row of $\mathbf{X}_{DS}$ corresponds to the array $1 \rightarrow 2$ where the treatments *1* and *2* are labelled with green and red dye, respectively. The first two columns of the design matrices correspond to the dye effects and the remaining columns correspond to the treatment effects. The size of the design matrix of the design *2CR* is larger than that of the design *DS* because the *2CR* design includes a reference sample along with the treatments under investigation.

$$\mathbf{X}_{2CR} = \begin{pmatrix} 1 & -1 & 1 & 0 & 0 & -1 \\ 1 & -1 & 1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 1 & 0 & -1 \\ 1 & -1 & 0 & 1 & 0 & -1 \\ 1 & -1 & 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 & 1 & -1 \end{pmatrix}, \quad \mathbf{X}_{DS} = \begin{pmatrix} 1 & -1 & 1 & -1 & 0 \\ 1 & -1 & -1 & 1 & 0 \\ 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 0 & -1 & 1 \\ 1 & -1 & -1 & 0 & 1 \\ 1 & -1 & 1 & 0 & -1 \end{pmatrix}$$

Table 2.1: The design matrices for the *2CR* and *DS* designs.

In this example, let us assume that the biological question of interest is to compare two treatments *1* and *2*, i.e., $\tau_1 - \tau_2$. The contrast vectors corresponding to the effect $\tau_1 - \tau_2$ are

$$\mathbf{C}_{2CR} = (0, 0, 1, -1, 0, 0)' \quad \text{and} \quad \mathbf{C}_{DS} = (0, 0, 1, -1, 0)' \ ,$$

for the designs *2CR* and *DS*, respectively. Similar to the design matrices, the first two elements of the contrast vectors also correspond to the dye effects and the other columns correspond to the treatment effects. For a general overview of the contrast vectors or matrices in the context of microarray experiment, see §2.2.1.

The effect of interest, $\tau_1 - \tau_2$, is estimable for both the designs *2CR* and *DS* under investigation. Estimability ensures that there exist at least one linear combination of the responses by which an unbiased estimate of the effect of interest can be obtained. Assume, z_i and $z_i^\star$ are the responses corresponding to the i^{th} $(i = 1, 2, \ldots, 6)$ row of the design matrices $\mathbf{X}_{2CR}$ and $\mathbf{X}_{DS}$, respectively. It is easy to show that

$$E(z_1 - z_3 + z_2 - z_4)/2 = \tau_1 - \tau_2 \quad \text{and} \quad E(z_1^\star - z_2^\star - z_3^\star + z_4^\star - z_5^\star + z_6^\star)/4 = \tau_1 - \tau_2$$

for the designs *2CR* and *DS*, respectively. This shows that there exist at least one linear combination of the responses by which the effect of interest $\tau_1 - \tau_2$ can be estimated unbiasedly by both the designs *2CR* and *DS*. Hence, the effect $\tau_1 - \tau_2$ is estimable with respect to both the designs *2CR* and *DS*. In general, an effect is said to be estimable with respect to a design if it satisfies the equation (2.6).

The values of the E–optimality criterion can be obtained for these two designs by using the equation (2.8). The *DS* design is found to be more efficient compared to the *2CR* design for estimating the effect $\tau_1 - \tau_2$ and the corresponding E–optimality criterion values are

$$\phi(DS, \tau_1 - \tau_2) = 6.0 \quad \text{and} \quad \phi(2CR, \tau_1 - \tau_2) = 2.0,$$

respectively. This means that in terms of the relative efficiency one can conclude that the *2CR* design needs to be replicated three times to attain the same efficiency as the *DS* design for estimating $\tau_1 - \tau_2$. Assume now that for the *2CR* design either the $1 \rightarrow R$ or $2 \rightarrow R$ array is missing. In the context of computing the E–optimality criterion, the dimension of the design matrix is reduced in the presence of missing arrays, e.g., if $1 \rightarrow R$ array is missing then the resulting design matrix will be $\mathbf{X}_{2CR}$ without the 1st or 2nd row of it. In this case, the E–optimality criterion value reduces to 1.3. Similarly for the *DS* design, the E–optimality criterion value reduces to 3.6 if any of the arrays connecting the treatments *1* and *2* is missing and reduces to 5.1 if the missing array is associated with the treatment *3*. This shows that a missing array may considerably reduce the efficiency of the estimates and the amount of the reduction depends on the type of the missing array.

In case of two missing arrays, the effect $\tau_1 - \tau_2$ is not estimable for the *2CR* design if both the arrays of the type $1 \rightarrow R$ or $2 \rightarrow R$ are missing. If one array of each type is missing, the E–optimality criterion value reduces to 1.0, i.e., one has only 50 percent of the initial pre-planned efficiency. On the other hand, all the residual designs corresponding to the *DS* design with two missing arrays can be used to estimate the effect $\tau_1 - \tau_2$ unbiasedly (with less efficiency though). The estimate $\hat{\tau}_1 - \hat{\tau}_2$ could be more efficient if none of the arrays that connects treatments *1* and *2* is missing than the cases where one of this type of arrays is missing. If more than three arrays are missing, the effect $\tau_1 - \tau_2$ is not estimable anymore with respect to the *DS* design.

This example shows a number of different situations that may occur if the possibilities of missing arrays are considered in selecting efficient microarray designs for a given experimental layout.

2.4.1 Robustness Criteria

So far no attempt has been made to systematically investigate the robustness of microarray designs. In the following, we propose three different robustness criteria to measure the robustness of a design for the given experimental questions. The proposed robustness criteria will be used in §3.1 for analyzing robustness of different designs. All of the following robustness criteria depend on the possible array constellations with a fixed number of missing arrays.

Let ξ be the design of which the robustness properties will be investigated and $\mathbf{X}_n$ be the associated design matrix of size n. For $m(< n)$ missing arrays, let

$$\mathcal{R}_m(\mathbf{X}_n) = \{\mathbf{X}_{n,1}^{(-m)}, \mathbf{X}_{n,2}^{(-m)}, \ldots, \mathbf{X}_{n,w}^{(-m)}\} \tag{2.11}$$

be the set of design matrices corresponding to the possible $w = \binom{n}{m}$ residual designs which can be constructed from the design ξ by leaving m out of n arrays, i.e., each of the residual designs has $(n-m)$ arrays, where $\mathbf{X}_{n,t}^{(-m)}$ is the design matrix corresponding to the t^{th} residual design, $t = 1, 2, \ldots, w$. Further let

$$\mathcal{R}_m^\star(\mathbf{X}_n, \mathbf{C}'\boldsymbol{\beta}) \subseteq \mathcal{R}_m(\mathbf{X}_n)$$

denote the set of residual designs for which the effect of interest $\mathbf{C}'\boldsymbol{\beta}$ is estimable. Let $w^\star \leq w$ denote the cardinality of $\mathcal{R}_m^\star(\mathbf{X}_n, \mathbf{C}'\boldsymbol{\beta})$.

Breakdown Number

The simplest of the robustness criteria is the breakdown number which represents the minimum number of missing arrays that leads to at least one residual design for which the effect of interest is not estimable. More specifically, the breakdown number, say m_0, of a design states that the effect of interest is estimable with respect to all the residual designs with $(m_0 - 1)$ missing arrays, but there exists at least one residual design with m_0 missing arrays for which the effect is no longer estimable. Formally, the *breakdown* *number* of the design ξ can be defined with respect to the effect $\mathbf{C}'\boldsymbol{\beta}$ as

$$\begin{aligned} BDN(\xi, \mathbf{C}'\boldsymbol{\beta}) &= \min_m \{\mathbf{C}'(\mathbf{X}'\mathbf{X})^-(\mathbf{X}'\mathbf{X}) \neq \mathbf{C}' \text{ for at least one } \mathbf{X} \in \mathcal{R}_m(\mathbf{X}_n)\} \\ &= \min_m \{\mathcal{R}^\star_m(\mathbf{X}_n, \mathbf{C}'\boldsymbol{\beta}) \subset \mathcal{R}_m(\mathbf{X}_n)\}, \end{aligned}$$

where $\mathbf{X}_n$ denote the design matrix of the design ξ. For example, to estimate the effect $\tau_1 - \tau_2$ from the previous example with the 1×3 experimental layout, the breakdown number of the *2CR* design is two, i.e., two missing values may lead to non-estimable comparisons. On the other hand, the breakdown number of the *DS* design is four, i.e., with respect to the breakdown numbers the *DS* design is more robust than the *2CR* design for the 1×3 experimental layout. If more than one question is of interest, the minimum of the corresponding breakdown numbers can be used as a robustness criterion.

Average Efficiency

For the design of interest, the average of an efficiency criterion over the residual designs with a specific number of missing arrays can also be used as a robustness criterion. For a design ξ, the *average efficiency* for estimating $\mathbf{C}'\boldsymbol{\beta}$ with m missing arrays can be defined in terms of the E–optimality criterion as

$$\bar{\phi}_m(\xi, \mathbf{C}'\boldsymbol{\beta}) = \frac{1}{w} \sum_{\mathbf{X} \in \mathcal{R}^\star_m(\mathbf{X}_n, \mathbf{C}'\boldsymbol{\beta})} \frac{\operatorname{tr}(\mathbf{C}'\mathbf{C})}{\lambda_{\max}\big(\mathbf{C}'(\mathbf{X}'\mathbf{X})^-\mathbf{C}\big)},$$

where w is the total number of the residual designs that can be obtained from the design ξ with m missing arrays and $\mathbf{X}_n$ is the design matrix corresponding to the design ξ. If more than one effect is of interest, instead of the E–optimality criterion the overall efficiency can be used in the definition of the average efficiency. For example, to estimate the effect $\tau_1 - \tau_2$ from the previous example with the 1×3 experimental layout, the average efficiency value for the *2CR* design with one missing array is 1.56. This criterion could be useful in selecting designs when the corresponding breakdown numbers are equal. In case of no missing array, the average efficiency criterion deduces to the overall efficiency.

Proportion of the Effective Designs

For a specific number of missing arrays, the proportion of the residual designs for which the effect of interest is estimable can be used to assess the robustness of a design. We call this robustness criterion the *proportion of the effective designs* which can be defined for the design ξ with m missing arrays as,

$$pED_m(\xi, \mathbf{C}'\boldsymbol{\beta}) = \frac{w^\star}{\binom{n}{m}},$$

where $w^\star$ is the cardinality of $\mathcal{R}^\star_m(\mathbf{X}_n, \mathbf{C}'\boldsymbol{\beta})$. For example, to estimate the effect $\tau_1 - \tau_2$ from the previous example with the 1×3 experimental layout, the proportion of the effective designs for the *2CR* design with two missing arrays is 13/15. This criterion is more informative than the breakdown number when the number of missing arrays is greater than or equal to the breakdown number. If more than one effect is of interest, the minimum of the corresponding proportion of the effective designs can be used as a robustness criterion.

2.5 Simulation Study

In this section, a simulation study is considered to demonstrate a few benefits of using an efficient design over the inefficient ones in the context of microarray experiments. This is one way to show the amount of losses one could experience due to the selection of an inefficient design. Assume a 1×4 experimental layout where four treatments, namely, *1*, *2*, *3*, and *4* are to be compared. As an example, the comparison of the treatments *1* and *2* (i.e., $\tau_1 - \tau_2$) is considered as the effect of interest in this simulation study. Three designs *3CR*, *3CL*, and *3XL* are used for the comparison and each of these designs has 12 arrays. The selected designs are composed of three replications of the *CR*, *CL*, and *XL* designs. Figure 2.2 shows the graphical representations of the *CR*, *CL*, and *XL* designs for the 1×4 experimental layout.

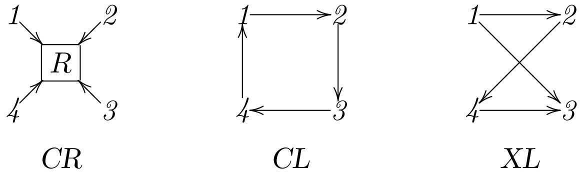


Figure 2.2: Graphical representation of the *CR*, *CL*, and *XL* designs for 1×4 experimental layout.

Table 2.2 shows the values of the *E*–optimality criterion corresponding to the effect $\tau_1 - \tau_2$ for the designs that are considered for this simulation study. Among these designs, the *3CL* design is the most efficient for estimating the effect $\tau_1 - \tau_2$ and the corresponding *E*–optimality criterion value is 8.00. For the *3XL* and *3CR* design, the

Design	Number of arrays, n	E–optimality, ϕ
3CR	12	3.00
3CL	12	8.00
3XL	12	6.00

Table 2.2: The values of the E–optimality criterion corresponding to the designs *3CR*, *3CL*, and *3XL* with respect to the effect $\tau_1 - \tau_2$.

values of the E–optimality criterion are 6.00 and 3.00, respectively.

Assume that 10,000 known genes are spotted on each array and only 50 of these genes are known to be differentially expressed for comparing the treatments *1* and *2*. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ denote two mutually exclusive sets of genes where the former contains the 9950 genes which are not differentially expressed with respect to the effect $\tau_1 - \tau_2$ and the latter contains the remaining 50 genes. To simulate raw expression measurement for each gene, a measurement error model is used. A brief description of this model is given in the following section.

2.5.1 Rocke–Durbin's Measurement Error Model

Rocke and Durbin (2001) proposed the following model for microarray expression measurement Y_{jg} corresponding to the g^{th} gene in the j^{th} channel (dye),

$$Y_{jg} = a_j + b_j X_{jg}\, e^{\nu_g + \zeta_{jg}} + \omega_g + \psi_{jg}, \tag{2.12}$$

where a_j is the background signal corresponding to the j^{th} dye, b_j is the dye–specific slope, X_{jg} is the *mRNA* concentration of the g^{th} gene in the sample that is labelled with j^{th} dye, ν_g and ω_g are the gene–specific error term, and ζ_{jg} and ψ_{jg} are the error terms corresponding to both gene and dye. The error terms of this model have both multiplicative $(\nu_g + \zeta_{jg})$ and additive $(\omega_g + \psi_{jg})$ components. The multiplicative errors are related to labelling, scanning, and spot features, whereas the additive errors are related to local background. The distributions of the error components ν_g, ζ_{jg}, ω_g, and ψ_{jg} are assumed to be $N(0, \sigma^2_\nu)$, $N(0, \sigma^2_{\zeta_j})$, $N(0, \sigma^2_\omega)$, and $N(0, \sigma^2_{\psi_j})$, respectively.

Cui et al. (2003) used the model (2.12) for comparing different data transformation techniques (e.g., logarithm transformation, shift transformations, curve fitting transformations, variance stabilizing transformations, etc.) that are commonly used in microarray data analysis. They also simulated some common features of microarray data by varying different parameters of the model (2.12). For example, excess variation at the low expression end can be simulated by using a large channel–specific additive errors

ψ_{jg}. Large value of the channel–specific multiplicative error ζ_{jg} increases the variation at the high end of the expression measurements (see Cui et al., 2003, for details).

2.5.2 Simulation of Microarray Expression Data

In this section, the Rocke–Durbin's measurement error model (2.12) is used to simulate raw expression measurements for the two channels at each spot on an array. Besides the background measurements, slopes, and variances of different error components, the distribution of the true *mRNA* concentration X_{jg} in the competing samples require to be specified. It is assumed that X_{jg} has a lognormal distribution with mean μ_{jg} and variance σ^2_{jg} (Hoyle et al., 2002).

Table 2.3 shows the parameter values of Rocke–Durbin's measurement error model (2.12) that are used in this simulation study. Assume that the background signals, slopes, and error variances are equal for both the channels $Cy3$ and $Cy5$. The mean of the true *mRNA* concentration of the genes $g \in \mathcal{C}_2$ in the treatment *1* is higher by the amount $\Delta \geq 0$ compared to the *mRNA* concentration of the other genes. To see the effect of Δ in microarray data analysis, different values of Δ ranging from 0.5 to 2.0 are considered. For each of the designs that are considered in this simulation study, the gene expression measurements corresponding to two channels $Cy3$ and $Cy5$ are independently generated for each of its arrays according to the assumed model (2.12) with the parameter values defined in Table 2.3.

Parameters	$Cy3$	$Cy5$
a	0.00	0.00
b	0.50	0.50
ν	1.10	1.10
ζ	0.10	0.10
ω	0.10	0.10
ψ	1.00	1.00

Parameters	Treatments	$\mathcal{C}_1$	$\mathcal{C}_2$
μ	*1*	8.00	8.00+Δ
	2–4	8.00	8.00
σ^2	*1–4*	0.80	0.80

Table 2.3: Selected parameter values of the Rocke–Durbin's measurement error model that are used to simulate microarray expression data.

2.5.3 Analysis of the Simulated Data

For each design, the raw expression measurements are transformed by logarithms and then the log–transformed expression measurements are normalized by a lowess regression model (Yang et al., 2002b). For each gene, the normalized ratio of the expression measurements corresponding to the i^{th} $(i = 1, 2, \ldots, 12)$ array of the *3CL* or *3XL* design can be written as

$$z_i = \delta_1 - \delta_2 + \tau_k - \tau_{k'} + \epsilon_i, \;\; k \neq k' = 1, 2, 3, 4,$$

where δ_j and τ_k are the dye and treatment effects, respectively. In matrix notation, this model can be written as

$$\mathbf{Z} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{Z} = (z_1, z_2, \ldots, z_{12})'$ is the vector of responses, $\mathbf{X}$ is the design matrix, and $\boldsymbol{\beta} = (\delta_g, \delta_r, \tau_1, \ldots, \tau_4)'$ is the vector of the parameters. In case of the *3CR* design, the reference sample R is labelled with a red dye for all the arrays and the corresponding regression parameter $\boldsymbol{\beta} = (\delta_g, \delta_r, \tau_1, \ldots, \tau_4, \tau_R)'$. A detailed discussion of this model is given in §2.2.

The null hypothesis of interest $H_0 : \tau_1 - \tau_2 = 0$ can be written in terms of a suitable contrast matrix as $H_0 : \mathbf{C}'\boldsymbol{\beta} = 0$, where

$$\mathbf{C} = \begin{cases} (0, 0, 1, -1, 0, 0, 0)' & \text{for the design } \textit{3CR} \\ (0, 0, 1, -1, 0, 0)' & \text{for the designs } \textit{3CL} \text{ of } \textit{3XL} \end{cases}$$

Using the test statistic T_0 (see §2.2.3), the raw p–values are are calculated for each gene. The false discovery rate procedure (Benjamini and Hochberg, 1995) is used to compute the corresponding adjusted p–values. A brief description of the methods of inference is given in §2.2.3.

In this simulation study, the performances of the designs are compared on the basis of the corresponding estimates of the true positives and false positives. True positives are those genes which belong to the class $\mathcal{C}_2$ and are detected as differentially expressed. On the other hand, the false positives are the genes which belong to the class $\mathcal{C}_1$ and are detected as differentially expressed.

True Positives

In this context, the probability of the true positives indicate the power of a design in correctly detecting differentially expressed genes $g \in \mathcal{C}_2$. The probabilities of the true positives are computed from 100 simulations for each value of Δ. The estimate of the probability of true positives at a specific Δ (with a fixed level of significance α_0) can be expressed for our simulated data as

$$\widehat{T}_p = \frac{1}{100} \sum_{s=1}^{100} \frac{1}{50} \sum_{g \in \mathcal{C}_2} I(p_g < \alpha_0),$$

where p_g is the adjusted p–value corresponding to the g^{th} gene, $I(\cdot)$ is the indicator function. As expected, Figure 2.3 shows that the probability of true positives increases as Δ increases for all the competing designs. However, at a fixed value of $0.5 < \Delta < 2.0$ the *3CR* design is out performed by the other two designs, e.g., at $\Delta = 1.5$, the *3CR* design can detect only 40 percent of the differentially expressed genes whereas, the other two designs can detect more than 80 percent of the genes.

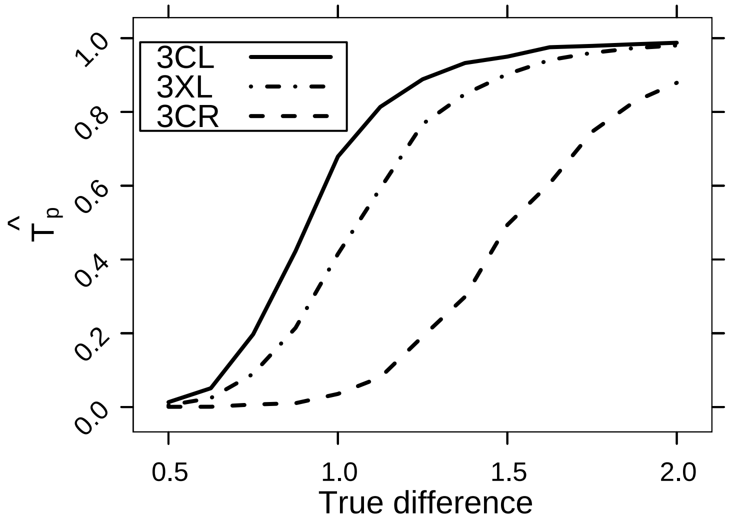


Figure 2.3: Distributions of the estimates of the probability of the true positives over the true difference in the gene expression levels for the designs *3CR*, *3CL*, and *3XL*.

