POWER AND SAMPLE SIZE DETERMINATION FOR TIME COURSE MICROARRAY DIFFERENTIAL EXPRESSION STUDIES: A POSITIVE FALSE DISCOVERY RATE AND PERMUTATION-BASED SIMULATION METHOD

By

Joanne C. Beer

A THESIS

Presented to the Department of Public Health & Preventative Medicine
and the Oregon Health & Science University School of Medicine
in partial fulfillment of the requirements for the degree of

Master of Science

July 2013

Department of Public Health & Preventative Medicine

School of Medicine

Oregon Health & Science University

CERTIFICATE OF APPROVAL

This is to certify that the Master's thesis of

Joanne C. Beer

has been approved

Dongseok Choi, PhD (Thesis Advisor)

Thuan Nguyen, PhD (Committee Member)

Kemal Sonmez, PhD (Committee Member)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank the members of my Thesis Advisory Committee. First, my advisor Dr. Dongseok Choi for his ongoing guidance, encouragement, and the generous amounts of time he spent meeting with me. I am also grateful to him for always having a supply of high quality dark chocolate on hand. Dr. Thuan Nguyen was a pleasure to have as a professor and to later work with as teaching assistant. In Dr. Kemal Sonmez's fascinating biomedical informatics courses I learned about microarray studies and programming for the first time, including the R language. Their comments on this thesis are greatly appreciated. I further thank the coffeehouse Southeast Grind for being the only coffeehouse in Portland open 24 hours a day, and for offering excellent lattes and a soundtrack for my thesis work. Also thanks to Anna Bananas coffeehouse on NW 21st. This thesis would not have been possible if my former employer Dr. Kathleen Myers had not suggested that I pursue study in a biostatistics program, something I had not previously considered, so much thanks to her. Special thanks to my friend Geoffrey Arnold for taking an interest in my project, for allowing me to talk about and reflect on the work, and for offering his advice and wisdom regarding the writing process. Finally, I would not have come so far without the love and support of my family. I dedicate this thesis to my parents, Janet and Donald Beer.

ABSTRACT

Microarray experiments allow researchers to assess the levels of gene expression for tens of thousands of genes at a time. A frequent goal of microarray experiments is to identify genes which are differentially expressed across various biological conditions. Several methods have been developed for determining sample size for differential expression microarray experiments, but few methods have been extended to time course experiments in which gene expression is measured over a series of time points. This thesis proposes a flexible method for sample size and power analysis of time course microarray experiments using a positive false discovery rate type I error control. Because microarray data is often observed to deviate from the assumption of normality underlying the use of parametric t-tests and F-tests, and since it has been increasingly recognized that accounting for the correlation structure of gene expression data is important for accurately estimating error rate and sample size, the method relies on a permutation-based null distribution for the test statistics. Results of simulation-based sample size and power calculations are compared to those of other published sample size methods for both static and time course microarray experiments.

1 Introduction

Measuring the abundance of the various RNA molecules in a cell provides information about the cell's active gene expression profile, known as the transcriptome, which is an important determinant of cellular function and phenotype (Lockhart & Winzeler, 2000). Microarrays have made it possible to measure the abundance of a large number of messenger RNA (mRNA) molecules simultaneously, and thus to compare global gene expression at the transcription level between different cell types, between treatment and control groups, and to examine changes in gene expression levels over time. Determining sample size for microarray experiments is important to ensure the sufficient statistical power necessary to detect biologically meaningful differences while avoiding needless expense, as microarray experiments can be time consuming and expensive. Optimal sample size prevents too many samples and also prevents underpowered studies which use too few samples. Because gene expression is a dynamic process, changing dramatically at times in response to perturbations in the cellular environment and during routine cellular events like DNA replication and cell division (Lockhart & Winzeler, 2000), time course experiments can provide valuable information about changes in gene expression over time. A search of the literature (as of June 2013) found only one published method, Yang et al. (2003), for determining sample size for time course microarray experiments.

Microarray differential expression studies seek significant differences in mean gene expression levels between two or more groups. Typically this involves carrying out a hypothesis test for each gene. After preprocessing of microarray data, significance testing is usually done by calculating a statistic for each gene and comparing these against a cutoff value. Sample size calculation in the classical single-hypothesis testing case, for example when comparing two group means (μ_1, μ_2) using a t-test or a z-

test, depends on several parameters pre-specified by the researcher, including the expected absolute difference in population means ($|\mu_1 - \mu_2| = \Delta$), expected population variances (σ_1^2, σ_2^2), as well as the desired power ($1-\beta$), significance level (α), and group allocation ratio. Assuming data follow normal distributions and an allocation ratio of one, for a two-sided hypothesis test the sample size for each group is given by

$$n_1 = n_2 = \frac{(\sigma_1^2 + \sigma_2^2)(z_{\alpha/2} + z_\beta)^2}{\Delta^2} \tag{1}$$

where $z_{\alpha/2}$ and z_β are values of the standard normal distribution with areas of $\alpha/2$ and β to the right, respectively.

Determining sample size in the case of microarray studies depends on similar parameters, but due to the high dimensionality of microarray data and the inherent dependence of gene expression levels, the procedure is not as straightforward as in the classical single-hypothesis testing case. Microarray experiments are often limited to small sample sizes due to economic or practical constraints. Consequently, the assumption of a normal or t-distribution for test statistics or other assumptions relying on $n \to \infty$ asymptotics may not be valid. Furthermore, since genes function in interdependent networks, some correlation between gene expression is expected, which violates the assumption of independent hypothesis tests and may also contribute to non-normality of test statistics (Efron, 2007). In addition to gene correlation present within individual arrays, experimental methodology may inadvertently introduce correlation across microarrays. Some sample size determination approaches use bootstrapping or permutation-based methods to estimate the null distribution of test statistics in order to avoid making false parametric assumptions and to account for correlation (Li et al., 2005; Lin et al., 2010; Tibshirani, 2006). True mean differences and variances may be projected using similar pilot data, however researchers may

need to decide whether it is reasonable to assume equal true mean differences for all differentially expressed genes and equal variances for all genes. Doing so may simplify sample size calculations, but these assumptions are not realistic. Finally, determination of the appropriate type I error control for each test is complicated in microarray sample size calculation by the multiple testing problem. In addition to these parameters, time course microarray differential expression studies introduce several other variables which may have implications for sample size, including study design (e.g. cross-sectional or longitudinal), the number of time points and their spacing, and the expected temporal trend for gene expression (e.g. linear, periodic, etc.).

2 Background

2.1 Multiple testing problem and false discovery rate

Since microarray experiments involve thousands of hypothesis tests, a hypothesis testing procedure with no correction for multiple testing would result in an unacceptably high type I error rate. For example, at a significance level of 0.05 for each test, when all tests are independent and all null hypotheses are true, testing 10,000 genes would give an expected 500 false positive results by chance. In general, when testing m independent hypotheses each at significance level α, the familywise error rate (FWER), which is the probability of making at least one type I error among all hypothesis tests, is equal to $1 - (1 - \alpha)^m$. The case of $\alpha = 0.05$ is depicted in Figure 1. The FWER quickly approaches one as m becomes large.

The problem of multiple comparisons was first addressed through methods controlling the FWER (defined as $\Pr(V \geq 1)$ according to the notation in Table 1). The simplest approach is the Bonferroni correction, which evenly distributes the FWER between all m hypothesis tests by adjusting the per-comparison significance level to

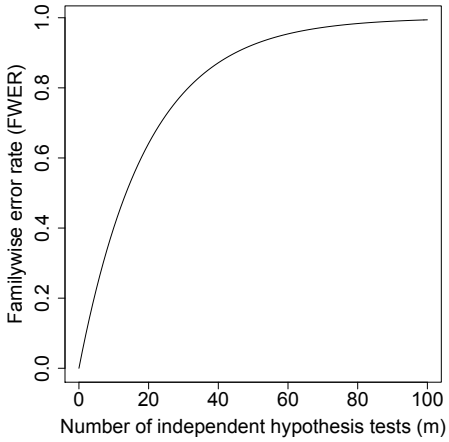

Figure 1: Familywise error rate versus number of independent hypothesis tests, each at a per-comparison significance level of 0.05

$\alpha^* = FWER/m$. Given the number of hypotheses typically tested in microarray studies, this tends to result in a p-value threshold that is too conservative. For the previously discussed example of 10,000 hypothesis tests, $\alpha^* = 5 \times 10^{-6}$. Other stepwise FWER-controlling approaches which rely on first ordering p-values and then calculating a different threshold for each, such as Holm's procedure or the Westfall and Young procedure, give modest improvements in power, but often not enough to detect many real changes in gene expression (Reiner et al., 2003).

Table 1: Possible outcomes of testing m hypotheses

	Declared non-significant (accept null)	Declared significant (reject null)	Total
Null true	U	V	m_0
Alternative true	T	S	m_1
Total	$m - R$	R	m

The FWER may not be the best quantity to control for microarray studies, however. Benjamini and Hochberg (1995) observed that controlling the FWER in the

strong sense, i.e. under all possible configurations of null and alternative hypotheses, as the Bonferroni and other FWER-controlling methods do, makes sense under conditions where an erroneous rejection of the null hypothesis for one test is likely to indicate that other rejections of null hypotheses are also erroneous. When this is not the case, then the number of acceptable erroneous rejections may depend on the total overall number of rejections. As an alternative to FWER control, Benjamini and Hochberg propose to control the false discovery rate (FDR), which is the expected proportion of falsely rejected null hypotheses out of all rejected null hypotheses times the probability of making at least one rejection.