False Positives

Figure 2.4 shows the receiver operating characteristic (ROC) curves corresponding to the competing designs. The ROC curve is widely used in diagnostic tests for evaluating the performance of a new procedure relative to the gold standard (Zhou et al., 2002). It plots the probability of the true positives (sensitivity) against the probability of the false positives (1–specificity). In the context of microarrays, the probability of the false positives can be considered as an estimate of the false discovery rate which is often used microarray data analysis for controlling the significance level of a test. In this case, the raw expression measurements are simulated with $\Delta = 1.50$. The probabilities of the false positive are calculated at different values of the level of significance by using 100 simulations. The estimate of the probability of the false positives at a specific value of the level of significance α_0 can be expressed for our simulation data as

$$\widehat{F}_p = \frac{1}{100}\sum_{s=1}^{100}\frac{1}{9950}\sum_{g\in\mathcal{C}_1} I(p_g < \alpha_0).$$

This analysis also shows that the *3CL* design performs better than the *3CR* or *3XL* design. Allowing five percent of false positives, the *3CL* design can detect almost all the differentially expressed genes, but the *CR* design can detect only 60% of those.

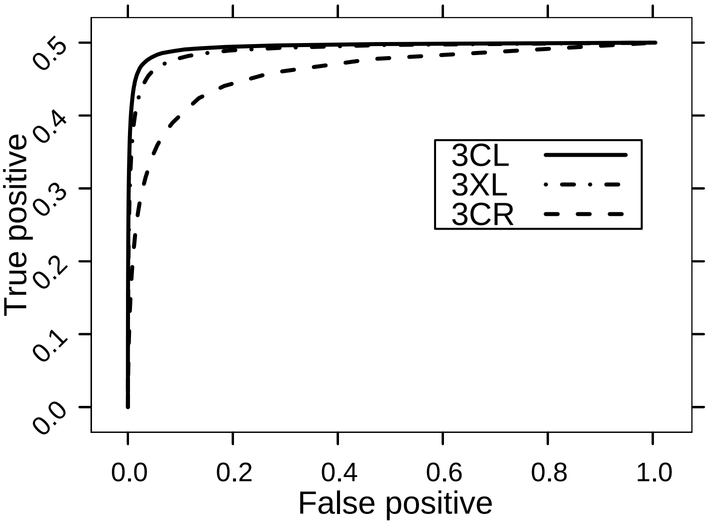


Figure 2.4: ROC curve for comparing the designs *3CR*, *3CL*, and *3XL* for 1 × 4 experimental layout.

2.6 Conclusion

In this chapter, the global *ANOVA* model (Kerr and Churchill, 2001b) which is commonly used for analyzing microarray expression data is briefly reviewed. The connection between the global *ANOVA* model and the gene–specific *ANOVA* model (Landgrebe et al., 2004) is shown analytically. The concept of estimability and methods of inference are also reviewed in the context of microarray experiments.

For a given experimental question, a procedure of selecting good designs from a set of candidate designs has been described in this chapter. The procedure is defined by using the E–optimality as an efficiency criterion, but other efficiency criteria such as, A– or D–optimality can also be used after suitable adjustment for the dimensions. When more than one question is of interest, the importance of the individual question can be specified in the procedure of selecting good designs.

In microarray experiments, missing observations are often observed due to unreliable spot measurements. To assess the performance of a microarray design in estimating the effects in the presence of possible missing observations, three robustness criteria, namely, the breakdown number, average efficiency, and proportion of the effective designs are proposed in this chapter. Robustness criteria can also be incorporated in the selection procedures along with the efficiency criteria. For a specific design, robustness criteria are defined on the basis of its possible residual designs with a specific number of missing arrays.

At the end, a simulation study is considered to show the benefits of using efficient designs in microarray experiments. This study showed that for a 1×4 experimental layout, the inefficient common reference design could miss about 40 percent of the differentially expressed genes, whereas, an efficient loop design can detect almost all the differentially expressed genes.

Chapter 3

Examples of Efficient and Robust Microarray Designs

3.1 Introduction

In this chapter, we analyze the efficiency and robustness of some important microarray designs for the one–way and two–factor factorial experiments. The basic designs for each of the experimental layouts are defined first and the replications or combinations of the basic designs are used to construct the composite designs. We do not consider the combinations of the common reference designs with other types of the basic designs because one of our objectives is to compare the common reference design with the other competing designs in terms of the efficiency and robustness criteria. When more than one effect is of interest, we restrict our search only to the designs for which all the effects are estimable.

An **R**(R Development Core Team, 2004) package *robustMAdesigns* was written that can be used to compute the different efficiency and robustness criteria of a design. The description of the different functions of the package is given in Appendix *A*. The package will be available on request or can be downloaded from the web site of the Department of Medical Statistics, University of Göttingen (`www.ams.med.uni-goettingen.de`).

We use the notation *nDesign* for specifying a design where n denotes the number of replications of the design *Design*. For example, *2CL* denotes the design which is composed of two replications of the *CL* design. We further denote a design *Design* with reverse dye labelling by $Design_r$, e.g., the CL_r design uses the reverse dye labelling compared to the *CL* design. If the arrays of the two different basic designs are combined to construct a composite design then the resulting design is named after the two basic designs, e.g., the design *CL*/*CL*$_r$ is obtained by combining the arrays of the designs *CL* and *CL*$_r$. The naming protocol is graphically explained in Figure 3.1.

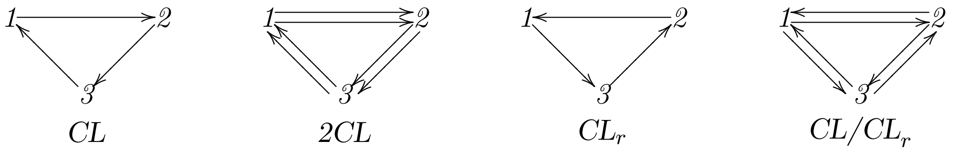


Figure 3.1: Examples of the designs for 1 × 3 experimental layout to demonstrate the naming protocol that is used in this dissertation for microarray designs.

3.2 One–Way Factorial Experiments

The one–way factorial experiment is the simplest experimental design where the levels of the factor are to be compared. Throughout this dissertation, we use the term treatment to define a level of the factor under investigation. The one–way factorial experiment is usually denoted by $1 \times K$, where K is the total number of the treatments under investigation.

In the following sections, designs for 1×3 and 1×4 experimental layouts are considered as examples to demonstrate the procedure of selecting good microarray designs from a set of candidate designs.

3.2.1 Microarray Designs for 1×3 Experimental Layout

The basic microarray designs for a 1×3 experimental layout are *CR* and *CL* designs; each of which consists of three arrays. For this layout, the *DS* design has six arrays and can be obtained by combining the arrays of the two loop designs *CL* and *CL*$_r$, where *CL*$_r$ denotes the design which uses the reverse dye labelling compared to the design *CL* (see Figure 3.2 for graphical representations of these two loop designs), i.e., for the 1×3 experimental layout, the *DS* design can also be denoted by CL/CL_r. In this section, different microarray designs for the 1×3 experimental layout with six, nine, and 12 arrays are considered for comparing their performances in estimating the effect of interest $\tau_1 - \tau_2$. The effect $\tau_1 - \tau_2$ is chosen without loss of generality because, for this experimental layout, each of the pairwise comparisons ($\tau_1 - \tau_2$, $\tau_2 - \tau_3$, or $\tau_1 - \tau_3$) can be estimated with equal efficiency by using the *CL* and *CR* designs. The respective values of the E–optimality criterion and the robustness criteria corresponding to the effect $\tau_1 - \tau_2$ are shown in Table 3.1. For each of the competing designs, the number of

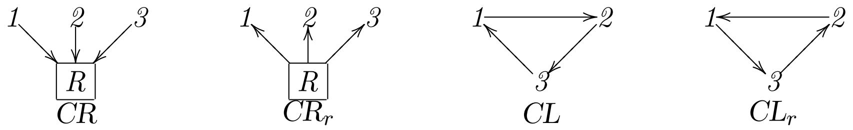


Figure 3.2: Graphical representations of the basic microarray designs for 1 × 3 experimental layout. Each of the designs has three arrays. The difference between the designs *CR* and *CR*$_r$, or *CL* and *CL*$_r$ lies in the dye labelling protocol. For example, if $Cy5$ is used to label the reference sample R for the design *CR* then $Cy3$ will be used to label the reference sample for the design *CR*$_r$.

			Number of missing arrays, m						
			0	1		2		3	
Design	n	*BDN*	$\bar{\phi}_0$	$\bar{\phi}_1$	pED_1	$\bar{\phi}_2$	pED_2	$\bar{\phi}_3$	pED_3
2CR	6	2	2.00	1.56	1.00	1.11	0.87	0.67	0.60
CR/CR$_r$	6	2	2.00	1.43	1.00	0.87	0.87	0.40	0.40
2CL	6	2	6.00	4.63	1.00	2.94	0.80	1.20	0.40
CL/CL$_r$	6	4	6.00	4.63	1.00	3.29	1.00	1.80	1.00
3CR	9	3	3.00	2.60	1.00	2.20	1.00	1.80	0.98
3L	9	3	9.00	7.79	1.00	6.55	1.00	5.24	0.96
2CL/CL$_r$	9	5	9.00	7.79	1.00	6.57	1.00	5.35	1.00
4CR	12	4	4.00	3.62	1.00	3.24	1.00	2.86	1.00
4CL	12	4	12.00	10.85	1.00	9.70	1.00	8.53	1.00
3CL/CL$_r$	12	6	12.00	10.85	1.00	9.70	1.00	8.55	1.00
2CL/2CL$_r$	12	8	12.00	10.85	1.00	9.70	1.00	8.55	1.00

Table 3.1: The values of the efficiency and robustness criteria for the selected designs for 1×3 experimental layout to estimate the effect $\tau_1 - \tau_2$ with different number of missing arrays.

arrays n, breakdown numbers *BDN*, average efficiencies with m missing arrays $\bar{\phi}_m$, and proportion of the effective designs with m missing arrays pED_m are shown in Table 3.1.

For constructing designs with six arrays, two replications of the basic designs are used with/without reverse dye labelling, i.e., the designs, namely, *2CR*, *CR/CR*$_r$, *2CL*, and *CL/CL*$_r$ are considered in this case. These designs are selected in such a way that

(i) the loop and common reference designs can be compared and

(ii) the effect of the reverse dye labelling in the common reference and loop designs can be assessed.

Table 3.1 shows that the reverse dye labelling does not affect the efficiency of the estimate of the effect $\tau_1 - \tau_2$, provided there is no missing observation in the data (i.e., $m = 0$). For example, the designs *2CL* and *CL/CL*$_r$ are found to be equally efficient for estimating the effect $\tau_1 - \tau_2$. In general, the loop designs are found to be more efficient than the common reference designs. In terms of the relative efficiency, one can conclude that it would take two replications of the *2CR* or *CR/CR*$_r$ design to attain the same efficiency of the loop designs (*2CL* or *CL/CL*$_r$) for estimating the effect $\tau_1 - \tau_2$.

As it is already mentioned in §2.4.1, the performance of a design for estimating the effects of interest in the presence of missing observations can be assessed from the performances of the corresponding residual designs for estimating the same effects. For a design with six arrays (e.g., *2CL*, *2CR*, etc.), at most 6 $\left(= \binom{6}{1}\right)$ residual designs can be considered with one missing array. The proportion of the effective designs with one missing array *pED*$_1$ show that all the residual designs (corresponding to both the loop and common reference designs) can be used to estimate the effect $\tau_1 - \tau_2$ unbiasedly. The average efficiency values with one missing array ($\bar{\phi}_1$) show that on an average, the loop designs would be more efficient than the common reference designs.

In case of two missing arrays, 15 $\left(= \binom{6}{2}\right)$ residual designs can be considered for each of the competing designs with six arrays. The values of the proportion of the effective designs with two missing arrays *pED*$_2$ reveal that the *2CL* design is less robust compared to the *2CR* or *2CR*$_r$ design because 87 percent of the residual designs corresponding to the latter can estimate the effect $\tau_1 - \tau_2$ unbiasedly, where as only 80 percent of the residual designs corresponding to the *2CL* design can do so. However, the *CL/CL*$_r$ design, a loop design that uses the reverse dye labelling, is found to be the most robust because the effect $\tau_1 - \tau_2$ can be estimated unbiasedly by using all the corresponding residual designs, i.e, for the *CL/CL*$_r$ design *pED*$_2$ $= 1.00$. This shows that the reverse dye labelling can improve the robustness of the loop designs, but not of the common reference designs. On an average, the *CL/CL*$_r$ design is found to be the most efficient one in the presence of two missing observations.

In §2.4.1, the breakdown number is defined as a robustness criterion which indicates the minimum number of the missing observations that could lead to at least one of its corresponding residual designs for which the effect is not estimable. The *CL/CL*$_r$ design is found to be more robust compared to the competing loop and common reference designs because the breakdown number of the *CL/CL*$_r$ design is four, whereas the breakdown number of the other competing designs is two. This means that if the *CL/CL*$_r$ design is used in a microarray experiment then the effect of interest $\tau_1 - \tau_2$ can be estimated unbiasedly in the presence of at most three missing observations.

Table 3.1 also shows the comparisons of the loop and common reference designs with nine and 12 arrays in terms of their robustness and efficiency criteria. The microarray designs with nine and 12 arrays are constructed by combining three and four replications of the basic designs, respectively. For the designs with nine arrays, the *2CL/CL$_r$* design is found to be the most robust design (breakdown number is five) and is more efficient compared to the *3CR* design.

For the designs with 12 arrays, the *2CL/2CL$_r$* design is found to be the most robust design and the corresponding breakdown number is eight. As before, the *4CR* design is found to be less efficient compared to the loop designs (*4CL*, *2CL/2CL$_r$*, *3CL/CL$_r$*) irrespective of the number of missing arrays. The *2CL/2CL$_r$* design is found to be more robust (breakdown number is eight) compared to the *3CL/CL$_r$* design (breakdown number is six). This is because the *2CL/2CL$_r$* design is more balanced with respect to the dye bias compared to the *3CL/CL$_r$* design.

3.2.2 Microarray Designs for 1×4 Experimental Layout

The basic *CR* and *CL* designs for a 1×4 experimental layout consist of four arrays each. The *DS* design for this experimental layout has 12 arrays and unlike the 1×3 experimental layout, it cannot be obtained by combining the replications of the basic loop designs. In this section, designs with eight, nine, and 12 arrays for the 1×4 experimental layout are compared in estimating different pairwise treatment comparisons. The microarray designs for the 1×4 experimental layout with eight and 12 arrays are obtained by combining the replications of the basic *CR* and *CL* designs. For the designs with nine arrays, the Bechhofer–Tamhane (*B–T*) design (Bechhofer and Tamhane, 1981) is considered. The graphical representations of the designs that are considered in this section are displayed in Figure 3.3. The values of the *E*–optimality criterion and robustness criteria corresponding to the effect $\tau_1 - \tau_2$ are reported in Table 3.2 for the competing designs.

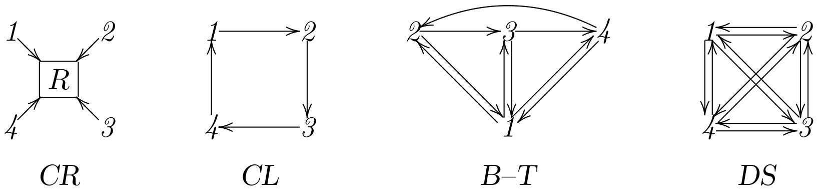


Figure 3.3: Graphical representations of the selected microarray designs for 1×4 experimental layout with four, nine, and 12 arrays.

First, the designs are compared with respect to the effect $\tau_1 - \tau_2$. Among the designs with eight arrays, the loop designs (*2CL* and *CL/CL$_r$*) are found to be more efficient than the common reference (*CR/CR$_r$* and *2CR*) designs, provided there is no missing

			Number of missing arrays, m						
			0	1		2		3	
Design	n	*BDN*	$\overline{\phi}_0$	$\overline{\phi}_1$	pED_1	$\overline{\phi}_2$	pED_2	$\overline{\phi}_3$	pED_3
2CR	8	2	2.00	1.67	1.00	1.33	0.93	1.00	0.79
CR/CR$_r$	8	2	2.00	1.60	1.00	1.20	0.93	0.81	0.79
2CL	8	2	5.33	4.45	1.00	3.21	0.86	1.80	0.57
CL/CL$_r$	8	4	5.33	4.45	1.00	3.58	1.00	2.67	1.00
$B-T$	9	4	6.67	5.69	1.00	4.71	1.00	3.73	1.00
3CR	12	3	3.00	2.70	1.00	2.40	1.00	2.10	0.99
3CL	12	3	8.00	7.22	1.00	6.41	1.00	5.41	0.98
2CL/CL$_r$	12	6	8.00	7.22	1.00	6.42	1.00	5.63	1.00
DS	12	6	8.00	7.18	1.00	6.36	1.00	5.54	1.00

Table 3.2: The values of the efficiency and robustness criteria for the selected designs for 1×4 experimental layout to estimate $\tau_1 - \tau_2$ with different number of missing arrays.

observation in the data. As before, for a given design the corresponding residual designs are used to estimate the robustness criteria of the design. Table 3.2 shows that the breakdown number of the *CL/CL*$_r$ design is four, whereas the breakdown number of the other competing designs is two. This shows that the *CL/CL*$_r$ design is more robust compared to the other comparable designs that are considered in this section. Similar to the 1 × 3 experimental layout, it is also observed for the 1 × 4 experimental layout that the reverse dye labelling can improve the robustness only of the loop designs and on an average, the loop designs are found to be more efficient than the common reference designs.

For the designs with 12 arrays, the *DS*, *3CL*, and *2CL/CL*$_r$ designs are equally efficient for estimating the effect $\tau_1 - \tau_2$ and are more efficient than the *3CR* design in the presence of no missing observation. Table 3.2 shows that the *DS* and *2CL/CL*$_r$ designs are found to be more robust than the *3CL* or *3CR* design in terms of the breakdown numbers. The *B–T* design, which has nine arrays, outperforms all the designs with eight arrays in terms of both the breakdown numbers and average efficiencies. Moreover, this design is more robust than the *3CR* and *3CL* designs and more efficient than the *3CR* design.

For a 1 × 4 experimental layout, at most six pairwise treatment comparisons can be considered. The *CR* design can be used to estimate each of the pairwise treatment comparisons with equal efficiency because for this design, each of the treatments is connected to the reference sample by only a single array, e.g., *1* → *R*, *2* → *R*, etc. On the other hand, for the loop designs the number of arrays required to connect a pair of treatments depends on the choice of the pair, e.g., for the design *CL* of Figure 3.3, treatments *1* and *2* are connected by a single array 1 → 2, but at least two arrays (e.g., *1* → *2* and *2* → *3*) are required to connect the treatments *1* and *3*. The loop design losses efficiency if the number of arrays that are needed to connect the pair of treatments increases. For example, the *3CR* design can be used to estimate the effects $\tau_1 - \tau_2$ and $\tau_1 - \tau_3$ unbiasedly with the same *E*–optimality value 3.00, but the corresponding *E*–optimality values for the *3CL* design are 8.00 and 6.00, respectively. This means that the *3CL* design is less efficient for estimating the effect $\tau_1 - \tau_3$ compared to the effect $\tau_1 - \tau_2$ because it needs two arrays to connect the treatments *1* and *3*, but needs only one array for connecting the treatments *1* and *2*. However, Table 3.3 shows that the *3CL* design is more efficient compared to the *3CR* design for estimating the effect $\tau_1 - \tau_3$. For each of the loop designs with 12 arrays, the breakdown numbers are found to be the same with respect to the effects $\tau_1 - \tau_2$ and $\tau_1 - \tau_3$.

			Number of missing arrays, m						
			0	1		2		3	
Design	n	*BDN*	$\overline{\phi}_0$	$\overline{\phi}_1$	*pED*$_1$	$\overline{\phi}_2$	*pED*$_2$	$\overline{\phi}_3$	*pED*$_3$
3CR	12	3	3.00	2.70	1.00	2.40	1.00	2.10	0.99
3CL	12	3	6.00	5.33	1.00	4.67	1.00	3.96	0.98
2CL/CL$_r$	12	6	6.00	5.33	1.00	4.67	1.00	4.00	1.00
DS	12	6	8.00	7.18	1.00	6.36	1.00	5.54	1.00

Table 3.3: The values of the robustness and efficiency criteria for the selected designs for 1 × 4 experimental layout to estimate $\tau_1 - \tau_3$ with different numbers of missing arrays.

In microarray analysis, often the researchers are interested in more than one experimental question in a single experiment. Suppose the interest is in estimating all the pairwise treatment comparisons. As it is mentioned in §2.3.1, the overall efficiency can be used as an efficiency criterion when more than one question is of interest. Figure 3.4 displays the distribution of the average efficiencies over the number of missing arrays with respect to all the pairwise treatment comparisons for the competing designs with 12 arrays. It shows that the loop designs (*3CL* and *2CL/CL$_r$*) are more efficient than the *3CR* design. For a moderate number of missing observations ($m < 5$), the *B–T* design with nine arrays is found to be more efficient than the *3CR* design, which has 12 arrays. In this case, the *DS* design is found to be more efficient than the other competing designs.

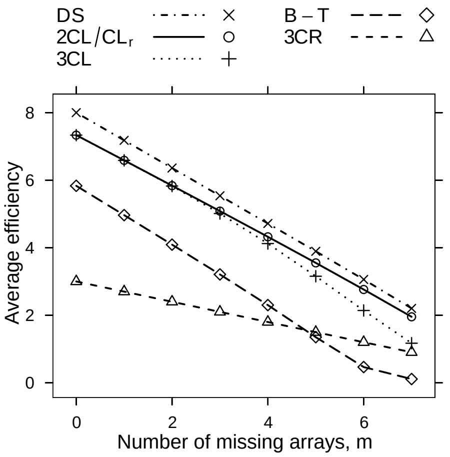


Figure 3.4: Distributions of the average efficiency over the number of missing arrays for the selected designs for 1×4 experimental layout. All the pairwise treatment comparisons are considered as the effects of interest.

3.3 Multi–Factor Factorial Experiments

In the following sections, designs for 2×2 and 3×2 experimental layouts are considered to demonstrate the use of efficiency and robustness criteria in selecting good microarray designs from a set of candidate designs. The main effects and the interaction are assumed to be effects of interest.

3.3.1 Microarray Designs for 2×2 Experimental Layout

In a 2×2 factorial experiment, each of the two factors (say A and B) has two levels. Landgrebe et al. (2004) discussed some basic types of microarray designs for the 2×2 experimental layout, namely, common reference (*CR*), circular loop (*CL*), cross loop (*XL*), cross-swap (*XS*), A–swap (*AS*), and B–swap (*BS*); each of these designs has four arrays. Figure 3.5 shows the graphical representations of these basic designs where a pair of numbers is used to specify a treatment combination. The number at the first position indicates the treatment level of the factor A and that at the second position corresponds to the factor B, e.g., *12* represents the treatment combination corresponding to the first level of the factor A and the second level of the factor B. In this section, the designs with four and eight arrays are compared in terms of the robustness and efficiency criteria. The objective of this comparison is to select the best designs from the set of candidate designs when the effects of interest are main effects A and B, and interaction $A \times B$.

The associated values of the efficiency and robustness criteria for the basic designs

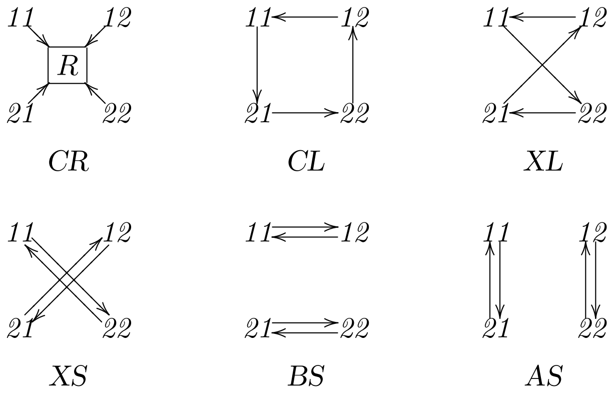


Figure 3.5: Graphical representations of the basic microarray designs for 2 × 2 experimental layout, each of which has four arrays.

are illustrated in Table 3.4. Among the six basic designs, the *CR*, *CL*, and *XL* designs can only be used to estimate all three effects. The *CL* and *XL* designs are found to be more efficient than the *CR* design for estimating any of the three effects of interest. The *CR* design is also less robust compared to the *CL* or *XL* design in the sense that none of its residual designs with one missing array can be used to estimate any of the effects of interest, i.e., $pED_1 = 0.00$ for all the effects A, B, $A \times B$. However, a half of the residual designs with one missing array corresponding to the *CL* or *XL* design can be used to estimate two out of the three effects, e.g., for the design *CL*, $pED_1 = 0.50$ corresponding to the effects A and B.

	E–optimality				pED_1		
Design	A	B	$A \times B$	Overall	A	B	$A \times B$
CR	1.00	1.00	1.00	1.00	0.00	0.00	0.00
CL	2.00	2.00	4.00	2.67	0.50	0.50	0.00
XL	2.00	4.00	2.00	2.67	0.50	0.00	0.50
XS	4.00	4.00	NA	NA	1.00	1.00	NA
AS	4.00	NA	4.00	NA	1.00	NA	1.00
BS	NA	4.00	4.00	NA	NA	1.00	1.00

Table 3.4: The values of the E–optimality criterion and proportion of the effective designs with one missing array for the basic microarray designs for 2 × 2 experimental layout.

If all three effects are of equal interest, the designs *CL* and *XL* are found to be equally efficient in terms of the overall efficiency. But, the design *CL* is preferable to the design *XL* because the interaction, which is usually the most important effect in a multi–factor factorial experiment, can be estimated more efficiently by the former design than the latter.

The designs with eight arrays for the 2 × 2 experimental layout are constructed by using the combinations/replications of the corresponding basic designs. In Table 3.5, the E–optimality values and breakdown numbers of the designs that can be used to estimate all three effects of interest are reported.

	E–optimality				*BDN*			
Design	A	B	$A \times B$	Overall	A	B	$A \times B$	min
2CR	2.00	2.00	2.00	2.00	2	2	2	2
XL/BS	4.00	8.00	6.00	6.00	2	3	4	2
CL/XL_r	4.00	6.00	6.00	5.33	4	4	4	4
CL/XS	6.00	6.00	4.00	5.33	4	4	4	4
CL/BS	2.00	6.00	8.00	5.33	2	4	3	2
XL/XL_r	4.00	8.00	4.00	5.33	4	4	4	4
XL/XS	6.00	8.00	2.00	5.33	4	3	2	2
CL/CL_r	4.00	4.00	8.00	5.33	4	4	4	4
2CL	4.00	4.00	8.00	5.33	2	2	2	2

Table 3.5: The values of the E–optimality criterion and breakdown number for some selected composite designs for 2 × 2 experimental layout.

- If all three effects are of equal interest, the *2CR* design is found to be less efficient compared to the loop designs that are reported in Table 3.5. In terms of the overall efficiency, the *XL/BS* design is found to be the most efficient one. However, the *XL/BS* design is less robust compared to the CL/CL_r, CL/XL_r, *CL/XS*, and XL/XL_r designs in terms of the minimum of the breakdown numbers.

- Table 3.6 shows a comparison between the CL/CL_r, CL/XL_r, *CL/XS*, and XL/XL_r designs on the basis of the average efficiency. It reveals that the CL/CL_r or XL/XL_r design is preferable to the designs CL/XL_r or *CL/XS* if the number of

	Number of missing arrays, m			
Design	0	1	2	3
XL/XL_r	5.333	4.355	3.378	2.389
CL/CL_r	5.333	4.355	3.378	2.389
CL/XS	5.333	4.361	3.362	2.345
CL/XL_r	5.333	4.361	3.323	2.256

Table 3.6: The values of the average efficiency for some selected designs for 2 × 2 experimental layout when the main effects and interaction are of equal interest.

the missing observations is more than two. This shows that when the main effects and interaction are of equal interest the *XL/BS* design is preferable if the number of missing array is less than two, otherwise, *CL/CL*$_r$ or *XL/XL*$_r$ design should be used for this case.

- If only the interaction is of interest, the *CL/BS* and *CL/CL*$_r$ designs are found to be the most efficient designs. The *CL/CL*$_r$ design is preferable to the *CL/BS* design because the latter is more robust than the former in terms of the minimum of the breakdown numbers and the average efficiencies (see Figure 3.6 for the distribution of the average efficiency over the number of missing arrays when interaction is the effect of interest).

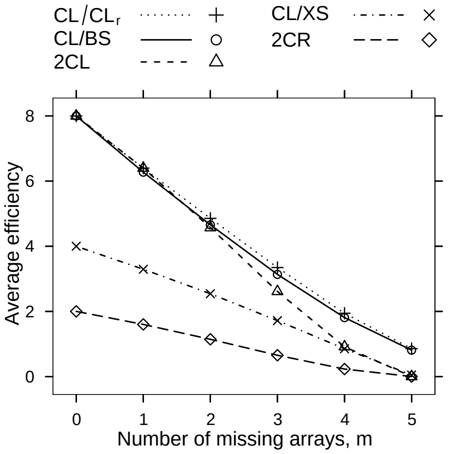


Figure 3.6: Distributions of the average efficiency with respect to interaction over the number of missing arrays for the designs for 2 × 2 experimental layout.

- If the effects $A \times B$ and B are of interest, the *XL/BS* and *CL/BS* designs are found to be the most efficient designs and both the designs are equally efficient with respect to the minimum of the corresponding breakdown numbers. In this case, the *XL/BS* design is more preferable to the *CL/BS* design because it can be used to estimate the third effect A more efficiently compared to the *CL/BS* design.

Reverse dye labelling plays an important role if more than one replication of a design is used to construct the composite design of interest. Similar to the one–way experimental layouts, it is observed that using about a half of the replications with reverse dye labelling improves the robustness of the design. For example, the *2CL* and *CL/CL*$_r$ designs can be used to estimate the effect $A \times B$ with the same efficiency if there is no missing observation, but the *CL*/*CL*$_r$ design is more robust than the *2CL* design in terms of the breakdown numbers.

3.3.2 Microarray Designs for 3×2 Experimental Layout

In a 3×2 factorial experiment, one factor, say, A has three levels and the other, say, B has two levels and we assume the effects of interest are the main effects (A, B) and interaction ($A \times B$). Landgrebe et al. (2004) discussed some basic types of the microarray designs for the 3×2 experimental layout and reported the efficient designs for estimating different combinations of the effects of interest. The basic designs, circular loop (*CL*), cross–loop (*XL*), triangular loop (*TL*), *A*–loop (*AL*), *B*–swap (*BS*), and star-swap (*RS*), for 3×2 experimental layout are graphically shown in Figure 3.7.

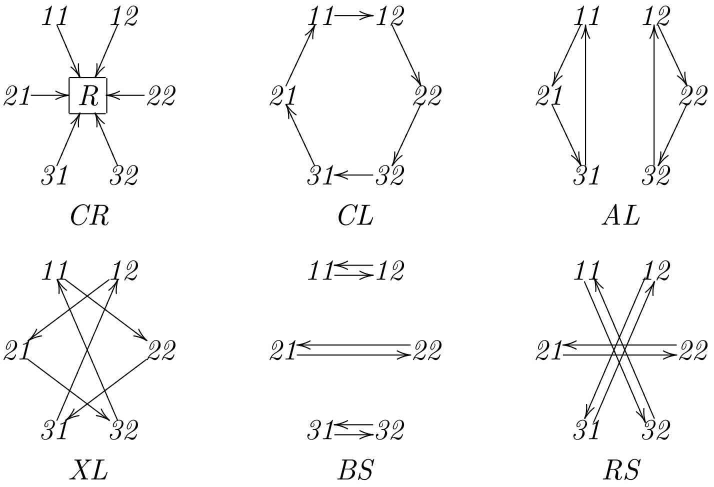


Figure 3.7: Graphical representations of the basic microarray designs for 3×2 experimental layout, each of which has six arrays. Treatment combinations are specified by a pair of the treatment labels corresponding to the factors A and B.

Table 3.7 shows the values of the E–optimality criterion and proportion of the effective designs for the basic designs with respect to the effects of interest. Among the basic designs with six arrays, the *CR*, *XL*, and *CL* designs can only be used to estimate all three effects of interest. The *XL* design is found to be more efficient than the *CR* design for estimating the effects A and B, and the *CL* design is more efficient than the *CR* design for estimating the effects $A \times B$ and B. If all three effects are of equal interest, the *CL* and *XL* designs are found to be more efficient than the *CR* design in terms of the overall efficiency. and the *XL* design is more efficient than the *CL* design. The *CL* design can be used if only the interaction is of main interest.

	E–optimality				pED_1		
Design	A	B	$A \times B$	Overall	A	B	$A \times B$
CR	2.00	1.00	2.00	1.67	0.00	0.00	0.00
XL	6.00	4.00	2.00	4.00	0.00	0.00	0.00
CL	2.00	1.10	6.00	2.70	0.33	0.00	0.00
AL	6.00	NA	6.00	NA	1.00	NA	1.00
BS	NA	4.00	8.00	NA	NA	1.00	1.00
RS	NA	4.00	NA	NA	NA	1.00	NA

Table 3.7: The values of the E–optimality criterion and proportion of effective designs with one missing array for the basic designs for 3 × 2 experimental layout.

The designs with 12 arrays for the 3×2 experimental layout are constructed from the combinations/replications of the corresponding basic designs. In this case, the number of the candidate designs is large and the concept of the non–inferior designs (see §2.3.2 for the definition) is used to reduce the number of candidate designs. Among the 15 possible loop designs with 12 arrays, six designs are found to be non–inferior. The E–optimality values and breakdown numbers of the *2CR*, *2XL*, and non–inferior designs are reported in Table 3.8. Except the *CL/AL* design, all the non–inferior designs can be used to estimate all three effects more efficiently than the *2CR* design when there is no missing observation in the data. However, the *CL/AL* design is more efficient than the *2CR* design with respect to the effects A and $A \times B$.

	E–optimality				*BDN*			
Design	A	B	$A \times B$	Overall	A	B	$A \times B$	min
2CR	4.00	2.00	4.00	3.33	2	2	2	2
AL/XL	12.00	4.00	8.00	8.00	4	4	4	4
AL/BS	6.00	4.00	14.00	8.00	4	4	4	4
XL/BS	6.00	8.00	10.00	8.00	4	4	4	4
XL/XL_r	12.00	8.00	4.00	8.00	4	4	4	4
CL/XL	8.00	5.14	8.00	7.05	4	4	4	4
CL/AL	8.00	1.10	12.00	7.03	4	2	4	2
2XL	12.00	8.00	4.00	8.00	2	2	2	2

Table 3.8: The values of the E–optimality criterion and breakdown number for the designs for 3 × 2 experimental layout.

- If all three effects are of equal interest, the *AL/XL*, *AL/BS*, XL/XL_r, and *XL/BS* designs are the most efficient in terms of the overall efficiency and the most robust in terms of the minimum of the breakdown numbers. Table 3.9 displays a comparison between these four designs on the basis of the average efficiency. It shows

	Number of missing arrays, m			
Design	0	1	2	3
XL/BS	8.000	6.424	5.197	4.082
XL/XL$_r$	8.000	6.286	5.097	3.975
AL/BS	8.000	6.252	5.096	3.974
AL/XL	8.000	6.198	4.990	3.811

Table 3.9: The values of the average efficiency for some selected design for 3 × 2 experimental layout when the main effects and interaction are of equal interest.

that the design *XL/BS* outperforms the other three designs if there is at least one missing observation in the data.

- If the effects A and $A \times B$ are of interest, the *CL/AL*, *AL/XL*, and *AL/BS* designs are the most efficient ones, however, the *CL/AL* design is less robust compared to the other two designs.
- The *AL/BS* and *XL/BS* designs are found to be the best designs when the effects $A \times B$ and B are of interest and both of these designs are equally robust.

Figure 3.8 shows the distribution of the average efficiency and proportion of the effective designs of the estimates of the interaction for different designs over the number of missing arrays. The *AL/BS* design is found to be the most efficient design up to six

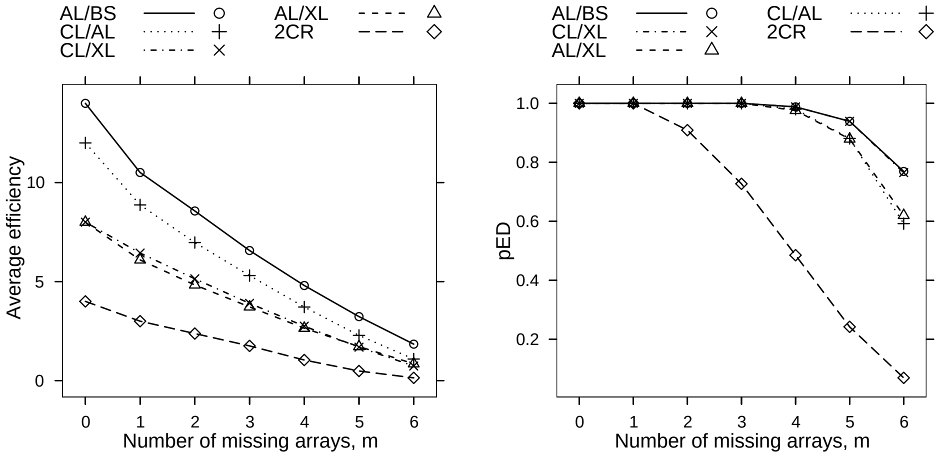


Figure 3.8: Distributions of the E–optimality criterion and the proportion of the effective designs corresponding to interaction over the number of missing observations for the designs for 3 × 2 experimental layout.

missing arrays and most of the designs (except *XL/XL$_r$*) are found to be more efficient than the *2CR* design up to four missing arrays. It also shows the proportion of the designs that can estimate the interaction in the presence of the missing arrays. About 10% of the residual designs with two missing arrays corresponding to the *2CR* design cannot be used to estimate the interaction.

3.4 Conclusion

In this chapter, the use of the efficiency and robustness criteria to select good microarray designs from a set of candidate designs are described. Designs from both the one–way and multi–factor factorial experiments are considered. The pairwise treatment comparisons are considered as the effects of interest for the one–way factorial experiments and the main effects and interaction are considered as the effects of interest for the multi–factor factorial experiments. For different experimental layouts, the common reference design is compared with the loop and other basic/composite designs in terms of the robustness and efficiency criteria.

Designs from two one–way experimental layouts (1×3 and 1×4) with different number of arrays are considered. For the pairwise treatment comparisons, the common reference designs are less efficient compared to the loop designs. However, the common reference design is more robust than the loop design if only one replication of the design is considered, i.e., if $n = K$, where n is the number of available arrays and K is the number of treatments to be compared. That means, for a 1×3 experimental layout the common reference design is preferable to the loop design if the experimenter is interested to conduct an experiment with three arrays only. But the performance of the common reference designs does not remain the same if the number of available arrays is large enough to consider at least two replications of the designs, i.e., if $n \geq 2K$. In this case, the common reference design is found to be less robust compared to the loop designs, e.g., for the 1×3 experimental layout the *CL*/*CL*$_r$ design is preferable to the *2CR* design. Reverse dye labelling improves the robustness of the loop designs, but does not do so for the corresponding common reference design. More specifically, the *CL*/*CL*$_r$ design is more robust than the *2CL* design, but in terms of the robustness no improvement can be observed if the *CR*/*CR*$_r$ design is used instead of the *2CR* design. For the 1×4 experimental layout, even a design with nine arrays (e.g., *B–T* design) can outperform the common reference design with 12 arrays in terms of the robustness and efficiency criteria. The best designs for the 1×3 and 1×4 experimental layouts are reported in Table 3.10 for different numbers of arrays.