$$FDR = \mathrm{E}\left(\frac{V}{R} I_{\{R>0\}}\right) = \mathrm{E}\left(\frac{V}{R}\middle| R > 0\right) \Pr(R > 0) \tag{2}$$

Benjamini and Hochberg chose the quantity in expression (2) over other potential candidates such as $\mathrm{E}(V/R|R > 0)$, $\mathrm{E}(V|R = r)/r$, or $\mathrm{E}(V)/\mathrm{E}(R)$ because these quantities are equal to one when null hypotheses are true for all tests ($m_0 = m$) and at least one hypothesis is rejected, in which case $v = r$. Consequently, no significance threshold can be chosen such that these quantities are certain to be less than or equal to this threshold regardless of m_0. On the other hand, a sequential p-value method can be applied regardless of the value of m_0 using the quantity given in (2). The Benjamini and Hochberg FDR-controlling procedure involves first ordering the m p-values from smallest to largest, $P_{(1)} \leq P_{(2)} \leq \ldots \leq P_{(m)}$. The null hypotheses are then rejected for p-values less than or equal to $P_{(k)}$, where k is the largest i for which $P_{(i)} \leq iq^*/m$, and q^* is the desired maximum false discovery rate. Familywise error rate is controlled at the FDR level when null hypotheses are true for all tests. The random variable $Q = \mathrm{E}(V/R)$ is defined to be zero when $R = 0$, as no false discovery can be made when there are no rejections. When $m_0 = m$, $s = 0$ and $v = r$, and

thus when $v = 0$, $Q = 0$, and when $v > 0$, $Q = 1$, so $\Pr(V > 1) = \mathrm{E}(Q)$. Hence this procedure provides weak control of FWER. When $m_0 < m$, the FDR is less than or equal to the FWER.

Storey (2002, 2003) defined a quantity called positive FDR (pFDR),

$$pFDR = \mathrm{E}\left(\frac{V}{R}\middle| R > 0\right) = \frac{FDR}{\Pr(R > 0)}. \tag{3}$$

Storey (2003) argues that we should want the false discovery rate to equal one when $m_0 = m$, and cases where no discoveries are made are uninteresting. In fact for genomics studies, $\Pr(R > 0) \approx 1$, so pFDR and FDR are nearly the same. Even so, FDR is actually controlled at $q^*/\Pr(R > 0)$ when positive findings occur, which can be misleading when $\Pr(R > 0)$ is not close to one. Since a sequential p-value method cannot be applied to pFDR, rather than fix the error rate and then estimate the rejection region ($\hat{k}$), as in the Benjamini and Hochberg (1995) method, the procedure for controlling pFDR first fixes the rejection region and then estimates pFDR. This is advantageous since the FDR procedure only controls FDR on average, and the reliability of the method depends on the unknown variability of $\hat{k}$. Furthermore, the pFDR procedure uses information in the data in order to estimate m_0, which results in a more powerful error-controlling method. This is accomplished by assuming that under independence, p-values are exchangeable, coming from the null uniform[0, 1] distribution with probability π_0 and from the alternative distribution with probability π_1. The largest p-values are most likely to come from the null distribution, and so an estimate of $\hat{\pi}_0$ is then obtained for some well-chosen tuning parameter, λ:

$$\hat{\pi}_0(\lambda) = \frac{\#\{p_j : p_j > \lambda\}}{(1 - \lambda)m}. \tag{4}$$

Storey (2003) defines the q-value as the minimum pFDR at which a given observed

statistic can be called significant. This is analogous to the p-value, which can be understood as the minimum type I error rate at which an observed statistic can be called significant.

Microarray studies are often exploratory in nature, and additional research such as real-time reverse transcriptase polymerase chain reaction (RT-PCR), northern blots, or database searches are used to confirm results (Yang & Speed, 2002). Hence, from a microarray experiment, biologists usually would like to obtain a list of candidate genes to be independently verified later using more specific tests. In this context, rather than strictly controlling type I error, a few false positive results are acceptable for the sake of obtaining greater power. Since the emphasis is on gene discovery and because it is more powerful than usual FWER control methods, some version of FDR has been generally accepted as a more suitable basis for type I error control for microarray studies (Allison et al., 2006; Reiner et al., 2003; Storey, 2002). Thus, sample size and power estimation for procedures using FDR may also depend on an estimate of the proportion of true null hypotheses ($\hat{\pi}_0$) and a specified FDR level.

2.2 Microarray sample size calculation methods

Several sample size determination methods for microarray differential expression studies have been published. Nguyen et al. (2008) have reviewed several of these methods and categorize them according to some combination of type of error control (e.g. FDR or FWER), computational method (e.g. analytical, simulation-based), and the underlying assumptions (e.g. independence or dependence of gene expression, true mean differences and variance of gene expression, use of a particular test statistic or study design).