For the multi–factor factorial experiments, the basic designs for the 2×2 and 3×2

Layout	n	Effects	Best designs
1×3	6	$\tau_1 - \tau_2$	CL/CL_r
	9	$\tau_1 - \tau_2$	$2CL/CL_r$
	12	$\tau_1 - \tau_2$	$2L/2L_r$
1×4	8	$\tau_1 - \tau_2$	CL/CL_r
	12	$\tau_1 - \tau_2$	$2CL/CL_r$
	12	all-pairs	DS

Table 3.10: The best designs for the experimental layouts 1 × 3 and 1 × 4 with different number of arrays.

layouts are chosen from the literature (Landgrebe et al., 2004) and the main effects and interaction are assumed to be the effects of interest. For a fixed number of arrays, we reported a number of designs that are more efficient and robust than the corresponding common reference design for different combinations of the effects of interest. The common reference designs are found to be less robust compared to the loop designs. Except for the common reference designs, the reverse dye labelling can improve the robustness of the designs when there are at least two replications of a basic design are used. The best designs for the 2×2 and 3×2 experimental layouts are reported in Table 3.11.

Layout	n	Combinations of the effects			
		$A \times B$	$A \times B$, A	$A \times B$, B	$A \times B$, A, B
2×2	4	CL		CL, XL	CL, XL
	8	CL/CL_r		XL/XL_r, CL/CL_r	CL/XS, CL/XL_r XL/XL_r, CL/CL_r
3×2	6	CL	CL, XL	CL	XL
	12	AL/BS	AL/XL, AL/BS	AL/BS, XL/BS	AL/BS, XL/XL_r AL/XL, XL/BS

Table 3.11: For different combinations of effects, the best designs for the experimental layouts 2 × 2 and 3 × 2 with different number of arrays.

In this chapter, the efficient and robust designs are selected from a small set of candidate designs. For a given experimental layout or number of treatment combinations, the complete set of candidate designs could be very large. One should examine all the possible candidate designs to select the most efficient design. In the next chapter, a procedure of selecting good designs is described which considers both the efficient and robustness criteria in the selection process and examines almost all the possible candidate designs.

Chapter 4

Introduction to Genetic Algorithms for Microarray Designs

4.1 Introduction

An efficient design ensures smaller variance of the estimates of the effects under investigation compared to an inefficient design. Efficient estimates are important for statistical analysis because inefficient estimates may lead to unreliable conclusions. In the context of microarray experiments, only a few papers have been published so far on the considerations of the efficiency criteria in selecting good designs. However, the inefficiency of the commonly used common reference designs has been pointed out, both theoretically and empirically, in several studies (e.g., Kerr and Churchill, 2001b; Landgrebe et al., 2004; Vinciotti et al., 2005).

Kerr and Churchill (2001b) first investigated the efficiency of microarray designs and considered A–optimality as the efficiency criterion for evaluating designs for the one–way factorial experiments. For a $1 \times K$ experimental layout, they reported that the circular loop design is A–optimal among the designs with K arrays when $K \leq 8$. The A–optimal designs with $(K + 2)$ arrays are also reported for $K \leq 13$. For the designs with $2K$ arrays, the interwoven loop designs are found to be A–optimal if $5 \leq K \leq \mathbf{1}0$. Yang et al. (2002b) suggested efficient designs for both the time–course and two–factor factorial microarray experiments. They considered the overall efficiency as an efficiency criterion. The overall efficiency criterion is useful when more than one effect is of equal interest. Glonek and Solomon (2004) suggested to search efficient designs from the class of admissible designs. The admissibility concept is not a new concept (Kiefer, 1959) which is commonly used in the statistical decision theory (Casella and Berger, 1990, §10.4). It states that no non–admissible design can be used to estimate any of the effects of interest more efficiently compared to an admissible design. This procedure reduces the size of the search space, but is not feasible when a large number of treatments are to be compared. Landgrebe et al. (2004) suggested some basic designs for the 2×2 and 3×2 experimental layouts and constructed composite designs by using the combinations/replications of the basic designs. They considered E–optimality (see §2.3.1 for details) as the efficiency criterion and reported the efficient basic and composite designs for different numbers of arrays. All of these studies suggested alternatives to the common reference designs without explicit efficiency calculations and none of these studies provides a general method for selecting efficient microarray designs for a fixed number of available arrays and a given set of experimental questions.

Microarray experiments can be considered as incomplete two–factor block experiments of block size two when more than two treatments are to be compared (Kerr and Churchill, 2001b). An incomplete block design is said to be balanced if each pair of the treatments appears together in the same number of blocks and any treatment does not appear more than once in any block. John and Mitchel (1977) defined a class of regular graph designs which contains only those incomplete block designs for which the number of occurrences of any two treatment pairs does not differ by more than one. The balanced incomplete block designs, if exists, are optimal with respect to a very general class of efficiency criteria including the E–optimality criterion (Cheng, 1980). John and Mitchel (1977) conjectured that an optimal incomplete block design is a regular graph design if it exists. Methods of constructing optimal block designs by using computer algorithms are discussed in several studies (e.g., Nguyen, 1994; Whitaker et al., 1990).

The efficient designs that can be obtained from the literature on incomplete block designs have little practical importance in the context of microarray experiments. This is because the underlying strategy for these studies is to define families of optimal designs, not to find a good design for a fixed number of blocks and a given set of research questions. Moreover, the balanced incomplete block or regular graph designs exist only for suitable combinations of the number of blocks (arrays) and number of treatments. But in microarray experiments, the number of arrays the experimenter wants to conduct the experiment with, depends on the available resources and the effects of interest which could be different for different experiments. All the above mentioned procedures are described only for the one–way experimental layout with all possible pairwise comparisons as the effects of interest.

Microarray data often contain missing observations due to unreliable spot measurements. The effects of interest could be less efficient or even non–estimable in the presence of missing observations. In §2.4.1, the robustness criteria are proposed which can be used to assess the quality of a design in the presence of missing observations. So far, no attempt has been made to incorporate the robustness considerations in the search for good microarray designs.

The main objective of this chapter is to develop a general procedure for selecting good microarray designs which can be used for both the one–way and multi–factor factorial experiments. To select good designs, a naive approach would be to evaluate all the possible designs that can be constructed for the given experimental layout and the number of arrays the experimenter wants to conduct the experiment with. The experimental layout specifies the number of possible arrays, e.g., for a 3×2 experimental layout, which has six treatment combinations, a total of 30 $(= 2 \cdot \binom{6}{2})$ arrays can be considered. The naive search of all possible candidate designs could be computationally infeasible if the number of available arrays is large, e.g., for the 3×2 experimental layout, a total of 1,623,160 designs can be constructed with six available arrays and it takes about 22 hours to compute the E–optimality criterion corresponding only to the main effects and interaction by using a C program which was written to implement the naive search. In this chapter, genetic algorithms (GAs) are used to develop a search procedure which can be used to select good microarray designs for a given number of treatment combinations and a fixed number of available arrays. This procedure is flexible enough to incorporate robustness considerations in the search process.

A genetic algorithm is a stochastic search technique that mimics some common features of natural evolution to find near–optimal, if not optimal, solutions of the problem under investigation. It encodes the problem into a chromosome–like data structure where each chromosome represents a search point in the space of the potential solutions of the problem. Holland (1975) first introduced the algorithm in the mid 70's and since then, it has been widely applied to a broad range of problems including combinatorial optimization. GAs are computationally simple, but powerful in their search for improvement and can provide a robust search in complex spaces (Goldberg, 1989).

GAs deal with a population of chromosomes that evolves over generations for the improvement of the quality of the population. The quality of a chromosome is assessed by the value of the corresponding objective function of the problem. The goal of a search is to find the solutions for which the objective function is optimal or at least near–optimal. The canonical GAs mimic three most important mechanisms of natural evolution, namely, selection, inheritance, and variability, to define its own operators.

The *GA* operators are used to generate chromosomes for the next generation (offspring chromosomes) from the current population of chromosomes. The genetic algorithm works under the assumption that quality of the offspring chromosomes are better compared to their parents and after evolving a reasonable number of generations at least a near–optimal solution of the problem can be reached. A pseudo code of the canonical genetic algorithm is given in Table 4.1.

```
create initial population
evaluate the fitness of each individual of the population
repeat
   select parent population (selection)
   select pairs at random from parent population for mating
   apply crossover operator to each pair of parents (inheritance)
   apply mutation operator to each offspring (variability)
   evaluate fitness of each individual of the new population
until terminating condition
```

Table 4.1: Pseudo code for canonical Genetic Algorithm.

The widely used calculus–based search methods, e.g., Newton–Raphson method, are restrictive to some assumptions like continuity and existence of the derivatives of the objective function. Moreover, these methods are less efficient to find an optimum when the objective function is multimodal. On the other hand, such assumptions are not required for *GA*s. It works with a large number of solutions simultaneously, so different paths for searching improved solutions can be considered in parallel. That means, *GA* exploits the search space very highly, hence there would be less probability of finding a false optimum compared to the calculus–based methods which usually consider a single path for updating intermediate solutions.

Besides the schema theorem (Holland, 1975), the mathematical foundation of the genetic algorithms is not well developed. A schema is a similarity template which describes a subset of strings with similarities at some string positions. The quality of a schema can be characterized by its order (number of the fixed positions in the template) and defining length (distance between first and last defining positions). The schema theorem states that the number of low–order and fitted schemata is increased exponentially over generations. Recently, Greenhalgh and Marshall (2000) showed that genetic algorithms can search a global optimum with any specified level of confidence if it runs for a sufficiently long time, i.e., a genetic algorithm converges in probability to a global optimum.

This chapter presents an introduction of the genetic algorithms in the context of microarray designs. The problem and fitness function will be described in §4.2, the proposed encoding method is discussed in §4.2.2, and the important operators of the genetic algorithm are described in §4.3.

4.2 The Problem and Fitness Function

In the problem of selecting good microarray designs, the number of arrays n, the experimenter wants to conduct the experiment with, and the number of treatments K or the experimental layout $\mathcal{L}$ under investigation must be known beforehand. For a multi–factor factorial experiment, the number of the treatments is the product of the treatment levels associated with the factors under investigation. In microarray experiments, each array compares a pair of the treatments and by using a reverse dye labelling protocol, at most two different arrays can be considered for each pair of the treatments. So, the total number of the arrays that can be constructed from the available K treatments is $N(K) = 2 \cdot \binom{K}{2}$.

Let $\mathcal{A} = \{a_i,\, i = 1, 2, \ldots, N(K)\}$ be the set of the possible $N(K)$ arrays, where a_i is the label of the i^{th} array. Let $\mathbf{C}'\boldsymbol{\beta}$ be the effect of interest where $\mathbf{C}$ is a contrast matrix and $\boldsymbol{\beta}$ is the vector of the regression parameters of the gene–specific *ANOVA* model

$$\mathbf{Z} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

This model is described in §2.2 of this dissertation. It can be shown that there are $T(n, K) = \binom{N(K)+n-1}{n}$ ways n arrays can be selected from the set $\mathcal{A}$ in such a way that unlimited repetitions are allowed (with replacement) (Jackson and Thoro, 1990, page 67).

Let $\mathcal{H}(n, K) = \{\xi_t \mid t = 1, 2, \ldots, T(n, K)\}$ be the set of $T(n, K)$ selections of n arrays and each of the selections is known as an experimental design of size n for the experimental layout $\mathcal{L}$. Formally, a design $\xi \in \mathcal{H}(n, K)$ can be defined as

$$\xi = \big\{a_{i_1}, a_{i_2}, \ldots, a_{i_n} \mid i_k \in \{1, 2, \ldots, N(K)\}, \forall\, k = 1, 2, \ldots, n\big\},$$

where a_{i_k} is the ${i_k}^{\text{th}}$ element of $\mathcal{A}$. The order of the arrays in a microarray design is not important. For notational simplicity, we use $N = N(K)$ and $T = T(n, K)$ in the remaining of this thesis.

To select the best design from a set of candidate designs, we follow the procedure that is commonly used in statistical inference for finding the best unbiased estimator of a parameter of interest (e.g., Casella and Berger, 1990, §7.3). The procedure has two steps, first, select the designs by which an unbiased estimate of the effect of interest can be obtained, i.e., examine whether the effects are estimable (see §2.2.2 for the details on the estimability concept in the context of microarray designs). In the search process, estimability is a necessary condition because no design is selected as the best design unless the effect is estimable with respect to the design. If more than one effect is of interest, only the designs for which all the effects are estimable can be selected. In the second step, the quality of the designs for which the effect is estimable, is assessed by its ability to provide an estimate of the effect with a smaller variance. In practice, efficiency criteria are used to quantify the quality of a design with respect to the effect of interest.

In a *GA*, the fitness function measures the quality of a chromosome in terms of a scalar quantity which is known as fitness. For a chromosome of length n, the fitness function can be defined as $f : \mathbb{R}^n \to \mathbb{R}$. Actually, fitness is the quantity that connects the *GA* to the problem under investigation. In this application of the *GA*, the efficiency criterion is used as the fitness function. So, the fitness is an estimate of the efficiency criterion.

The problem of finding good microarray designs can be formally defined in terms of the E–optimality criterion as

$$\left.\begin{array}{lc} \text{maximize} & \phi(\xi, \mathbf{C}'\boldsymbol{\beta}),\ \forall\, \xi \in \mathcal{H}(n, K) \\ \text{subject to} & \mathbf{C}'(\mathbf{X}_t'\mathbf{X}_t)^-(\mathbf{X}_t'\mathbf{X}_t) = \mathbf{C}',\ \ \forall\, \xi_t \in \mathcal{H}(n, K), \end{array}\right\} \tag{4.1}$$

where $\phi(\xi, \mathbf{C}'\boldsymbol{\beta})$ is the E–optimality criterion of the design ξ with respect to the effect of interest $\mathbf{C}'\boldsymbol{\beta}$ and $\mathbf{X}_t$ is the design matrix of the design ξ_t. If more than one effect is of interest, instead of the E–optimality criterion $\phi(\cdot, \cdot)$, the overall efficiency $\bar{\phi}(\cdot, \cdot)$ can be used in (4.1). A brief description of the efficiency criteria, especially the E–optimality criterion, is given in §2.3.1.

This is a constrained optimization problem for which the search space contains both feasible and infeasible solutions. The feasible solutions, which we are interested in, correspond to the designs for which the effect of interest $\mathbf{C}'\boldsymbol{\beta}$ is estimable. Since each design of the set $\mathcal{H}(n, K)$ can be considered as a solution of the optimization problem (4.1), the term "feasible designs" is used in the remaining of this section for the feasible solutions and the "infeasible designs" is used for the infeasible solutions.

We already mentioned that microarray data often contain missing observations due to unreliable spot measurements. So, the experimenter may be interested not only in efficient designs but also in robust designs. To quantify robustness considerations in the context of microarray designs, three criteria have been proposed in §2.4.1. To incorporate robustness in the search procedure, the problem of finding good microarray designs (4.1) can be re–defined for $M(< n)$ missing arrays as

$$\left.\begin{array}{ll} \text{maximize} & \sum_{m=0}^{M} \bar{\phi}_m(\xi, \mathbf{C}'\boldsymbol{\beta}),\ \forall\, \xi \in \mathcal{H}(n, K) \\ \text{subject to} & \mathbf{C}'(\mathbf{X}_t'\mathbf{X}_t)^{-}(\mathbf{X}_t'\mathbf{X}_t) = \mathbf{C}',\ \ \forall\, \mathbf{X}_t \in \mathcal{R}(\mathbf{X}_n),\ \xi_t \in \mathcal{H}(n, K), \end{array}\right\} \tag{4.2}$$

where $\bar{\phi}_m(\xi, \mathbf{C}'\boldsymbol{\beta})$ $(m = 1, 2, \ldots, M)$ is the average efficiency with m missing arrays (see §2.4.1), $\mathcal{R}(\mathbf{X}_n)$ is the set of the residual designs with m missing arrays corresponding to the design matrix $\mathbf{X}_n$ (see §2.11), and the design matrix of the design ξ_t is $\mathbf{X}_n$.

In practice, all the residual designs corresponding to a feasible design may not be feasible. For infeasible designs the E–optimality criterion cannot be defined, hence the fitness cannot be defined. Infeasible designs may appear in the intermediate populations too. To overcome this problem, a real–valued penalty can be assigned to an infeasible design or the infeasible designs can be excluded from the analysis. In the following section, the use of the penalty function in genetic algorithm is described in the context of microarray experiments.

4.2.1 Penalty Function

Though *GA*s are widely used for unconstrained optimization problems, their applications can also be found in constrained optimization problems (Michalewicz, 1992). The constraint optimization problems may contain infeasible solutions because some solutions may not satisfy the given constraints. Two main approaches have been considered in *GA*s for handling constraints (Richardson et al., 1989; Goldberg, 1989), which are:

(i) excluding all infeasible solutions from the intermediate populations and

(ii) assigning a penalty to the infeasible solutions.

In the context of microarray design, the intermediate populations may contain infeasible designs for which the effects of interest are non–estimable. Although infeasible designs are not of interest at the end, these should not be excluded from the intermediate populations because a feasible design could be very close to an infeasible design in the search space. A feasible design can be obtained from an infeasible design by changing some of the arrays which can be done by the *GA* operators. The penalty function depends on the problem under investigation, but it must satisfy the following two conditions:

- the minimum fitness of the feasible solutions must be greater than the maximum fitness of the infeasible solutions,
- the fitness of an infeasible solution with a smaller number of violations of the constraints must be greater than the other infeasible solutions.

One simple penalty function for the problems similar to (4.1) or (4.2) would be considering zero as the fitness for the infeasible designs. If only an effect is of interest, this approach guarantees that the fitness of a feasible design must be greater than that of an infeasible design. In case of more than one effect, fitness of a design can be considered as the sum of the fitnesses corresponding to the feasible designs. That means, only the estimable effects contribute to the fitness.

4.2.2 Encoding the Problem

Encoding of a *GA* specifies the procedure of expressing a solution of the problem under investigation in terms of a chromosome and genes, which are considered as the data structure of *GA*s. In the context of microarray experiments, each design of the candidate set $\mathcal{H}(n, K)$ is considered as a solution of the problem (4.1) and so as a chromosome. In *GA*s, genes are assumed to be the components of the chromosomes and in microarray experiments, arrays are the components of the microarray designs. That means, an individual array can be considered as a gene of *GA*s.

Binary coding is the most commonly used encoding procedure but applications with other approaches such as real–valued or gray coding can also be found in the *GA* literature. Davis (1991) reported a better performance of the non–binary coding in different applications. The choice of the encoding procedure depends on the nature of the problem under investigation. In the context of microarray designs, two different encoding procedures can be considered.

- The first possibility is to consider a string of natural numbers of length N, the number of possible arrays, to represent a design. For a design with n arrays, the N positions of the string is filled by natural numbers in such a way that the sum of the numbers equals n. The natural numbers represent the number of times the corresponding array is replicated in the design.
- Another possibility is to consider a string of array labels of length n to represent a design of size n.

We call these two encoding procedures as natural and label coding, respectively. These coding procedures are graphically described in the following example.

Consider a 1×3 experimental layout where three treatments *1*, *2*, and *3* are compared with each other. A graphical representation of the possible six arrays for the 1×3 experimental layout is shown in Figure 4.1(a) where a pair of treatment labels and an arrow are used to represent an array. For example, $\mathit{1} \longrightarrow \mathit{2}$ represents the array on which the treatments *1* and *2* are hybridized. The treatments at the arrow head and arrow tail are labelled with a green and a red dye, respectively. A design with four arrays $\{a_2, a_2, a_3, a_6\}$ is arbitrarily chosen and is graphically shown in Figure 4.1(b). The corresponding natural and label coding are shown in Figure 4.1(c) where a circle is used to represent an array and the natural numbers or array labels are printed inside the circles to specify the arrays of the design. In this example, N and n take the value six and four, respectively.

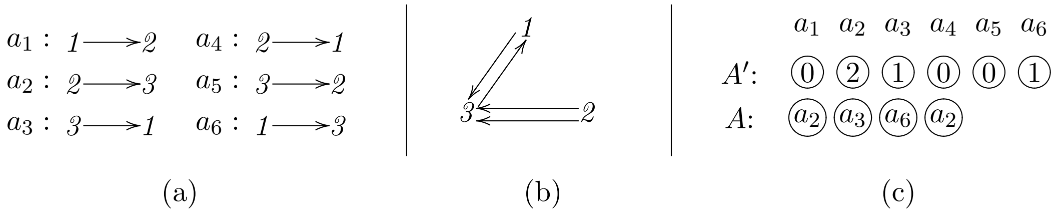


Figure 4.1: (a) possible arrays for 1 × 3 experimental layout, (b) a specific design with four arrays for the 1 × 3 experimental layout, (c) representation of the design in (b) in terms of the natural (A') and label (A) coding.

In our implementation of the *GA*, the label coding is used for encoding the problem. In the context of microarray experiments, the main problem of using natural coding is that the number of arrays n can vary over generations because of using a crossover operator to generate offspring population (see §4.3.2 for more explanations).