Jung (2005) proposed a sample size procedure designed to control FDR for comparison of gene expression for two groups. Sample size estimation is derived from the

allocation proportion between the two groups, the total number of candidate genes, the number of prognostic (i.e. differentially expressed) genes, the standardized mean differences (i.e. effect sizes) of prognostic genes, the required number of true rejections, and the desired FDR level. The test statistic for a two sample t-test for gene j is given by

$$T_j = \frac{\bar{X}_{1j} - \bar{X}_{2j}}{\hat{\sigma}_j \sqrt{n_1^{-1} + n_2^{-1}}} \tag{5}$$

where $\hat{\sigma}_j^2$ is the pooled sample variance. The effect size for gene j is defined as $\delta_j = [\mathrm{E}(X_{1j}) - \mathrm{E}(X_{2j})]/\sigma_j$. For large n, the test statistic T_j is assumed to be distributed according to the standard normal distribution under the null hypothesis, and normally distributed with mean $\delta_j \sqrt{n a_1 a_2}$ and variance one under the alternative hypothesis, where $n = n_1 + n_2$ is overall sample size and $a_1 = n_1/n$ and $a_2 = n_2/n$ are allocation ratios. Jung notes that if sample sizes must be small then central (under the null) and non-central (under the alternative) t-distributions with $n-2$ degrees of freedom can be used.

Sample size for a two-sided hypothesis test is found by solving

$$h(n) = \sum_{j \in \mathcal{M}_1} \bar{\Phi}(z_{\alpha^*/2} - |\delta_j| \sqrt{n a_1 a_2}) - s = 0, \tag{6}$$

where $\bar{\Phi}(\cdot)$ is the survivor function of the standard normal distribution (1−cumulative distribution function), $\mathcal{M}_1$ is the set of truly differentially expressed genes, which has cardinality m_1, and s is the target number of correctly identified significant genes, which may be less than or equal to m_1. The sum is the expected number of correctly rejected null hypotheses $\mathrm{E}(S)$. Note that power for this method is assessed using the quantity $\mathrm{E}(S)/m_1$, which is the average power (i.e. average sensitivity) over all hypothesis tests. The FDR level is $f = m_0\alpha^*/(m_0\alpha^* + s)$, which is the expected proportion of falsely rejected null hypotheses of all rejections made at a per-comparison

type I error level α^*. For a desired FDR level, power, and projected value for m_0, α^* can be found by $\alpha^* = sf/(m_0(1-f))$. There is a closed form solution for sample size when projected effect sizes are equal for all differentially expressed genes, which is essentially the same sample size formula as in a single-hypothesis testing case with significance level α^* and power $(1-\beta^*)$,

$$n = \left\lfloor \frac{(z_{\alpha^*/2} + z_{\beta^*})^2}{a_1 a_2 \delta^2} \right\rfloor + 1, \tag{7}$$

where $\lfloor x \rfloor$ denotes the greatest integer less than x (i.e. the floor function). Otherwise, if effect sizes are projected to be different, Jung suggests solving numerically using the bisection method.

Liu and Hwang (2007) developed a method which uses many of the same assumptions as Jung (2005). Their method differs in that it is based primarily on central and non-central t-distributions for test statistics corresponding to the null and alternative hypotheses concerning gene expression, respectively, and they use a critical value rather than a type I error rate. They report that their method gives the same result as Jung's in similar settings.

While the assumption of normally distributed gene expression data (after preprocessing) and hence the applicability of t-tests and F-tests is invoked in several sample size methods (Jung, 2005; Liu & Hwang, 2007; Pounds & Cheng, 2005; Yang et al., 2003), this may not always be justifiable for genomics studies (Hardin & Wilson, 2009; Kerr & Churchill, 2001; Nguyen et al., 2008). Nguyen et al. (2008) further note that the assumption of equal effect size required for the closed form solution is unrealistic. Also, while this method assumes independence or weak dependence of gene expression, dependence of gene expression has been shown to affect FDR, power, and sample size estimation (Kim & van de Wiel, 2008; Li et al., 2005).

Jung addresses the issue of dependence by considering a compound symmetry correlation structure, using 400 independent blocks of 10 genes with correlation coefficient 0.6 simulated from a normal distribution. To test the method for non-normal correlated data, gene expression data was generated from a correlated asymmetric distribution. In both cases 5000 simulations, equal allocation ratios, 40 significant genes with an effect size of one, and number of truly alternative genes declared significant (s) equal to 12 or 24 were used. For both the correlated normal and the correlated asymmetric data, the median of the observed true rejections was very close to the nominal value of s, but the interquartile ranges were doubled from the independent case. Hence, there is more variability in the number of true rejections about the nominal value of s in cases of dependent or skewed distributions when using this method.