4.3 Genetic Algorithm Operators

4.3.1 Selection Operator

The selection operator of a *GA* specifies a scheme by which parent chromosomes are selected from the current population of chromosomes. The selection scheme follows the rule *survival of the fittest*, i.e., chromosomes with high fitness will have more chance to be selected in the parent population than those with low fitness. The selected parent chromosomes are then paired up for mating and hence, produce offspring chromosomes by using *GA*'s recombination operators: crossover and mutation. A number of selection schemes are available in *GA* literature (e.g., Goldberg, 1989). In the following sections, two selection schemes, namely, sampling proportional to fitness and remainder stochastic sampling, that are used in our implementation of the *GA*, are briefly described.

Sampling Proportional to Fitness

The sampling proportional to fitness *(SPF)* is a simple selection scheme for which a selection probability is assigned to each chromosome of the current population. The parent chromosomes are selected from the current population by using a sampling with replacement procedure where selection probabilities are used as the corresponding weights. The proportions of the individual fitness to the total fitness of the current population are often used as the selection probabilities. The selection probability corresponding to the i^{th} chromosome is $p_i = f_i / \sum f_i$, where f_i is the fitness of the i^{th} chromosome, $i = 1, 2, \ldots, P$, and P is the population size at each generation.

Remainder Stochastic Sampling

The remainder stochastic sampling *(RSS)* uses the expected fitness (Whitley, 1994; Goldberg, 1989) to select parent chromosomes from the current population. For the i^{th} chromosome of the current population, the corresponding expected fitness can be obtained by

$$f_i^* = P \frac{f_i}{\sum_{j=1}^{P} f_j},\ i = 1, 2, \ldots, P.$$

The *RSS* works in two steps, at the first step the integer part of the expected fitness is used as the number by which the corresponding chromosome of the current population is copied into the parent population. At the second step, remaining chromosomes of the parent population are selected randomly by using the fractional part of the expected fitness as the selection probabilities. This step is similar to the *SPF* procedure except the sampling is done without replacement instead of with replacement. For example, a chromosome with the expected fitness 3.46 indicates that the chromosome is copied three times in the parent population and also has a 46% chance of getting another copy into the parent population.

4.3.2 Crossover Operator

Crossover is a simple exploratory operator which is considered by some practitioners as the heart of the genetic algorithm. The crossover operator mimics inheritance of the natural evolution. It works on chromosome level and combines two parent chromosomes to produce two offspring chromosomes. The crossover operator is applied to a pair of parent chromosomes according to a user–specified crossover probability p_c which is usually considered between 0.50 to 0.90. Different forms of the crossover operator are used in the genetic algorithms. In our implementation, we have used three types of the crossover operator, namely,

(i) one–point,

(ii) two–points, and

(iii) uniform crossover.

A brief description of these three crossover operators is given in the following sections.

One–point Crossover

In the one–point crossover, first, an integer $l \in \{1, 2, \ldots, n-1\}$ is randomly selected and each of the parent chromosomes is cut into two parts by the position l. Then the head or tail segment of the parent chromosomes are swapped and the resulting pair of the chromosomes is considered as two offspring chromosomes. As an example consider two designs D_a and D_b for the 1 × 3 experimental layout, each of which has seven arrays. The offspring designs D'_a and D'_b are obtained by the one–point–crossover where the randomly selected integer is three and the tail segments of the parent designs are swapped. This example is graphically shown in Figure 4.2.

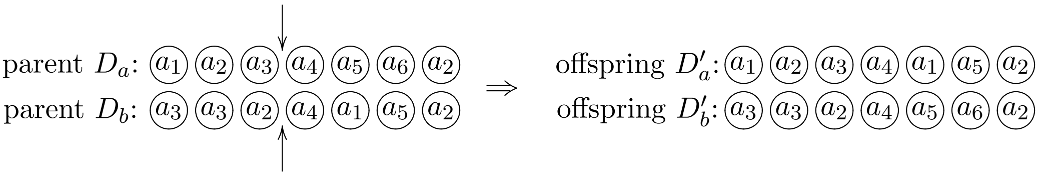

parent D_a: a_1 a_2 a_3 a_4 a_5 a_6 a_2
parent D_b: a_3 a_3 a_2 a_4 a_1 a_5 a_2
⇒
offspring D'_a: a_1 a_2 a_3 a_4 a_1 a_5 a_2
offspring D'_b: a_3 a_3 a_2 a_4 a_5 a_6 a_2

Figure 4.2: Graphical representation of the one–point crossover operator with the label coding.

Figure 4.3 shows the above example with the natural coding. In this case, the number inside the circles indicate the number of times the corresponding array is replicated in the design, e.g., the array a_2 is replicated twice in the parent design D_a. Though the coding methods are different, the parent designs D_a and D_b of Figures 4.2 and 4.3 are composed of the same set of seven arrays. By using the same cut off integer three, the offspring designs D'_a and D'_b can be obtained. The main problem of using the natural coding in the context of microarray designs is that there is no guarantee that the number of arrays of the offspring designs will have the same number of arrays of its parent designs. For example, the offspring design D'_a and D'_b have six and eight arrays, respectively, whereas both the parent designs have seven arrays. Because of this limitation of the natural coding, the label coding is used in our implementation of the *GA*.

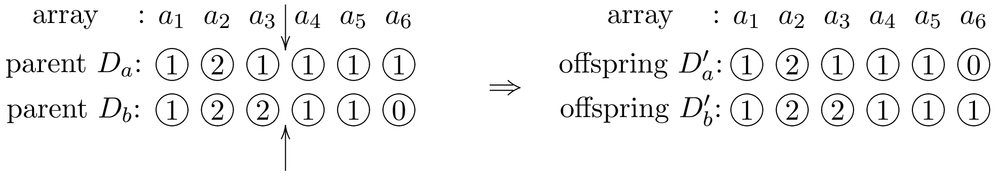

array : a1 a2 a3 a4 a5 a6
parent Da: 1 2 1 1 1 1
parent Db: 1 2 2 1 1 0
⇒
array : a1 a2 a3 a4 a5 a6
offspring D'a: 1 2 1 1 1 0
offspring D'b: 1 2 2 1 1 1

Figure 4.3: Graphical representation of the one–point crossover operator with the natural coding.

Two–points Crossover

In the two–points crossover, two distinct integers $l_1, l_2 \in \{1, 2, \ldots, n-1\}$ are randomly selected and each of the parent chromosomes is cut into three parts by the positions l_1 and l_2. In this case, two offsprings are obtained by swapping the middle segments of the parent chromosomes. Figure 4.4 shows an example of the two–points crossover with the parent designs D_a and D_b. The integers one and five are randomly selected and the arrays of the parent designs at the positions 2–4 are swapped to obtain offspring chromosomes D'_a and D'_b.

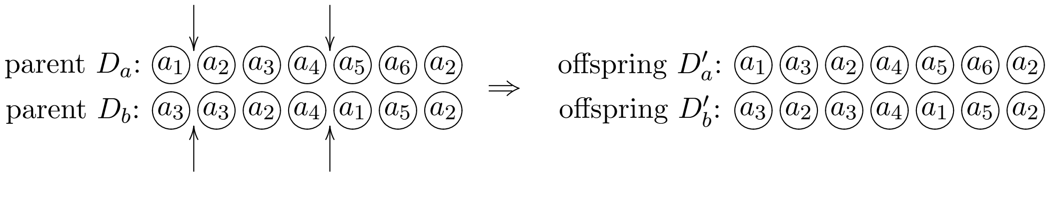

parent Da: a1 a2 a3 a4 a5 a6 a2
parent Db: a3 a3 a2 a4 a1 a5 a2
⇒
offspring D'a: a1 a3 a2 a4 a5 a6 a2
offspring D'b: a3 a2 a3 a4 a1 a5 a2

Figure 4.4: Graphical representation of the two–points crossover operator.

Uniform Crossover

Instead of randomly selecting one or two integers, in the uniform crossover a string of length n, which is known as crossover mask, is randomly chosen first. The elements of the crossover mask are randomly selected from the set $\{0, 1\}$. The genes of the offspring chromosomes are copied from both the parent chromosomes according to the appearance of the 0's and 1's in the crossover mask. For example, genes of the first parent are copied into the offspring chromosome at the positions where 1 appears in the crossover mask and similarly, genes of the second parent are copied at the positions where 0 appears in the crossover mask. For the second offspring chromosome, this procedure is repeated by reversing the labels of two crossover masks.

Figure 4.5 shows an example of the uniform crossover with two randomly selected crossover masks (c'over 1 and c'over 2). The arrays of the parent design D_a (D_b) are copied into the offspring design D'_a at the positions for which 1 (0) appears in the c'over mask 1 (2). For the offspring design D'_b, the arrays are copied from the parent design D_a (D_b) at the positions for which 0 (1) appears in the c'over mask 1 (2).

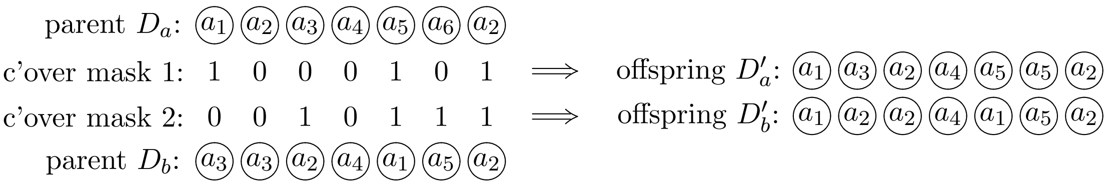

parent D_a: a_1 a_2 a_3 a_4 a_5 a_6 a_2
c'over mask 1: 1 0 0 0 1 0 1 ⟹ offspring D'_a: a_1 a_3 a_2 a_4 a_5 a_5 a_2
c'over mask 2: 0 0 1 0 1 1 1 ⟹ offspring D'_b: a_1 a_2 a_2 a_4 a_1 a_5 a_2
parent D_b: a_3 a_3 a_2 a_4 a_1 a_5 a_2

Figure 4.5: Graphical representation of the uniform crossover operator.

4.3.3 Mutation Operator

The mutation operator operates on the gene level to bring randomness to the search process. This operator mimics the variability of the natural evolution. It ensures that the entire search space is reachable and the process does not converge to a local optimum. In case of the binary coding of the fitness function, the mutation operator acts as a test for each of the genes of a chromosome to decide whether the associated bit will be flipped or not. The test is a Bernualli trial with a pre–specified mutation probability p_m which is usually considered as very small compared to the crossover probability. We slightly modify the procedure of the mutation operator because in our case, the chromosomes are represented by the strings of array labels instead of binary numbers. In our method, if the gene at the position $l \in \{1, 2, \ldots, n\}$ is selected for mutation then it is replaced by a randomly selected gene from the set of possible genes. For example, consider a design D_a which has seven arrays. Assume only the array at the 5^{th} position is selected for mutation. This array is replaced by a randomly selected array, say, a_2 and the resulting offspring design is denoted by D'_a. This example is graphically shown in Figure 4.6.

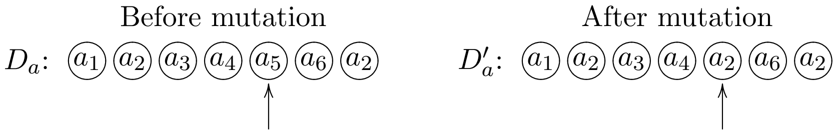

Before mutation
D_a: a_1 a_2 a_3 a_4 a_5 a_6 a_2
After mutation
D'_a: a_1 a_2 a_3 a_4 a_2 a_6 a_2

Figure 4.6: Graphical representation of the mutation operator.

The mutation operator plays an important role for increasing variability in the intermediate populations by inserting less frequent genes in the chromosomes. The choice of the mutation probability depends on the problem and it may affect the convergence of the process. A small value of the mutation probability may produce degenerate intermediate populations and hence the process may converge to a local optimum. On the other hand, a large value of the mutation probability may inject too much variability in the population for which the process may need too many generations to converge. As a compromise between these two extremes, use of an adaptive procedure for selecting mutation probability is getting popular in the *GA* community (Charbonneau, 2002). This method is based on a simple idea: it increases the mutation probability when the variability among the individual chromosomes is too small and decreases when it is too large.

4.4 Other Comments

4.4.1 Elitism

The elitism is a *GA* operator which ensures that the best chromosome (elite) of the population are preserved (Davis, 1991). The optimal solutions may appear at a generation and then disappear in the next generation due to crossover or mutation operator. Elitism can apply in two ways:

(i) replace the chromosome with minimum fitness of the offspring population by the elite of the parent population, and

(ii) store the elite of different generations without copying it to the population of the next generations.

Both the approaches are implemented in our **R** (R Development Core Team, 2004) package and our experience is that the first approach speeds up the convergence of the *GA*.

4.4.2 Stopping Rule

Because of its non–deterministic nature, the convergence criteria for the *GA* are not well defined compared to the other optimization procedures. Among the available convergence criteria, the simplest one is to run the *GA* for an arbitrarily fixed number of generations or till a pre–specified maximum evolution time. De Jong (1975) proposed the on–line and off–line strategies as the convergence criterion for the *GA*. The on–line strategy is defined as the average of all the fitness including the current population. On the other hand, the off–line strategy is a moving average of the best fitnesses to a particular time. The process can be stopped after either the on–line or off–line performance stabilizes. In our implementation of *GA*, a slightly modified version of the off–line strategy is used as the convergence criteria which depends on both the maximum and average fitness of the current population. The main steps of the convergence criteria are:

- the algorithm runs at least b generations to avoid premature convergence where b is termed as a burnout time,

- after running b generations, the *GA* is stopped if either the difference between the maximum and average fitness is very small or the best fitness remains the same in a reasonable number of successive generations.

In practice, the algorithm may stop if the maximum and average fitness become closer. In such a case the mutation and/or crossover probabilities need to be tuned to get at least b generations. The choice of the burnout time depends on the problem under investigation.

4.5 Conclusion

This chapter contains an introduction to the genetic algorithms in the context of microarray experiments. As far as we know, this is the first attempt of applying genetic algorithms in the context of microarray experiments. Thus, different operators of the *GA* are modified accordingly. For example, the label coding is proposed and used to encode the solutions of the problem instead of commonly used binary coding. This encoding procedure ensures that the crossover operators do not alter the fixed number of arrays of each design. The E–optimality criterion is used as a fitness function of the genetic algorithm. To incorporate robustness considerations in the search, the average efficiency, a robustness criterion, can also be used as a fitness function. Using its common operators, e.g., selection, crossover, mutation, etc., the *GA* can be used to optimize such fitness functions to find near–optimal, if not optimal, microarray designs for both one–way and multi–factor factorial experiment.

Chapter 5

Applications of Genetic Algorithms in Selecting Good Microarray Designs

5.1 Introduction

The problem of selecting efficient microarray designs can be considered as an optimization problem with a discrete search space (see equation 4.1). In this dissertation, we use a genetic algorithm to optimize such a problem. In the previous chapter, different operators of the genetic algorithm are defined in the context of the microarray experiments. In this chapter, the genetic algorithm is used to find good designs for a 3×2 experimental layout. The performance of the genetic algorithm in this context is assessed by applying it to one–way experimental layouts for which efficient designs are known for some specific numbers of arrays (Kerr and Churchill, 2001b).

An **R** (R Development Core Team, 2004) function is written to apply the genetic algorithm for selecting good microarray designs. The function is a part of our package *robustMAdesigns* which could be available on request or can be downloaded from the web site of the Department of Medical Statistics, University of Göttingen. The main inputs of this function are the number of available arrays n, the experimental layout $\mathcal{L}$, and the contrast matrices corresponding to the questions of interest. Instead of $\mathcal{L}$, the set of possible arrays $\mathcal{A}$ can also be used. The E–optimality criterion is used as the default fitness function, but other efficiency criteria such as D–, A–optimality can also be specified. The number of missing arrays can be specified to include robustness considerations in the search process. Different methods for the selection, crossover, and mutation operators can be specified. The number of designs for each generation, i.e., population size, must be specified and the default value is 50. The function provides a list of near–optimal, if not optimal, designs and the associated fitness values. A detailed description of the function can be found in Appendix *A*.

5.2 Efficient Designs for the 3×2 Experimental Layout

In a 3×2 factorial experiment, one factor, say, A has three levels *1*, *2*, and *3*, and the other factor, say, B has two levels *1* and *2*. Besides the interaction ($A \times B$), researchers could be interested in different combinations of the simple effects (A_{Bk}, $B_{Ak'}$) and main effects (A, B), $k = 1, 2$; $k' = 1, 2, 3$ where A_{Bk} denotes the simple effect of A at the k^{th} level of B. Landgrebe et al. (2004) suggested some basic designs for the 3×2 experimental layout (see Figure 3.7 for the graphical representations of the basic designs). Each of the basic designs has six arrays. The efficiency and robustness of the basic designs are investigated in §3.3.2 for different combinations of the main effects and interaction. The E–optimality values for different effects are illustrated in Table 5.1. Among the basic designs, only the *CR*, *CL*, and *XL* designs can be used to estimate all the simple effects, main effects, and interaction and the *XL* design is found to be the most efficient design if the simple effects, main effects, and interaction are considered as the effects of equal interest.

	Simple effects					Main effects		Int.	Overall eff.
Design	A_{B1}	A_{B2}	B_{A1}	B_{A2}	B_{A3}	A	B	$A \times B$	$\bar{\phi}$
CR	2.000	2.000	1.000	1.000	1.000	2.000	1.000	2.000	1.500
CL	3.000	3.000	2.400	1.333	2.400	2.000	1.091	6.000	2.653
XL	3.000	3.000	1.333	1.333	1.333	6.000	4.000	2.000	2.760
AL	6.000	6.000	*NA*	*NA*	*NA*	6.000	*NA*	6.000	6.000
BS	*NA*	*NA*	4.000	4.000	4.000	*NA*	4.000	8.000	4.800
RS	*NA*	*NA*	*NA*	4.000	*NA*	*NA*	4.000	*NA*	4.000

Table 5.1: The E–optimality and overall efficiency values of the basic designs for 3×2 experimental layout where *NA* indicates non–estimable effects and Int. denotes the interaction.

In this section, *GA*s are used to find efficient designs for the four arbitrarily chosen cases and the definitions of these cases are given in Table 5.2. For each case, efficient designs are selected for six, eight, 10, 12, 14, 15, 16, and 18 arrays. Throughout this chapter, the letter D is used to represent a design and the associated number of arrays and the specific case are defined in the subscript, e.g., D_{na} denotes a design with n arrays

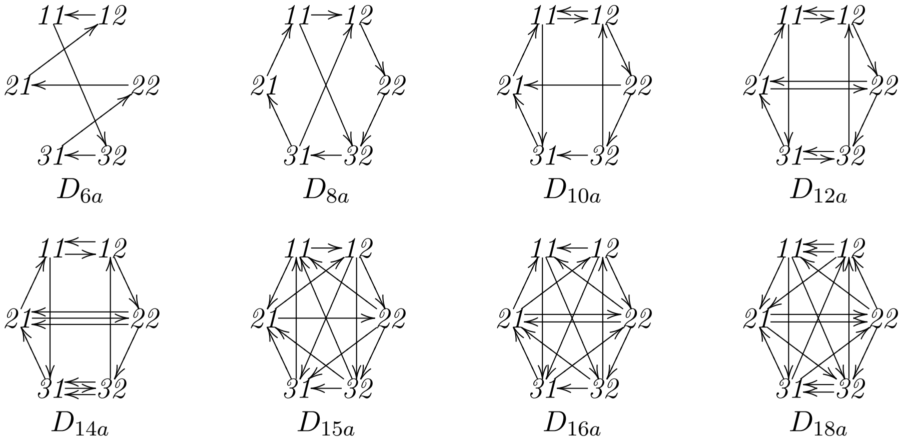
Different	Simple effects					Main effects		Int.
cases	A_{B1}	A_{B2}	A_{B3}	B_{A1}	B_{A2}	A	B	$A \times B$
Case a	✓	✓	✓	✓	✓	✓	✓	✓
Case b	✓	✓				✓	✓	✓
Case c			✓	✓	✓	✓	✓	✓
Case d						✓	✓	✓

Table 5.2: Different combinations of the simple effects, main effects, and interaction for which good designs for 3×2 experimental layout are reported.

for the *Case a*. In the following sections, the efficient designs for the above mentioned four cases are reported.