2.3 Time course microarray experiments

Time course microarray experiments investigate changes in gene expression patterns over time. Androulakis et al. (2007) have classified questions addressed by time course microarray experiments into four broad categories: (1) biological systems analysis, e.g. studies of the cell cycle and circadian clock; (2) growth and development, e.g. stem cell differentiation; (3) disease progression; and (4) response to controlled perturbation, e.g. after drug administration or trauma. In a similar fashion, Tai & Speed (2005) have grouped time course experiments into two categories: (1) *periodic*, characterized by regular patterns of gene expression, including cell cycle and circadian studies; and (2) *developmental*, where there may be few *a priori* expectations regarding temporal gene expression profiles, including studies of growth and treatment.

Another important distinction noted by Tai & Speed (2005), following Diggle et al. (2002) and standard statistical practice for time course studies in general, is that

between *longitudinal* and *cross-sectional* study designs. In longitudinal microarray studies, mRNA is extracted from the same experimental units (i.e. cell line, tissue, or organism) at each time point over the course of the experiment. Thus a temporal gene expression profile can be formed for each individual unit. In cross-sectional studies, on the other hand, mRNA samples are derived from different units at each time point. The gene expression profile obtained is then a population curve, based on the average gene expression measured at each time point. Longitudinal studies are advantageous over cross-sectional studies because they allow researchers to distinguish temporal trends within individuals from differences in baseline levels (Diggle et al., 2002). At the time of their writing, Tai & Speed (2005) note that cross-sectional time course microarray studies were more prevalent in the literature than longitudinal studies, however, which is probably due in part to difficulties in obtaining repeated mRNA samples from one individual.

Yang & Speed (2002) discuss a few ways in which time course microarray experiments differ from usual time course studies. The duration of studies may be relatively short with irregularly spaced time intervals (e.g. 0, 1, 4, 12, and 24 hours after an intervention). There may be only 10 to 20 time points considered. Also, there may be few *a priori* expectations about gene expression profiles, particularly for studies in the aforementioned developmental category.

2.4 Sample size calculation for time course microarray experiments

Yang et al. (2003) consider power and sample size given desired FDR for the case of two groups at a single time point, one group sequential time points, and two groups sequential time points. Their method is designed for cDNA microarray experiments

which may include several replicates for each biological sample. They use the mixed model below for expression level of a gene in the two-group sequential time point comparison case:

$$y_{ijtk} = \mu_{it} + \delta_{ijt} + \epsilon_{ijtk} \tag{8}$$

where i is treatment index, j is subject index, t is time point, and k is the replicate index. The quantity μ_{it} represents the mean expression level for the ith treatment at time t. This is considered to be a fixed effect. The quantity δ_{ijt} is the subject variation, and ϵ_{ijtk} is the experimental variation among replications. Both of these are assumed to be random effects with normal distributions and means of zero. For the case of sequential time points with two treatment groups, the null hypothesis is no difference between groups over time:

$$H_0 : \mu_{11} = \mu_{21}; \mu_{12} = \mu_{22}; \ldots; \mu_{1T} = \mu_{2T} \tag{9}$$

and the alternative is modeled using the linear expressions $\mu_{2t} = \mu_{1t} + \gamma t$ and $\delta_{ijt} = \delta_{ij} t$. The significance test used is a two-way analysis of variance with time and subject as factors. This model was designed to account for variation in slope between subjects which is often present in the case of longitudinal time course data with a linear time effect. The authors note that they found little increase in power when extending the model to include a quadratic time effect, and maintain that the linear model can serve as a good approximation to a quadratic effect when time is considered as a discrete variable. The adjusted significance level for each hypothesis test is $\alpha^* = \alpha \max(1, s)/m$, where α is the FDR, s is the expected number of significant genes, and m is the total number of genes. When $s = 0$ this is equivalent to the Bonferroni correction.

Liu and Hwang (2007) compared their own approach for the one time point,

two-sample case, which uses pFDR, against that of Yang et al. (2003), Pounds and Cheng (2005), Jung (2005), and simulation results. For the case of a two-sample t-test, with fixed and equal ratios of true mean difference (Δ) and standard deviation (σ) assumed for all differentially expressed genes ($\Delta/\sigma = 1$ and $\Delta/\sigma = 2$), and the proportion of truly null genes set at $\pi_0 = 0.5$, 0.9, and 0.95, the authors find that the method of Yang et al. consistently overestimates sample size with respect to both the Liu and Hwang method, Jung's method, and their simulation results, which agree fairly closely, while the Pounds and Cheng method underestimates sample size when $\Delta/\sigma = 1$.

2.5 Non-normality and dependence of gene expression

As previously mentioned, many significance testing and sample size calculation methods for microarrays rely on the assumption that gene expression data follow a normal distribution, and so t-tests and F-tests are appropriate. However, many researchers have noted non-normality in gene expression data. Kerr and Churchill (2001) state that they have consistently observed deviance from normality of residuals after fitting an analysis of variance (ANOVA) model to gene expression data, and prefer bootstrapping methods to classical t-tests and F-tests. Lee and Whitmore (2002) also report reluctance to assume a normal error term for their ANOVA model based on their experience with microarray data. Hardin and Wilson (2009) showed that after applying one of several standard methods of pre-processing to Affymetrix oligonucleotide array technical replicate spike-in data, distributions of gene expression data tend to be heavy-tailed and skewed, and they recommend methods that are robust against non-normality.

Another common simplifying assumption made in analysis of microarray data is that gene expression values are independent. Since genes are known to interact in

various regulatory networks, their expression levels are in fact not completely independent, and this may have deleterious consequences for significance testing and sample size calculations. Efron (2004, 2007) discusses how correlation can cause considerable widening or narrowing of the effective null distribution of test statistics. Kim and van de Wiel (2008) examined the performance of Benjamini and Hochberg's FDR, Storey's q-value, Significance Analysis of Microarray (SAM) (Tusher et al., 2001), and resampling-based significance testing procedures under conditions of dependence. They found that the adaptive Benjamini and Hochberg FDR-controlling procedure (Benjamini et al., 2006) was most robust under dependence, while the SAM and q-value methods underestimated the FDR. Li et al. (2005) also looked at the effect of dependence on FDR control with particular attention to implications for power and sample size. Their simulation, using real microarray data, showed that the Benjamini and Hochberg (1995) and Storey and Tibshirani (2003b) FDR controlling procedures (direct and resampling-based) under-control FDR when there is dependence among genes and the proportion of positive genes ($\pi_1 = 1 - \pi_0$) is less than 10 percent, which is a realistic scenario. Actual FDR was found to be as much as twice the pre-specified level, and thus the authors were able to control FDR at the desired level by making a crude adjustment, dividing the pre-specified FDR in half. They recommend using this crude adjustment given the absence of an analytical formula for accurately controlling FDR in general conditions. Li et al. (2005) conclude that the proportion of truly significant genes has a positive association with power and so this quantity, along with variance, true mean difference, and correlation, affects sample size estimation.

Nonparametric, resampling-based approaches including bootstrap or permutation methods have been used to account for non-normality and dependence in microarray data. Researchers may want to use resampling-based methods to avoid mak-

ing assumptions and obtain a "model-free" approach. R. A. Fisher first described a permutation-based hypothesis test in the classic text *The Design of Experiments* (originally published in 1935) (Fisher, 1966; Larson, 2011). Fisher's exact test for 2 $\times$ 2 tables functions by conditioning on the marginal totals to obtain the null distribution, counting the number of cases as or more extreme than the observed data, and dividing by the total number of permutations to obtain an exact p-value. The test assumes that observations are exchangeable under the null hypothesis, since they all come from the same distribution under the null (Larson, 2011). For a static (i.e. single time point) microarray study, the distribution of null test statistics is estimated in a similar manner by randomly permuting the group assignment labels, which amounts to permuting the columns of the matrix of gene expression data (where rows correspond to genes and columns correspond to individual units), and calculating test statistics for each permutation. As discussed in Efron (2012), using permutation preserves the correlation structure between genes, since entire columns are permuted intact, but permutation does not address potential correlation across individual units.

The null distributions derived by permutation are discrete, and this can be an issue when considering small sample sizes. In some cases the marginal type I error rate necessary to achieve a pre-specified level for FDR may be less than the smallest possible p-value, which is $1/B_{\max}$, where $B_{\max}$ is the maximum possible number of permutations. One strategy to overcome the problem of granularity is to pool the null test statistic distributions for all of the genes. If the test statistics are identically distributed and thus exchangeable under the null hypothesis, then a better estimate of the null distribution will be obtained when information from all genes is incorporated (Storey & Tibshirani, 2003a).

For example, as an alternative to sample size determination approaches that employ unrealistic assumptions such as independence of genes, equal variances, or equal

correlations between genes, Tibshirani (2006) proposed a permutation method which uses pilot data. The standard deviation for each gene and the overall null distribution of gene scores is estimated from pilot data by a permutation procedure, and then the FDR and false negative rate (FNR), a measure of type I error, are estimated for a given hypothesized mean difference in gene expression. This approach is similar to that of Li et al. (2005), with the exception that for Tibshirani's method test statistics (which are the same as the gene scores used in SAM (Tusher et al., 2001)) are permuted rather than the data, and thus the method is more generally applicable (e.g. to survival analysis data), and Tibshirani's method also reports FNR. Simulation studies were carried out with this method using both uncorrelated and correlated simulated data. One thousand genes and 20 samples were used, with 10 in each class. Measurements were generated from a standard normal distribution, with a mean difference of twofold change for significant genes, and various proportions of truly significant genes were tested. A correlation structure was introduced consisting of 10 blocks of 100 genes, with pairwise correlations of 0.5 in each block. The FDR and FNR curves were similar for uncorrelated and correlated data, but the 10 and 90 percentile curves for FDR and FNR were much wider in the correlated data, and thus Tibshirani advises large sample size and preserving the correlation structure in the genes rather than assuming independence, which is unrealistic.