5.2.1 Efficient Designs for the *Case a*

For the *Case a*, all the simple effects, main effects, and interaction are considered as the effects of interest. For this case, the selected efficient designs for different number of arrays are shown in Figure 5.1. The associated E–optimality and overall efficiency values are illustrated in Table 5.3. The D_{6a} design is the most efficient design with six arrays and is found to be more efficient than the most efficient basic design *XL*. The D_{8a}, D_{10a}, D_{12a}, and D_{14a} designs can be obtained by adding suitable arrays to the basic *CL* design. The basic *AL* design can be observed in the designs D_{10a}, D_{12a}, D_{14a}, and D_{16a}. The D_{16a} design can be obtained by adding four suitable arrays to the *AL/XL* design. The D_{12a} design is composed of the arrays of the designs *AL* and *BS*, i.e., D_{12a}



D_{6a} D_{8a} D_{10a} D_{12a}

D_{14a} D_{15a} D_{16a} D_{18a}

Figure 5.1: Graphical representations of the selected microarray designs for 3×2 experimental layout with respect to the effects of the *Case a*.

	Simple effects					Main effects		Int.	$\overline{\phi}$
Design	A_{B1}	A_{B2}	B_{A1}	B_{A2}	B_{A3}	A	B	$A \times B$	*Case a*
D_{6a}	3.000	3.000	2.400	2.400	2.400	2.400	4.000	4.000	2.950
D_{8a}	6.000	6.000	3.231	1.556	3.231	6.000	1.680	6.000	4.212
D_{10a}	7.500	7.500	5.231	3.119	3.119	6.000	2.194	10.000	5.583
D_{12a}	8.400	8.400	5.600	5.600	5.600	6.000	4.000	14.000	7.200
D_{14a}	8.571	8.571	7.554	5.714	7.554	6.000	5.217	15.000	8.023
D_{15a}	12.000	12.000	5.860	5.860	5.860	12.000	5.600	12.000	8.898
D_{16a}	12.000	12.000	5.872	8.000	5.872	12.000	6.345	12.000	9.261
D_{18a}	13.714	13.714	8.000	8.000	8.000	12.000	8.000	16.000	10.929

Table 5.3: The E–optimality and overall efficiency values corresponding to the effects of the *Case a* for the selected microarray designs for 3 × 2 experimental layout.

is a AL/BS design.

For the D_{10a} design, the treatments *11* and *12* require more *mRNA*s than the other treatments because for this design, two arrays are considered for comparing this pair of treatments, whereas one array is considered for the other pairs. However, considering an additional array for the treatment pairs (*21, 22*) or (*31, 32*) does not affect the overall efficiency. So in practice, depending on the amount of the available *mRNA*s researchers can choose the treatments on which the additional array should be considered. This scenario is also observed in the D_{14a} and D_{16a} designs. The D_{15a} design is constructed by using one array from all the 15 different possible pairs of the treatments.

5.2.2 Efficient Designs for the *Case b*

For the *Case b*, the simple effects of A, both the main effects, and the interaction are considered as the effects of interest. The graphical representations of the selected efficient designs are shown in Figure 5.2. The corresponding E–optimality and overall efficiency values are illustrated in Table 5.4. Since the simple effects of B are not of interest, the selected designs correspond to more efficient estimates of the main effect A than the main effect of B. The basic design AL is frequently observed in the selected efficient designs because a highly efficient estimate of the main effect A can be obtained by the AL design. The D_{12b} design is the AL/XL design. The most efficient design with six arrays D_{6b} is the XL design.

The last column of the Table 5.4 shows the overall efficiency values for the *Case b* if the selected efficient designs of the *Case a* would have been used. For this case, the overall efficiency for the *Case a* are calculated from the E–optimality values that are shown in Table 5.3, without those that correspond to the simple effects of B. In this

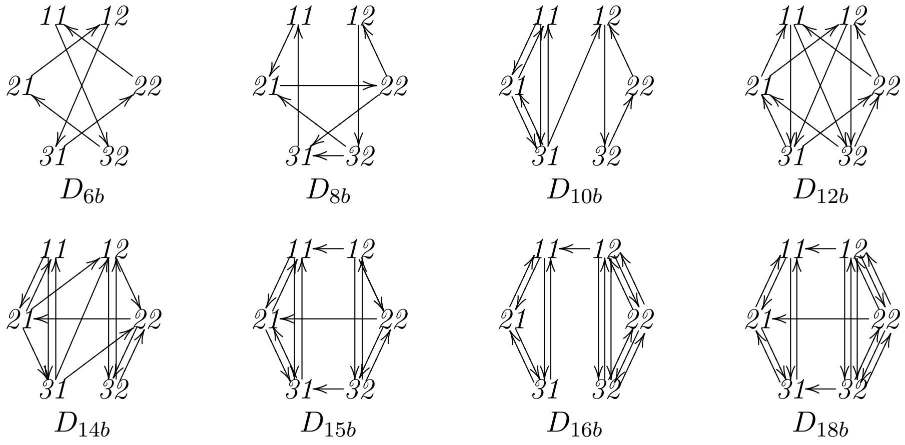


Figure 5.2: Graphical representations of the selected microarray designs for 3×2 experimental layout with respect to the effects of the *Case b*.

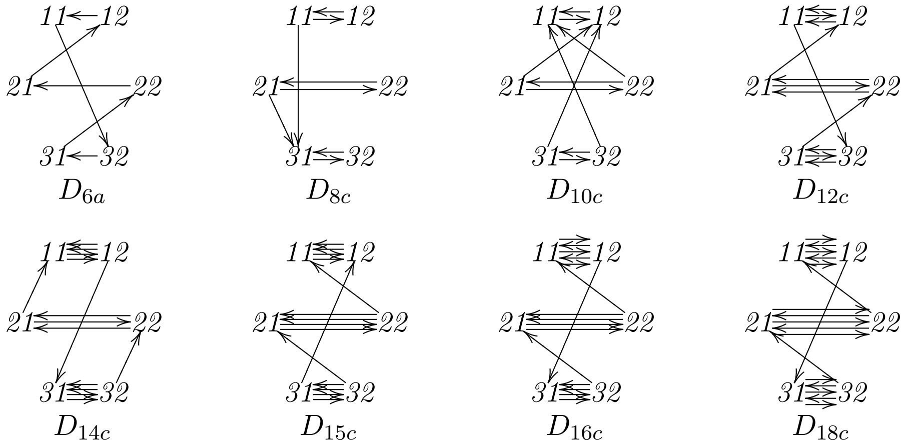
	Simple effects		Main effects		Int.	$\overline{\phi}$	
Design	A_{B1}	A_{B2}	A	B	$A \times B$	*Case b*	*Case a*
D_{6b}	3.000	3.000	6.000	4.000	2.000	3.600	3.280
D_{8b}	6.000	6.000	6.000	1.680	6.000	5.136	5.136
D_{10b}	12.000	6.000	8.000	0.462	8.000	6.892	6.639
D_{12b}	9.600	9.600	12.000	4.000	8.000	8.640	8.160
D_{14b}	12.000	12.000	12.000	2.053	12.000	10.011	8.672
D_{15b}	13.714	13.714	12.000	1.600	16.000	11.406	10.720
D_{16b}	12.000	18.000	14.400	0.533	14.400	11.867	10.869
D_{18b}	13.800	19.714	14.526	1.667	18.400	13.621	12.688

Table 5.4: The E–optimality and overall efficiency values corresponding to the effects of the *Case b* for the selected microarray designs for 3×2 experimental layout.

case, the designs for the *Case b* are more efficient than that of *Case a* except for the designs with eight arrays. Though the designs D_{8a} and D_{8b} are different in terms of the types of the arrays, but can be used to estimate the individual effects for the *Case b* with equal efficiency. The number of times each treatment needs to be hybridized on the arrays is different for these two designs. For the design D_{8a} (D_{8b}), all treatments are hybridized three times except for the treatment pairs *21* and *22* (*11*, *12*).

5.2.3 Efficient Designs for the *Case c*

In this case, the simple effects of B, both the main effects, and the interaction are considered as the effects of interest. Figure 5.3 shows the graphical representations of the selected efficient designs for this case. The corresponding E–optimality and overall efficiency values are illustrated in Table 5.5. The basic design *BS* is frequently observed in the selected designs because more effects corresponding to the factor B are of interest than to the factor A. The designs D_{12c} and D_{18c} can also be written as BS/D_{6a} and $2BS/D_{6a}$, respectively. The D_{15c} and D_{16c} designs can be obtained by adding suitable arrays to the D_{12c} design. So, the most important basic design for the *Case c* are D_{6a} and *BS*.



D_{6a} D_{8c} D_{10c} D_{12c}

D_{14c} D_{15c} D_{16c} D_{18c}

Figure 5.3: Graphical representations of the selected microarray designs for 3 × 2 experimental layout with respect to the effects of the *Case c*.

	Simple effects			Main effects		Int.	$\overline{\phi}$	
Design	B_{A1}	B_{A2}	B_{A3}	A	B	$A \times B$	*Case c*	*Case a*
D_{6c}	2.400	2.400	2.400	2.400	4.000	4.000	2.933	2.933
D_{8c}	4.000	4.000	4.000	0.889	4.000	8.000	4.148	3.616
D_{10c}	5.600	4.828	4.828	2.000	6.462	8.400	5.353	4.944
D_{12c}	6.545	6.545	6.545	2.769	8.000	12.000	7.068	6.800
D_{14c}	8.432	6.568	8.432	2.786	7.118	16.000	8.223	7.840
D_{15c}	8.558	8.558	8.558	2.344	9.946	16.000	8.994	7.860
D_{16c}	10.569	8.471	8.471	2.824	10.378	16.000	9.452	8.348
D_{18c}	10.543	10.543	10.543	2.857	12.000	20.000	11.104	10.000

Table 5.5: The E–optimality and overall efficiency values corresponding to the effects of the *Case c* for the selected microarray designs for 3 × 2 experimental layout.

All of these designs are found to be more efficient than the efficient designs for the *Case a* if those would have been used for this case.

5.2.4 Efficient Designs for the *Case d*

In this case only the main effects of the two factors and the interaction are considered as the effects of interest. Figure 5.4 shows the selected efficient microarray designs for this case. The corresponding E–optimality and overall efficiency values are illustrated in Table 5.6. In this case, the design D_{6a} is selected as the most efficient design with six arrays. The D_{8b} design can be obtained by using two suitable arrays to the design XL. The designs D_{12d}, D_{15d}, and D_{18d} contain the basic AL design. The D_{15d} design can be obtained by adding three suitable arrays to the design AL/D_{6a}. Besides the design D_{15d}, the D_{14d} designs also contains the D_{6a} design. The basic BS design can be found

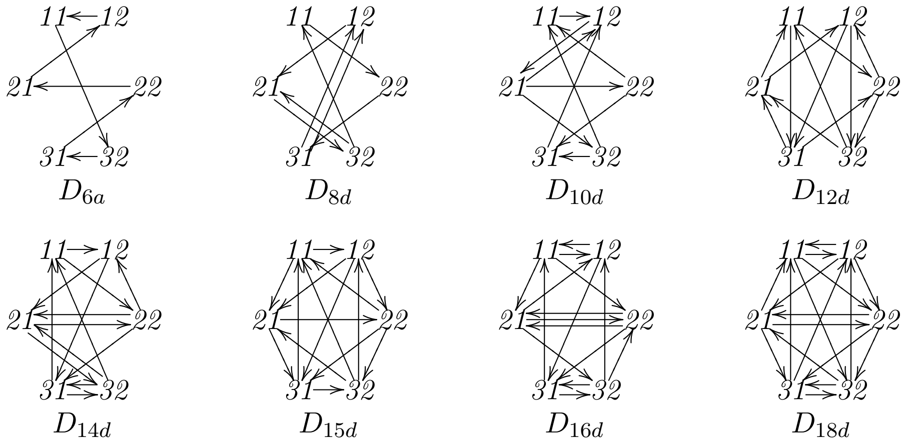


Figure 5.4: Graphical representations of the selected microarray designs for 3×2 experimental layout with respect to the effects of the *Case d*.

Design	Main effects A	B	Int. $A \times B$	$\overline{\phi}$ *Case d*	*Case a*	*Case b*	*Case c*
D_{6a}	2.400	4.000	4.000	3.467	3.467	4.000	3.467
D_{8d}	6.667	5.000	2.222	4.630	4.560	4.560	4.296
D_{10d}	6.000	6.383	6.000	6.128	6.064	5.487	5.621
D_{12d}	12.000	4.000	8.000	8.000	8.000	8.000	7.590
D_{14d}	8.727	7.332	10.309	8.790	8.739	8.684	8.635
D_{15d}	12.000	5.600	12.000	9.867	9.867	9.867	9.430
D_{16d}	8.793	7.604	14.167	10.188	10.115	9.778	9.734
D_{18d}	12.000	8.000	16.000	12.000	12.000	11.531	11.424

Table 5.6: The E–optimality and overall efficiency values corresponding to the effects of the *Case d* for the selected microarray designs for 3×2 experimental layout.

in the designs D_{16d} and D_{18d}.

The last three columns of Table 5.6 show the performance of the designs of the *Case* a, b, and c if these would have been used for the *Case d*. For the designs with six arrays, the D_{6b} design is found to be more efficient than the D_{6a} design. For 12 arrays, the D_{12a}, D_{12b}, and D_{12d} designs are found to be equally efficient for estimating the main effects and interaction if all these effects are of equal interest. However, the design D_{12a} is preferable to the other designs if the interaction is more important than the main effects and the designs D_{12d} and D_{12b} are preferable to the design D_{12a} if the main effect of A is more important than the other two effects.

5.3 Use of the Robustness Criteria in a Search for Good Designs

In the previous section, the efficient designs for the 3×2 experimental layout are reported for different numbers of arrays. Depending on the combination of the effects of interest, four different cases have been considered. In this section, we will examine whether the robustness criteria can improve a search for the good designs. That means, we are interested in the designs which are not only efficient but also robust with respect to missing observations.

First, consider an example with the designs for a 3×2 experimental layout, each of the designs has eight arrays. Assume the situation similar to the *Case a* where the simple effects, main effects, and interaction are considered as the effects of equal interest. By using a search with respect to (4.1) where only the efficiency criterion is used as the fitness function, two equally efficient designs D_{8a} and $D_{8a'}$ are found, where the overall efficiency, $\bar{\phi} = 4.212$. The graphical representations of the designs D_{8a} and $D_{8a'}$ are shown in Figure 5.5. More specifically, these two designs can be used to estimate all the

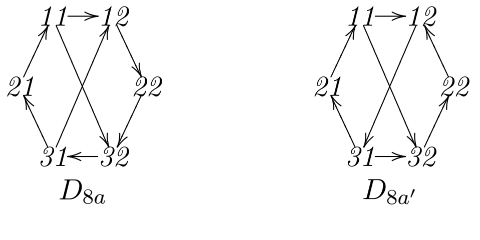


Figure 5.5: Graphical representations of two designs with eight arrays which are equally efficient for the *Case a*, but the design D_{8a} is more robust than the design $D_{8a'}$.

individual effects with equal efficiency. That means, the efficiency criterion cannot be used to find a winner from these two designs. Moreover, the number of hybridizations for each treatment is the same for both the designs, i.e., there is no difference between the designs in terms of the amount of *mRNA*s required for the hybridizations. In such a situation, average efficiency, a robustness criterion, can be used to find the best design. The problem of finding good designs in terms of both the efficiency and robustness criteria is defined in (4.2).

Both the designs break down if the there is more than one missing observation in the data, i.e., *BDN*=2 for both the designs. Eight residual designs can be constructed from each of these two designs with one missing array. Table 5.7 shows the overall efficiency values for each of the residual designs. For example, the overall efficiency for the design D_{8a} ($D_{8a'}$) reduces from 4.212 to 2.313 (3.091) if the array 32 → 31 (31 → 32) is missing. On the basis of the average efficiency with one missing array $\bar{\phi}_1$, the design D_{8a} is found to be more robust and efficient compared to the design $D_{8a'}$. With one missing array, there is a 70% chance that the D_{8a} design will be at least as efficient as the $D_{8a'}$ design.

	Missing array to pair of treatments								
Design	$(11,12)$	$(12,22)$	$(22,32)$	$(32,31)$	$(31,21)$	$(21,11)$	$(11,32)$	$(12,31)$	$\bar{\phi}_1$
D_{8a}	3.091	3.208	2.408	2.313	2.408	3.208	2.313	2.051	2.625
$D_{8a'}$	3.091	2.231	2.408	3.091	2.313	2.408	3.208	1.861	2.589

Table 5.7: The overall and average efficiency values of the residual designs corresponding to the designs D_{8a} and $D_{8a'}$ with one missing array.

The above example shows that robustness considerations can play a vital role in a search for good microarray designs. It is already mentioned that the average efficiency, which is a robustness criterion, can be used to define the fitness function along with the overall efficiency (see equation (4.2)). In this section, fitness of a design is defined as the sum of the overall efficiency and average efficiency with one or two missing arrays. One can consider more than two missing arrays for estimating the average efficiency which will contribute more to the fitness, but require more computational time. The designs that are obtained by optimizing this new fitness function are reported in the following sections only for the *Case a* and *d* with eight, 10, and 12 arrays.

5.3.1 Robust and Efficient Designs for the *Case a*

For this case, the simple effects, the main effects, and the interaction are considered as the effects of interest. For the designs with eight arrays, the D_{8a} and D_{8b} designs are found to be equally efficient. Both the designs break down if more than one observation is missing. The average efficiency values with one missing array are also similar for both the designs. For the designs with 10 arrays, the design D_{10a} is found to be the most robust and efficient design, and the corresponding breakdown number is three. The design D_{12a} is found to be the best design among the designs with 12 arrays and the corresponding breakdown number is four.

5.3.2 Robust and Efficient Designs for the *Case d*

The *Case d* is defined for the situation where only the main effects and interaction are considered as the effects of interest. Table 5.8 shows the results of this analysis.

		Effects					
Design	m	A	B	$A \times B$	BDN	$\overline{\phi}_m$	$\sum_m \overline{\phi}_m$
D_{8a}	0	6.000	1.680	6.000	2	4.560	4.560
	1	3.391	1.146	3.158		2.565	7.125
D_{8d}	0	6.667	5.000	2.222	2	4.630	4.630
	1	3.222	2.769	1.075		2.355	6.985
$D_{10d'}$	0	6.000	6.462	5.747	3	6.070	6.070
	1	4.596	5.131	4.051		4.593	10.663
D_{10d}	0	6.000	6.383	6.000	2	6.128	6.128
	1	4.127	5.114	4.031		4.424	10.552
D_{12a}	0	6.000	4.000	14.000	4	8.000	8.000
	1	4.708	3.541	10.508		6.252	14.252
	2	3.657	3.066	8.565		5.096	19.348
D_{12b}	0	12.000	4.000	8.000	4	8.000	8.000
	1	9.000	3.500	6.095		6.198	14.198
	2	7.323	2.980	4.831		5.045	19.243
D_{12d}	0	12.000	4.000	8.000	4	8.000	8.000
	1	9.000	3.500	6.095		6.198	14.198
	2	7.194	2.989	4.873		5.001	19.199

Table 5.8: Analysis of robust designs for 3 × 2 experimental layout with eight, 10, and 12 arrays when the main effects and interaction are of interest.

- For the designs with eight arrays, the design D_{8d} is found to be more efficient than the D_{8a} design when there is no missing observation. The minimum of the breakdown numbers is two for both the designs. If the designs are compared with respect to the average efficiency with one missing array along with the overall efficiency, the design D_{8a} is found to be preferable to the design D_{8d}.

- For the designs with 10 arrays, the design D_{10d} is the most efficient design with respect to the efficiency criterion. However, the design $D_{10d'}$ is found to be more robust compared to the D_{10d} design with respect to the breakdown numbers. If we consider both the robustness and efficiency criterion in the search, the design $D_{10d'}$ performs slightly better than the design D_{10d}. Figure 5.6 shows the graphical representation of the designs D_{10d} and $D_{10d'}$.

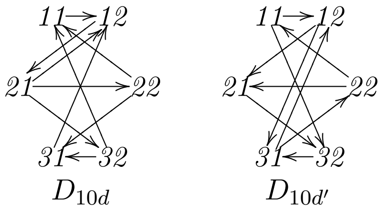


Figure 5.6: Graphical representations of two designs with 10 arrays. For the *Case d*, the design D_{10d} is more efficient than the design $D_{10d'}$, but the latter one is found to be more robust.

- For the designs with 12 arrays, the designs D_{12a}, D_{12b}, and D_{12d} are equally efficient for estimating the effects of the *Case d*. Moreover, all these designs are equally robust in terms of the minimum of the breakdown numbers. But with one missing array, the design D_{12a} is found to be more efficient than the other two designs. That means, the design D_{12a} is more robust and efficient compared to the competing designs. The design D_{12b} performs better than the design D_{12d} if two missing arrays are considered in the selection process.