Lin et al. (2010) compared Jung's (2005) method, referred to as the univariate method since it uses an independent hypothesis test for each gene, against several other sample size determination methods for proportion of truly significant genes (π_1) equal to 5, 10, and 20%, and desired power at 60, 70, 80, and 90%. The permutation method of Lin et al. uses a *confidence probability* formulation, originally proposed by Wang and Chen (2004), which is designed to achieve a certain power (i.e. sensitivity) for each gene with a specified probability (the authors use 95%), rather than the *aver-*

age power formulation used in Jung's and Tibshirani's methods. Under conditions of independent gene expression, using simulated data, Lin et al. compare the univariate method with a method also using the confidence probability formulation proposed by Tsai et al. (2005). The univariate method results in sample sizes that are either equal or one less than those given by the Tsai et al. (2005) method. It also achieves the desired average power or greater in each case except for one, and very close to the desired FDR in all cases. Under conditions of gene dependence, using a colon cancer dataset with average gene correlation of approximately 0.4, the authors compare the univariate method, using the average power formulation, with their own permutation method and the permutation method of Tibshirani, both using the confidence probability formulation. The authors confirm Jung's results using the univariate method, which achieves the desired FDR or less and reasonably close to the desired average power under conditions of dependence. However, in this case the sample size given by the univariate method ranges from 3 to 5 less than that given by the Lin et al. method. Tibshirani's method also results in larger sample sizes than the univariate method under dependence. The authors conclude that considerations of dependency are important for the methods using the confidence probability formulation, because sensitivity distributions depend on both effect sizes and correlations between genes, but for the average power formulation the univariate method may be used without accounting for correlation structure, in contrast to what was suggested by Li et al. (2005). They do not discuss differences in the variation of achieved FDR between the independent and dependent cases for the univariate method.

Contrary to the conclusion of Lin et al. (2010), given that both Jung (2005) and Tibshirani (2006) report greater variation in the FDR achieved when correlation structure is present, and Li et al. (2005) found FDR was under-controlled when gene correlation structure was present, this suggests that, while the crude estimate

suggested by Li et al. (2005) may be too extreme, correlation should not be entirely disregarded when attempting to achieve a given FDR level and average power in sample size calculation.

3 Methods

The primary objective of this work was to develop a flexible method for sample size and power analysis of time course microarray experiments. The method uses a positive false discovery rate type I error control and includes a permutation step for deriving the null distribution of the test statistics. A simulation study was carried out to determine average power for various sample sizes at particular chosen values of true mean difference, variance, number of time points, and desired FDR. For the purposes of the simulation, it was assumed that microarray data had already been $\log_2$-transformed and preprocessed to remove technical variation. It was further assumed that arrays were independent both within and across time points, i.e. that the data was derived from a cross-sectional (with a new group of subjects at each time point) rather than a longitudinal (following the same subjects over time) study design.

3.1 Cases considered

The following three cases were considered:

3.1.1 Two groups, single time point

For sample groups $i \in \{1, 2\}$ and for each gene $j \in \{1, 2, \ldots, m\}$, the null and alternative hypotheses were the following:

$$H_0 \colon \bar{x}_{1j} = \bar{x}_{2j} \text{ versus } H_1 \colon \bar{x}_{1j} \neq \bar{x}_{2j}$$

The test statistic was the same t-statistic used by Jung (2005),

$$T_j = \frac{\bar{X}_{1j} - \bar{X}_{2j}}{\hat{\sigma}_j \sqrt{n_1^{-1} + n_2^{-1}}},$$

where the pooled sample variance is given by $\hat{\sigma}_j^2 = \frac{(n_1-1)\hat{\sigma}_{1j}^2 + (n_2-1)\hat{\sigma}_{2j}^2}{n_1+n_2-2}$. Equal sample sizes $n_1 = n_2 = n_i$ were used for each group, so the overall sample size was $n_{\text{total}} = 2n_i$.