5.4 Comparisons of the *GA* Operators

In this section, performances of different types of the *GA* operators are compared by using simulation studies. For each comparison, $S = 500$ simulations are considered and the value of the overall efficiency corresponding to the most efficient design is recorded for each of the simulations $s = 1, 2, \ldots, S$. Let ξ_s be the most efficient design at the simulation s and $\overline{\phi}(\xi_s, \mathbf{C}'\boldsymbol{\beta})$ be the corresponding value of the overall efficiency, where $\mathbf{C}'\boldsymbol{\beta}$ is the effect of interest. The performances of the different *GA* operators can compared by the distributions of $\overline{\phi}(\xi_s, \mathbf{C}'\boldsymbol{\beta})$. To compare the performance of the design ξ_s with respect to a known design, the average of the relative efficiency can be used. In general, for the set of designs $\{\xi_s, s = 1, 2, \ldots, S\}$, the average of the relative efficiency with respect to the design ξ_0 can be expressed as

$$ARelE(\xi_0) = \frac{(1/S)\sum_{s=1}^{S} \overline{\phi}(\xi_s, \mathbf{C}'\boldsymbol{\beta})}{\overline{\phi}(\xi_0, \mathbf{C}'\boldsymbol{\beta})} \times 100, \tag{5.1}$$

where $\overline{\phi}(\xi_0, \mathbf{C}'\boldsymbol{\beta})$ is the overall efficiency of the design ξ_0 with the effect $\mathbf{C}'\boldsymbol{\beta}$ as the effect of interest. The $ARelE(\xi_0)$ indicates, on an average, how efficient the simulated designs with respect to the known ξ_0 design. The minimum of the relative efficiency with respect to the design ξ_0,

$$MRelE(\xi_0) = \frac{\min_s\{\overline{\phi}(\xi_s, \mathbf{C}'\boldsymbol{\beta}), s = 1, 2, \ldots, S\}}{\overline{\phi}(\xi_0, \mathbf{C}'\boldsymbol{\beta})} \times 100, \tag{5.2}$$

can be used to compare the efficiency of the worst design of the simulated designs with respect to the best design ξ_0.

For each comparison, the *GA* is applied to find efficient designs with 12 arrays for the 3 × 2 experimental layout and the effects of the *Case d* are considered as the effects of interest. The most efficient design for this setup is the design D_{12d} (see Table 5.6 and Figure 5.4 for details).

5.4.1 Selection Operator

In the *GA*, the selection operator is used to select the parent population from the current population. Two types of selection operators are considered in our implementation of the *GA*, which are: sampling proportional to fitness (*SPF*) and remainder stochastic sampling (*RSS*) (see §4.3.1 for details). For each of the selection operators, the *GA* is run 500 times for selecting efficient designs and the summary of the results are reported in Table 5.9. It shows that the average of the overall efficiency is slightly larger if the *RSS* is used as the selection operator compared to the *SPF*. The values of the average relative efficiency with respect to the best design D_{12d} show that the selection operator *RSS* performs better that the selection operator *SPF*. Another advantage of using the

Selection	Overall efficiency						Avg. no. of
operator	min	mean	median	max	sd	$ARelE(D_{12d})$	generations
SPF	7.004	7.657	7.603	8.000	0.231	95.71	1830
RSS	7.108	7.774	7.798	8.000	0.229	97.18	640

Table 5.9: A comparison of the performance of two selection operators *SPF* and *RSS* in selecting efficient microarray designs with 12 arrays. The one–point crossover operator is used with $p_c = 0.75$ and $p_m = 0.03$.

RSS over the *SPF* is that the former can find good designs with a smaller number of generations.

5.4.2 Crossover Operator

Three types of the crossover operator, namely, one–point, two–points, and uniform crossover are included in our implementation of the *GA*. For each operator, the *GA* is run 500 times and the summary of the results are reported in Table 5.10. There is no difference among these crossover operators in terms of the average of the overall efficiency, but the uniform crossover requires more generations to converge compared to the other two crossover operators.

Crossover	Overall efficiency						Avg. no. of
operator	min	mean	median	max	sd	$ARelE(D_{12d})$	generations
one–point	7.127	7.781	7.798	8.000	0.231	97.26	637
two–points	7.092	7.792	7.798	8.000	0.212	97.40	746
uniform	7.111	7.807	7.798	8.000	0.213	97.59	1109

Table 5.10: A comparison of three crossover operators in selecting efficient microarray designs with 12 arrays. The *RSS* as the selection operator and $p_m = 0.03$ are considered.

Selection of Crossover/Mutation Probability

To study the effect of crossover probability p_c on the performance of the search procedure, the *GA* is applied to find efficient designs with different values of p_c ranges from 0.50 to 0.90. For each value of p_c, 500 simulations are considered and the results of the simulation studies are reported in Table 5.11. It shows that the average of the overall efficiency increases with the crossover probability. For $p_c = 0.9$, the *GA* can find the designs that are very close to the best design D_{12d} in terms of the overall efficiency,

	Overall efficiency						Avg. no. of
p_c	min	mean	Median	max	sd	$ARelE(D_{12d})$	generations
0.5	6.982	7.491	7.481	8.000	0.267	93.64	140
0.6	6.969	7.592	7.595	8.000	0.272	94.90	194
0.7	7.037	7.746	7.512	8.000	0.257	96.09	340
0.8	7.014	7.842	8.000	8.000	0.198	98.07	884
0.9	7.261	7.923	8.000	8.000	0.137	99.04	>3000

Table 5.11: A comparison of *GA*'s performance in selecting efficient microarray designs for different values of the crossover probabilities. The *RSS* as the selection operator and $p_m = 0.03$ are considered.

but the algorithm requires longer computing time to converge for large values of the crossover operator.

The mutation operator brings variation to the *GA*. In practice, a very small value of the mutation probability, e.g., 0.001, is used in *GA*s. Different values of mutation probability are used in the analysis of the following section where the performance of the *GA* are examined in the context of previously known microarray designs.

5.5 Evaluation of the Performance of the *GA*

In this section, the performance of the *GA* in selecting efficient microarray designs are evaluated by a number of simulation studies. The performance of the *GA* is compared with respect to the efficient designs (as far we know) for both the one–way and two–factor factorial experiments. For the two–factor factorial experiment, designs that are reported in §5.2 are considered as the efficient designs and for the one–way factorial experiment, the efficient designs are selected from the literature.

For the two–factor factorial experiment, the performances of the *GA* are compared with respect to the designs D_{8d}, D_{12d}, and D_{15d} which are reported as the best design for the *Case d* with $n = 8, 12, 15$ (see §5.2). Different values of mutation probability ($p_m = 0.005, 0.01, 0.03$) are used in simulation studies to see its importance in the selection procedure. Results of the different simulation studies are shown in Table 5.12 which shows that the average overall efficiency increases as mutation probability increases for all the cases. As expected, the simulation study shows that the algorithm requires more

		Overall efficiency						Avg. no. of
n	p_m	min	median	mean	max	sd	$ARelE(D_{12d})$	generations
8	0.005	3.766	4.340	4.316	4.630	0.174	93.21	52
	0.010	3.845	4.343	4.354	4.630	0.164	94.04	75
	0.030	4.124	4.498	4.461	4.630	0.131	96.34	120
12	0.005	6.906	7.366	7.405	8.000	0.280	92.56	81
	0.010	6.954	7.481	7.487	8.000	0.273	93.58	125
	0.030	7.083	7.798	7.771	8.000	0.226	97.13	918
15	0.005	9.019	9.689	9.573	9.867	0.237	97.01	171
	0.010	9.117	9.740	9.646	9.867	0.201	97.76	286
	0.030	9.112	9.740	9.682	9.867	0.149	98.12	1330

Table 5.12: Results of a simulation study for assessing the performance of *GA*s in selecting efficient microarray designs for 3 × 2 experimental layout. The mutation probabilities are varied over three different values, but only one crossover probability ($p_c = 0.75$) is considered for all cases.

generations to converge for large values of the mutation probability. The values of the average of the relative efficiency (*ARelE*), which is defined in (5.1), show that the *GA* can find highly efficient designs with respect to the efficient designs that are suggested for the two–factor factorial experiment with $n = 8, 12, 15$. The average number of generations required to converge the algorithm is found to be very small, especially for $n = 8, 12$, which indicates premature convergence of the algorithm.

5.5.1 Comparison with Known Efficient Designs

It was already mentioned in §4.1 that Kerr and Churchill (2001b) suggested A–optimal designs with K, $K+2$, and $2K$ arrays for $1 \times K$ factorial experiments when all pairwise comparisons are considered as the effects of equal interest. In this section, we consider simulation studies to compare the designs that are found by *GA* and the designs that are reported by Kerr and Churchill (2001b) in terms of the overall efficiency criterion. As an example, we consider only $K = 6, 7, 8$. For each case, 500 simulations are considered and the minimum, mean, median, maximum, and standard deviation of the overall efficiency values are reported.

The results of the simulation study are shown in Table 5.13. The small values of the standard deviation indicate that the performance of the *GA* is consistent over different simulations. The values of the *ARelE* show that the *GA* algorithm can find designs with high efficiency with respect to the designs that are suggested by Kerr and Churchill (2001b) with $K = 6, 7, 8$. For this case, the values of the minimum of the relative efficiency (*MRelE*), which is defined in (5.2), are ranging from 90 percent to 99 percent,

		Overall efficiency						Avg. no. of
Layout	n	min	median	mean	max	sd	*ARelE*	generations
1×6	6	1.643	1.827	1.810	1.827	0.034	99.06	73
	8	2.681	2.725	2.743	2.773	0.024	98.91	114
	12	4.549	4.640	4.619	4.640	0.039	99.54	116
1×7	7	1.530	1.594	1.612	1.633	0.022	98.71	109
	9	2.419	2.433	2.435	2.444	0.009	99.65	127
	14	4.276	4.377	4.374	4.399	0.024	99.44	202
1×8	8	1.402	1.464	1.464	1.482	0.013	98.77	133
	10	2.139	2.211	2.204	2.213	0.012	99.60	152
	16	4.096	4.223	4.211	4.327	0.038	97.31	290

Table 5.13: Results of the simulation studies for selecting efficient designs from different one–way experimental layouts. All pairwise comparisons are considered as the effects of interest. For all the simulations $p_c = 0.75$ and $p_m = 0.04$ are considered.

which show that even the worst design of the 500 simulations is highly efficient with respect to the designs suggested by Kerr and Churchill (2001b). (For a specific n, the value of the *MRelE* can easily be obtained by the ratio of the minimum and maximum of the overall efficiency criterion values, i.e., by the ratio of the values at the columns 3 (min) and 6 (max) of Table 5.13).

5.6 Conclusion

For a fixed number of available arrays and the given treatment combinations or experimental layout, a number of designs can be considered for microarray experiments. The choice of the design depends, among other things, on its ability to estimate the effects of interest with high efficiency. There have not been many attempts for selecting good microarray designs by systematically examining the efficiency of the competing designs. A naive search to the complete set of possible designs could be infeasible in most of the cases because of resource constraints. The problem of selecting good microarray designs can be expressed as an optimization problem with a discrete search space, e.g., a combinatorial optimization problem. The usual calculus–based optimization methods cannot be used to find the optimal solution from a discrete search space because the corresponding objective function is not differentiable.

In this chapter, an application of genetic algorithms for selecting good microarray designs have been discussed. A genetic algorithm is a stochastic search technique which mimics some common features of the natural evolution such as, selection, inheritance, variability, etc. to find near–optimal, if not optimal, solutions of the problem under investigation. It has been widely used in a broad range of optimization problems that includes combinatorial optimization problems. In the context of microarray experiments, a brief description of the genetic algorithms is given in §4.

As an example, the 3×2 experimental layout is chosen for which efficient designs are reported for different numbers of arrays. Four different combinations of the simple effects, main effects and interaction are considered as the effects of interest. It has been shown that all the efficient designs are not necessarily robust. Inclusion of robustness considerations in the search process requires more computing time, robust designs are reported only for eight, 10, and 12 arrays. The reported designs are more efficient than the corresponding basic and composite designs, suggested by Landgrebe et al. (2004).

The analysis shows that the selection of the efficient designs depends on the specific combinations of the effects under investigation and also on the number of arrays the experimenter wants to conduct the experiment with. That means, a composite design, which is composed of the two replications of an efficient design of size n, may not necessarily be the most efficient design of size $2n$. For example, D_{6a} is the best design of size six for the *Case* a, but the $2D_{6a}$ design is not the best design with 12 arrays. We also reported that for a fixed number of arrays, different efficient designs can be found for different sets of research questions, e.g., efficient designs for the *Case* a may not be the efficient anymore if the effects of the *Case* d, i.e., the main effects and interaction, are considered as the effects of interest.

The performance of the *GA* depends on the selection of its parameters, e.g., values of the mutation or crossover probability, types of crossover and selection operators, etc. A number of simulations have been performed to get an idea about the suitable parameter values. In the context of microarray experiments, the reminder stochastic sampling outperforms the other selection operators in terms of finding good designs and faster convergence. A relatively high value of the mutation probability ($0.03 \leq p_m \leq 0.05$) seems to perform better than the usual smaller values. In terms of the efficiency of the design, there is no difference between the crossover operators, namely, one–point, two–points, and uniform crossover, but one–point crossover is faster compared to the other crossover operators, i.e., it requires less number of generations to converge. The related simulation study shows that *GA*s work well if the crossover probability is large, e.g., $p_c > 0.70$.

Since a genetic algorithm is a stochastic search process, the same near–optimal design may not be obtained from different runs. To examine the performance of the *GA* in selecting efficient microarray designs consistently, a number of simulation studies are considered. Besides the distribution of the overall efficiency of the designs that are selected by the *GA*, the averages of the relative efficiency of the selected designs with respect to some known efficient designs are also reported for all the simulation studies. For one–way factorial experiments, Kerr and Churchill (2001b) reported efficient designs for a small number of treatment combinations. We applied the *GA* to some of these cases and found that the algorithm performs fairly well.

An advantage of using *GA*s in selecting good microarray designs is that it provides a population of near optimal and robust designs not only a single optimal design. If a set of good designs is available, researchers can consider other constraints of the experiment, e.g., amount of the *RNA* available for different treatments, in selecting good design. A common practice of a *GA* is to run the algorithm for a large number of generations. The current implementation of the *GA* is not fast enough to run it for a large number of generations (e.g., $\geq 10,000$) in a reasonable time. Using *C* codes instead of **R** (R Development Core Team, 2004) would make the algorithm faster and more efficient.

Chapter 6

Conclusion

This dissertation deals with the selection of good microarray designs for one–way and multi–factor factorial experiments. For a given experimental layout, a number of microarray designs can be considered for the available number of arrays. Efficiency criteria are used to select efficient designs and the selection depends on the effects of interest. The benefits of using efficient designs instead of inefficient ones are illustrated by a simulation study. To assess the quality of a design in estimating the effects of interest in the presence of missing observations, three robustness criteria, namely, the breakdown number, average efficiency, and proportions of the effective designs, are proposed. The main objective of this dissertation is to demonstrate the use of the efficiency and robustness criteria in the selection for good microarray designs.

One of the objectives of this study is to compare the common reference design with the loop designs in terms of the efficiency and robustness criteria. For both the one–way and multi–factor factorial experiment, the carefully designed loop designs are found to be more efficient compared to the common reference design. The loop designs are found to be more robust than the corresponding common reference design for the multi–factor factorial experiment. For one–way factorial experiments, the loop designs are more robust than the common reference design if the number of arrays is larger than the number of the treatments to be compared. Reverse dye labelling can improve the robustness of the loop designs, but not of the common reference designs and it does not affect the efficiency criterion.

The problem of selecting good microarray designs is defined as an optimization problem with a discrete search space. The commonly used calculus–based optimization methods cannot be used for such a case because the corresponding objective function is not differentiable. In this dissertation, a genetic algorithm based search procedure is proposed which can be used to select good designs for different numbers of available arrays. This algorithm is general enough to find good designs for both the one–way and multi–factor factorial experiments.

Genetic algorithm is a stochastic search technique that mimics some common features, e.g., selection, inheritance, variability, etc., of natural evolution to find near–optimal, if not optimal, solutions of the problem under investigation. In genetic algorithm, the problem under investigation is encoded into a chromosome–like data structure where each chromosome represents a search point in the space of the potential solutions of the problem. The encoding procedure and the different important operators are defined in the context of microarray experiments. Genetic algorithm deals with a population of chromosomes that evolves over generations for the improvement of the quality of the population.

This algorithm can be used to find not only the efficient microarray designs, but also the robust ones. The average efficiency, a robustness criterion, with a specific number of missing arrays can be used to incorporate robustness considerations into the search process. If more than one question is of interest, the importance of the individual question can be specified, e.g., researchers may want to find the designs which could be used to estimate interaction more efficiently than the main effects. Genetic algorithm can provide a population of the near–optimal, if not optimal, and robust designs, not only a single design.

As an example, the 3×2 experimental layout is chosen to apply the genetic algorithm for different numbers of available arrays. The efficient designs are reported for four different combinations of the simple effects, the main effects, and interaction as the effects of interest. The efficient designs could be different for different combinations of the effects. Replications of an efficient design with a small number of arrays may not be the efficient design, i.e., for a given number of the available arrays, a new run of the genetic algorithm is required to find good designs. It has been shown empirically that the efficient designs are not necessarily the robust ones. The procedure of finding designs which are not only efficient, but also robust is described in §5.

The performance of the genetic algorithm in the context of microarray designs is assessed by simulation studies. The most efficient designs for one–way experimental layout are reported by Kerr and Churchill (2001b) for a small number of treatments. The simulation studies show that the genetic algorithm can find the designs which are highly efficient with respect to the designs that are reported by Kerr and Churchill (2001b). The performance of the genetic algorithm depends on its parameter values. To have a good idea about the parameters of the genetic algorithm, a number of simulation studies are performed. It shows that as a selection operator, the remainder stochastic sampling approach outperforms the sampling proportion to fitness approach in terms of the quality and time. Large value of the crossover probability, e.g., between 0.75 to 0.90, is found to be useful for microarray designs. A relatively large value of the mutation probability, e.g., between 0.01 to 0.03, instead of commonly suggested 0.001 performs well for its application to the microarray design problem.

6.1 Future Research

The major points of the future research related to this work are given in the following.

- The sources of variation in a microarray experiment can be partitioned into three categories, namely, biological variation, technical variation, and measurement error. The methods that are described in this dissertation are based on the global *ANOVA* model (2.5) which considers only the technical variation and measurement error as the sources of variation. One of the future works would be to incorporate biological variation in the search of good microarray design.

- The computing language **R** (R Development Core Team, 2004) is used to implement the genetic algorithm for our package *robustMAdesigns*. Implementing the package in faster computer language, e.g., *C*, *C++*, etc. will improve the performance of the genetic algorithm in searching good microarray design.

Appendix A

Descriptions of the Functions of `robustMAdesigns` Package

`contMatrix` *Generating a Contrast Matrix or Vector*

Description

Generates a contrast matrix/vector corresponding to a specific effect.

Usage

```
contMatrix(layout, effect, commRef=FALSE)
```

Arguments

`layout` experimental layout of interest.

`effect` treatment effect for which the contrast matrix/vector is required.

`commRef` indicates whether the associated design is common reference.

Details

One–way and two–factor factorial experiments can be specified as `layout`, e.g., `"1x2"` can be used for an one–way experimental layout with a factor that has two levels, `"3x2"` is for a two–factor factorial experiment with the factors that have three and two levels, respectively, etc.

For `effect`, pairwise comparisons (`"all-pair"`) and global (`"global"`) effects are available for one–way layouts and simple effects (`"simA"`, `"simB"`), main effects (`"mainA"`, `"mainB"`), and interaction (`"AxB"`) are available for two–factor factorial experiments.

Value

Contrast matrix (vector) is returned and its first two columns (elements) correspond to the dye effects.

Author(s)

A. H. M. Mahbub-ul Latif <mlatif@univdhaka.edu>

See Also

`desMatrix`, `eCriteria`, `rCriteria`

Examples

```
# For a 1x4 layout, contrast matrix corresponds to all pairwise comparisons
contMatrix(layout="1x4", effect="all-pair")
# For a 4x3 layout, contrast matrix corresponding to interaction
contMatrix(layout="4x3", effect="AxB")
```

`contrastEst` *Estimate and test a contrast matrix/vector*

Description

Estimate and test a contrast matrix/vector

Usage

```
contrastEst(z, dmat, cmat, alternative="two.sided")
```

Arguments

`z`	vector of responses.
`dmat`	design matrix of interest.
`cmat`	contrast matrix corresponding to the effect of interest.
`alternative`	specification of alternative hypothesis.

Value

Returns a vector of test statistics, estimate of the contrast matrix/vector, p–value, mean squared error, etc. Each row of the design matrix corresponds to a response.

Author(s)

A. H. M. Mahbub-ul Latif <mlatif@univdhaka.edu>

References

Searle, S. R. (1971) *Linear Models.* Wiley.

See Also

`desMatrix`, `contMatrix`, `estimable`

Examples

```
# design matrix
dmat <- desMatrix("1x3", design="CL")
dmat <- rbind(dmat, dmat)
# contraxt vector for comparing first and second factor level
con <- c(0,0,1,-1,0)
# response
x = rnorm(6)
#
contrastEst(z=x, dmat=dmat, cmat=con)
```

`desMatrix` *Generating a Design Matrix*

Description

Generates a design matrix corresponding to an experimental layout and a design.

Usage

```
desMatrix(layout, design="DS")
```

Arguments

`layout`	experimental layout of interest.
`design`	design from the layout of interest.

Details

For details of `layout` see `contMatrix`

Design matrices corresponding to the common reference (`"CR"`), circular loop (`"CL"`), and dye–swap (`"DS"`) designs are available for any experimental layout.

For 2 × 2 layout, the A–swap (`"AS"`), B–swap (`"BS"`), and cross–swap (`"XS"`) designs can also be used for `design`.

For 3 × 2 layout, the A–loop (`"AL"`), `"BS"`, triangular–loop (`"TL"`), cross–loop (`"XL"`), and star–swap (`"RS"`) designs can be used for `design`.

Value

A data frame of the design matrix of which first two columns correspond to the dye effects.

Author(s)

A. H. M. Mahbub-ul Latif <mlatif@univdhaka.edu>

See Also

`contMatrix`, `rCriteria`, `eCriteria`

Examples

```
# design matrix for dye-swap design from 1x3 layout
desMatrix(layout="1x3", design="DS")
# design matrix for circular loop design from 2x2 layout
desMatrix(layout="2x2", design="CL")
```

`eCriteria` *Efficiency Criteria*

Description

Computes efficiency criteria for a given design and the contrast matrix/vector.

Usage

```
eCriteria(dmat, cmat, cinfo=NULL, m=0, type="e", cname=NULL)
```

Arguments

`dmat` design matrix of interest.

`cmat` contrast matrix corresponding to the effect of interest.

`cinfo` a vector, represents the number of rows corresponding to the contrast matrices of interest.

`m` number of missing values to be considered for computing efficiency criteria.

`type` types of efficiency criteria.

`cname` names of the contrasts.

Details

For a specific effect, this function can compute the efficiency of a design.

Different efficiency criteria, namely, e–, a–, d–, and t–optimality can be specified in `type`.

If more than one effect is of interest, `cmat` is constructed by `rbind`–ing the corresponding contrast matrices and the vector `cinfo` specifies the number of rows of each of the contrast matrices.

Value

A `list` of efficiency criteria (`eff`) and corresponding set of missing arrays (`m`).

Author(s)

A. H. M. Mahbub-ul Latif <mlatif@univdhaka.edu>

References

Pukelsheim, F. (1993) *Optimal designs of experiments.* Wiley.

Landgrebe, J., Bretz, F., Brunner, E. (2005) Efficient design and analysis of two–color factorial microarray design. *Computational Statistics & Data Analysis (in press).*

See Also

`desMatrix`, `contMatrix`, `rCriteria`

Examples

```
# design matrix corresponding to a circular loop design from 3x2 layout
x <- desMatrix(layout = "3x2", design="CL")
# contrast matrices for the main effects of two factors
ccA <- contMatrix(layout="3x2", effect="mainA")
ccB <- contMatrix(layout="3x2", effect="mainB")
# combining the contrast matrices
cc = rbind(ccA, ccB)
# corresponding value of cinfo
cin = c(nrow(ccA), nrow(ccB))
# e-optimality without any missing value
eCriteria(dmat=x, cmat=cc, cinfo=cin, m=0)
# e-efficiency with at most two missing values
eCriteria(dmat=x, cmat=cc, cinfo=cin, m=2)
```

`estimable` *Estimability of an effect*

Description

Check the estimability of an effect.

Usage

```
estimable(dmat, cmat)
```

Arguments

`dmat`	design matrix of interest.
`cmat`	contrast matrix corresponding to the effect of interest.

Value

A logical variable indicating whether the given contrast is estimable.

Author(s)

A. H. M. Mahbub-ul Latif <mlatif@univdhaka.edu>

References

Searle, S. R. (1971) *Linear Models.* Wiley.

See Also

`desMatrix`, `contMatrix`

Examples

```
# design matrix
dmat <- desMatrix("1x3", design="CR")
# contraxt vector for comparing first and second factor level
con <- c(0,0,1,-1,0,0)
# checking estimability
estimable(dmat, con)
```

GA *Genetic Algorithm for Microarray Designs*

Description

Genetic Algorithms for selecting near–optimal microarray designs.

Usage

```
GA(dmat=NULL, layout=NULL, n, cmat, cinfo=NULL, Pcross=.75,
  crossType="one.point", Pmut=NULL, selectType="RSS", verbose=FALSE,
    wt=NULL, popSize=50,  maxGen=2000, burnOut=100, convergeNum=10,
       nIter=1, m=0, keep.elite=FALSE, scaling=TRUE, cname=NULL)
```

Arguments

`n`	number of available arrays.
`Pcross`	crossover probability.
`crossType`	crossover type, currently available options are: `"one.point"`, `"two.points"`, and `"uniform"`.
`Pmut`	mutation probability.
`selectType`	selection type, currently available options are: sampling proportional to fitness (`"SPF"`) and reminder stochastic sampling (`"RSS"`).
`verbose`	controls the print out during iterations.
`wt`	a vector of weights to indicate individual importance of effects of interest.
`popSize`	population size of a generation.
`maxGen`	maximum number of generations algorithm will run unless the convergence criteria is met.
`burnOut`	number of generations for burning out.
`convergeNum`	convergence criteria.
`nIter`	number of iterations.
`m`	number of missing values to be considered for computing fitness function.

`keep.elite` preserving the best design of each generation.

`scaling` linear scaling on fitness.

`cname` contrast names.

see `rCriteria` for `cmat`, `dmat`, `cinfo` and `contMatrix` for `layout`.

Details

This function can find near–optimal microarray designs for different number of arrays. Instead of specifying experimental layout, the design matrix corresponding to the list of arrays from which near–optimal designs will be searched can also be used.

Value

`GA` returns a list of objects including:

`design` a matrix of order `popSize`×`n` where each row represents a design of the population.

`opt` a matrix of the estimate of the efficiency criterion corresponding to the designs in `design`.

`opt.overall` estimates of the overall efficiency corresponding to each `design`.

`best` list of `design`, `opt`, and `opt.overall` corresponding to the best designs at each iteration.

`history` list of `design`, `opt`, and `opt.overall` corresponding to the best designs at each generation. This also contains the estimates of `median` and `mad` of the overall efficiencies at each generation.

`input` list of inputs, such as, `cmat`, `dmat`, `cinfo`.

`converge` convergence code, 0 indicates perfect convergence and 1 indicates either maximum number of generations reached or population become degenerate.

Author(s)

A. H. M. Mahbub-ul Latif <mlatif@univdhaka.edu>

References

Latif, A. H. M. Mahbub–ul (2005). Robustness and efficiency issues in complex statistical designs for two–color microarray experiments. Unpublished Ph.D. thesis. Georg–August–Universität Göttingen, Germany.

See Also

`desMatrix`, `contMatrix`, `rCriteria`, `eCriteria`

Examples

```
# near-optimal designs with 7 arrays for the layout 3x2 and main effects are of interest
#
# contrast matrix
# A = contMatrix(layout="3x2", effect="mainA")
# B = contMatrix(layout="3x2", effect="mainB")
# cmat = rbind(A, B)
# cinfo = c(nrow(A), nrow(B))
#
# run genetic algorithm
# res = GA(layout="3x2", n=7, cmat=cmat, cinfo=cinfo, verbose=T)
#
# to see the best designs
# print(res$best)
```

`rCriteria` *Robustness Criteria*

Description

Computes robustness criteria for a given design and contrast matrix/vector.

Usage

```
rCriteria(dmat, cmat, cinfo=NULL, type="e", cname=NULL)
```

Arguments

`dmat`	design matrix of interest.
`cmat`	contrast matrix corresponding to the effect of interest.
`cinfo`	a vector represents the number of rows corresponding to the contrast matrices of interest.
`type`	specifies the type of the efficiency criteria.
`cname`	name of the contrasts.

Details

Three different robustness criteria, namely, breakdown number (`bdn`), average efficiency (`avgEff`), and proportion of the effective designs (`pED`) are returned.

For details of `cmat` and `cinfo`, see `eCriteria`.

Value

A list of `bdn`, `avgEff`, and `pED`.

Author(s)

A. H. M. Mahbub-ul Latif <mlatif@univdhaka.edu>

References

Latif, A. H. M. Mahbub–ul (2005). Robustness and efficiency issues in complex statistical designs for two–color microarray experiments. Unpublished Ph.D. thesis. Georg–August–Universität Göttingen, Germany.

See Also

`eCriteria`, `desMatrix`, `contMatrix`

Examples

```
# robustness of a dye-swap design from 1x3 layout for comparing 1-2 and 1-3
con <- rbind(c(0,0,1,-1,0), c(0,0,1,0,-1))
# corresponding cinfo
cinfo <- c(1,1)
# design matrix
```

```
xx <- desMatrix(layout="1x3", design="DS")
# robustness criteria
rCriteria(dmat=xx, cmat=con, cinfo=cinfo)
```

Bibliography

Alwine, J., Kemp, D., and Stark, G. (1977). Method for detection of specific rnas in agarose gels by transfer to diazobenzyloxymethyl-paper and hybridization with dna probes. *Proc Natl Acad Sci U S A*, 74(12):5350–5354.

Bagchi, S. and Cheng, C.-S. (1993). Some optimal designs of block size two. *Journal of Statistical Planning and Inference*, 37:245–253.

Bechhofer, R. E. and Tamhane, A. C. (1981). Incomplete block designs for comparing treatments with a control: general theory. *Technometrics*, 23:45–57.

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

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(12):2022–2029.

Casella, G. and Berger, R. L. (1990). *Statistical Inference.* Duxbury Press, Belmont, CA, USA.

Charbonneau, P. (2002). An Introduction to Genetic Algorithms for Numerical Optimization. Technical Report NCAR/TN–450+IA, High Attitude Observatory National Center for Atmospheric Research, Boulder, Colorado, USA.

Cheng, C.-S. (1980). On the E–optimality of some block designs. *Journal of the Royal Statistical Society, series B*, 42(2):199–204.

Churchill, G. A. (2002). Fundamentals of experimental design for cDNA microarrays. *Nat Genet*, 32 Suppl:490–495.

Cochran, W. G. and Cox, M. G. (1992). *Experimental Designs, 2nd ed.* John Wiley & sons.

Cui, X., Kerr, M. K., and Churchill, G. A. (2003). Transformations for cDNA Microarray Data. *Statistical Applications in Genetics and Molecular Biology*, 2(1):Article 4.

Davis, L., editor (1991). *Handbook of genetic algorithms.* Van Nostrand Reinhold, New York.

De Jong, K. A. (1975). *"An Analysis of the behavior of a class of Genetic Adaptive Systems".* PhD thesis, University of Michigan, Ann Arbor, MI.

DeRisi, J., Penland, L., Brown, P., Bittner, M., Meltzer, P., Ray, M., Chen, Y., Su, Y., and Trent, J. (1996). Use of a cdna microarray to analyse gene expression patterns in human cancer. *Nat Genet*, 14(4):457–60.

Dey, A. (1993). Robustness of block designs against missing data. *Statistica Sinica*, 3:219–231.

Dobbin, K., Shih, J. H., and Simon, R. (2003a). Questions and answers on design of dual-label microarrays for identifying differentially expressed genes. *J Natl Cancer Inst*, 95(18):1362–1369.

Dobbin, K., Shih, J. H., and Simon, R. (2003b). Statistical design of reverse dye microarrays. *Bioinformatics*, 19(7):803–810. Evaluation Studies.

Dudoit, S., Fridlyand, J., and Speed, T. P. (2002a). Comparisons of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97:77–87.

Dudoit, S., Shaffer, J., and Boldrick, J. (2003). Multiple hypothesis testing in microarray experiments. *Statistical Sciences*, 18:89–112.

Dudoit, S., Yang, Y. H., Callow, M. J., and Speed, T. P. (2002b). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, 12:111–139.

Duggan, D. J., Bittner, M., Chen, Y., Meltzer, P., and Trent, J. M. (1999). Expression profiling using "cDNA" microarrays. *Nature Genetics*, 21(1 Suppl):10–14.

Eisen, M. and Brown, P. (1999). Dna arrays for analysis of gene expression. *Methods Enzymol*, 303:179–205.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, 95(25):14863–14868.

Glonek, G. F. V. and Solomon, P. J. (2004). Factorial and time course designs for cDNA microarray experiments. *Biostatistics*, 5(1):89–111.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison–Wesley, Reading, Massachusetts.

Greenhalgh, D. and Marshall, S. (2000). Convergence criteria for genetic algorithms. *SIAM Journal on Computing*, 30:269–282.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* The university of Michigan Press, Ann Arbor.

Hoyle, D., Rattray, M., Jupp, R., and Brass, A. (2002). Making sense of microarray data distributions. *Bioinformatics*, 18(4):576–584.

Huber, W., von Heydebreck, A., Sultmann, H., Poustka, A., and Vingron, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18 Suppl 1:96–104. Evaluation Studies.

Jackson, B. W. and Thoro, D. (1990). *Applied combinatorics with problem solving.* Addison–Wesley, New York.

Jin, W., Riley, R. M., Wolfinger, R. D., White, K. P., Passador-Gurgel, G., and Gibson, G. (2001). The contributions of sex, genotype and age to transcriptional variance in Drosophila melanogaster. *Nature Genetics*, 29(4):389–395.

John, J. A. and Mitchel, T. (1977). Optimal incomplete block designs. *Journal of the Royal Statistical Society, series B*, 39(1):39–43.

Kerr, M. and Churchill, G. (2001a). Statistical design and the analysis of gene expression microarray data. *Genetic Research*, 77:123–128.

Kerr, M. K. (2003). Design considerations for efficient and effective microarray studies. *Biometrics*, 59(4):822–828.

Kerr, M. K. and Churchill, G. A. (2001b). Experimental design for gene expression microarrays. *Biostatistics*, 2(2):183–201.

Kerr, M. K., Martin, M., and Churchill, G. A. (2000). Analysis of variance for gene expression microarray data. *J Comput Biol*, 7(6):819–837.

Khan, A. H., Ossadtchi, A., Leahy, R., and Smith, D. J. (2003). Error-correcting microarray design. *Genomics*, 81(2):157–165.

Kiefer, J. (1959). Optimum experimental designs. *Journal of the Royal Statistical Society, series B*, 21:272–304.

Landgrebe, J., Bretz, F., and Brunner, E. (2004). Efficient design and analysis of two color factorial microarray design. *Computational Statistics & Data Analysis (in press).*

Landgrebe, J. and Lübke, T. (2005). Lysosomal proteome and transcriptome. In *Lysosomes.* Landes Bioscience, New York.

Lee, M.-L. T., Lu, W., Whitmore, G. A., and Beier, D. (2002). Models for microarray gene expression data. *J Biopharm Stat*, 12(1):1–19.

Lee, M.-L. T. and Whitmore, G. A. (2002). Power and sample size for DNA microarray studies. *Stat Med*, 21(23):3543–3570.

Michalewicz, Z. (1992). *Genetic Algorithm + Data Structures = Evolution Programs.* Springer–Verlag.

Newton, M., Kendziorski, C., Richmond, C., Blattner, F., and Tsui, K. (2001). On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *J Comput Biol*, 8(1):37–52.

Nguyen, D. V., Wang, N., and Carroll, R. J. (2004). Evaluation of missing value estimation of microarray data. *Journal of Data Science*, 2:347–370.

Nguyen, N.-K. (1994). Construction of optimal block designs by computer. *Technometrics*, 36(3).

Pukelsheim, F. (1993). *Optimal Designs of Experiments.* John Wiley & sons, New York.

R Development Core Team (2004). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria.

Richardson, J. T., Palmer, M. R., Liepins, G. E., and Hilliard, M. (1989). Some guidelines for genetic algorithms with penalty functions. In *Proceedings of the third international conference on Genetic Algorithms*, pages 191 – 197, San Francisco. Morgan Kaufmann.

Rocke, D. and Durbin, B. (2001). A model for measurement error for gene expression arrays. *J Comput Biol*, 8(6):557–569.

Schuchhardt, J., Beule, D., Malik, A., Wolski, E., Eickhoff, H., Lehrach, H., and Herzel, H. (2000). Normalization strategies for cDNA microarrays. *Nucleic Acids Res*, 28(10):e47.

Searle, S. R. (1971). *Linear Models.* John Wiley & sons, New York.

Shah, K. R. and Sinha, B. K. (1989). *Theory of Optimal Designs.* Springer-Verlag, Heidelberg.

Simon, R., Radmacher, M. D., and Dobbin, K. (2002). Design of studies using DNA microarrays. *Genet Epidemiol*, 23(1):21–36.

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

Smyth, G. K. and Speed, T. (2003). Normalization of cDNA microarray data. *Methods*, 31(4):265–273.

Southern, E., Maskos, U., and Elder, J. (1992). Analyzing and comparing nucleic acid sequences by hybridization to arrays of oligonucleotides: evaluation using experimental models. *Genomics*, 13(4):1008–1017.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525. Evaluation Studies.

Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A*, 98(9):5116–5121.

Vinciotti, V., Khanin, R., D'Alimonte, D., Liu, X., Cattini, N., Hotchkiss, G., Bucca, G., de Jesus, O., Rasaiyaah, J., Smith, C. P., Kellam, P., and Wit, E. (2005). An experimental evaluation of a loop versus a reference design for two-channel microarrays. *Bioinformatics*, 21(4):492–501.

Whitaker, D., Triggs, C. M., and John, J. A. (1990). Construction of block designs using mathematical programming. *Journal of the Royal Statistical Society, series B*, 52(3):497–503.

Whitley, D. (1994). A Genetic Algorithm Tutorial. *Statistics and Computing*, 4:65–85.

Wolfinger, R. D., Gibson, G., Wilfinger, E. D., Bennett, L., Hamadeh, H., Bushel, P., Afshari, C., and Paules, R. S. (2001). Assessing gene significance from cDNA microarray expression data via mixed models. *Journal of Computational Biology*, 8(6):625–637.

Yang, Y., Buckley, M. J., Dudoit, S., and Speed, T. P. (2002a). Comparisons of methods for image analysis on cDNA microarray data. *Journal of computational and graphical statistics*, 11:108–136.

Yang, Y. and Speed, T. (2002). Design issues for cDNA microarray experiments. *Nat Rev Genet*, 3(8):579–588.

Yang, Y. H., Dudoit, S., Luu, P., Lin, D. M., Peng, V., Ngai, J., and Speed, T. P. (2002b). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res*, 30(4):e15.

Zhou, X.-H., Obuchowski, N. A., and McClish, D. K. (2002). *Statistical Methods in Diagnostic Medicine*. Wiley–Interscience, New York.

Curriculum Vitae

31.12.1970	Born in Bogra, Bangladesh
1982–1985	Secondary School Certificate Examination Pabna Zilla School, Bangladesh
1985–1987	Higher Secondary School Certificate Examination Rajshahi College, Bangladesh
1988 – 1992	Bachelor of Science (Hons.) in Statistics Department of Statistics, University of Dhaka, Bangladesh
1993–1995	Master of Science (Thesis group) in Statistics Department of Statistics, University of Dhaka, Bangladesh Supervisor : Prof. M. Ataharul Islam
1996–1999	Lecturer, Institute of Statistical Research and Training University of Dhaka, Bangladesh
1999–2001	Master of Science in Statistics University of British Columbia, Canada Supervisor : Prof. Harry Joe
2002 –2005	Ph.D. student at the Department of Medical Statistics and Center for Statistics, Georg–August–Universität–Göttingen Supervisor : Prof. Dr. Edgar Brunner Co–supervisor : Prof. Dr. Manfred Denker