3.1.2 One group, time trend

For time points $t = 1, 2, \ldots, t_{\max}$ and for each gene $j \in \{1, 2, \ldots, m\}$, an ordinary least-squares regression model,

$$Y_j = \beta_{0j} + \beta_{1j} t + \epsilon_{jt},$$

was fit to the data, where Y_j is the mean gene expression level. The null hypothesis was no linear trend over time, and the alternative hypothesis was the presence of a linear trend over time.

$$H_0 \colon \beta_{1j} = 0 \text{ versus } H_1 \colon \beta_{1j} \neq 0$$

A fixed sample size n_t was used at each time point, so the overall sample size was equal to $n_{\text{total}} = n_t t_{\max}$. The test statistic was $T_j = \frac{\hat{\beta}_{1j}}{S_{\hat{\beta}_{1j}}}$, where

$$S^2_{\hat{\beta}_{1j}} = \frac{S^2_{Y_j|T}}{S^2_T(n_{\text{total}}-1)} = \frac{\frac{\sum_{i=1}^{n_{\text{total}}}(Y_{ij}-\hat{Y}_{ij})^2}{n_{\text{total}}-2}}{\frac{\sum_{i=1}^{n_{\text{total}}}(T_i-\bar{T})^2}{n_{\text{total}}-1}(n_{\text{total}}-1)} = \frac{\sum_{i=1}^{n_{\text{total}}}(Y_{ij}-\hat{Y}_{ij})^2}{(n_{\text{total}}-2)\sum_{i=1}^{n_{\text{total}}}(T_i-\bar{T})^2}.$$

3.1.3 Two groups, time trend

For time points $t = 1, 2, \ldots, t_{\max}$, sample groups $i \in \{1, 2\}$, and for each gene $j \in \{1, 2, \ldots, m\}$, an ordinary least-squares regression model,

$$Y_j = \beta_{0j} + \beta_{1j}t + \beta_{2j}I_{\{\text{Group}\}} + \beta_{3j}tI_{\{\text{Group}\}} + \epsilon_{jt},$$

was fit to the data, where Y_j is the mean gene expression level, and where $I_{\{\text{Group}\}} = 0$ for the control group and $I_{\{\text{Group}\}} = 1$ for the treatment group. The null hypothesis was no difference in linear trend over time between the two groups (i.e. no significant time-group interaction), and the alternative hypothesis was a difference in linear trend over time between the two groups (i.e. the presence of a significant time-group interaction).

$$H_0 \colon \beta_{3j} = 0 \text{ versus } H_1 \colon \beta_{3j} \neq 0$$

Fixed and equal sample sizes $n_{1t} = n_{2t} = n_{it}$ were used at each time point, thus the overall sample size was equal to $n_{\text{total}} = 2n_{it}t_{\max}$. The test statistic was $T_j = \frac{\hat{\beta}_{3j}}{S_{\hat{\beta}_{3j}}}$, where $S^2_{\hat{\beta}_{3j}}$ is the fourth diagonal element of the variance-covariance matrix of the least squares parameter estimates, given by $\text{Var}(\hat{\beta}) = (\mathbf{T}^T\mathbf{T})^{-1}\sigma^2$. Here $\mathbf{T}$ is an $m \times 2n_{it}t_{\max}$ matrix of integers corresponding to the time points. As specified in

Hastie et al. (2009), the variance σ^2 is estimated by

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n_{\text{total}}} (Y_{ij} - \hat{Y}_{ij})^2}{n_{\text{total}} - 4}.$$

3.2 Simulation algorithm

For each case, the following steps were carried out for values of sample size:

3.2.1 Data simulation

For the sake of simplicity in the initial simulation study, equal gene expression variance was assumed for all genes. Upon examining distributions of gene expression variances in a pilot microarray dataset (Appendix 8.1 and 8.2), a value of $\sigma_j^2 = 0.1$ was determined to be reasonable given the variance medians for each sample group. Likewise, equal true mean difference (Δ) was assumed for all differentially expressed genes in a given simulation. In each of the three cases discussed above, the gene expression data for the m_0 genes that were not differentially expressed was randomly sampled from a Normal$(0, 0.1)$ distribution. For the two group, single time point case, the m_1 differentially expressed genes were randomly sampled from a Normal$(\Delta, 0.1)$ distribution. For the time trend models, the m_1 differentially expressed genes were randomly sampled from a Normal $\left(\Delta \frac{t-1}{t_{\max}-1}, 0.1\right)$ distribution for $t = 1, 2, \ldots, t_{\max}$, such that they achieved the desired true mean difference at the end of the series of time points.

3.2.2 Permutation step

Simulated gene expression data was organized into matrices with 5000 rows representing genes and n_{total} columns corresponding to individual microarrays, which were themselves assigned to a group, time point, or both. For the two group cases, the

permutation step was achieved by permuting the columns of the gene expression data at random between the treatment and control groups (i.e. random assignment of arrays to treatment or control group). Permutation was done within time points for the two group time trend case, that is, assignment of each array to a particular time point was held fixed while treatment/control group assignment was permuted. For the one group time trend case, permutation was done across time points, such that the groups of n_t arrays at each time point were altogether randomly assigned a new unique time point. A test statistic was calculated for each permutation, generating an null distribution for each gene, and the null test statistics were pooled across all genes, resulting in a total number of null test statistics equal to mB, where m is the total number of genes and B is the number of permutations carried out.

For the two group single time point case, with equal group sample sizes $n_1 = n_2 = n_i$, the maximum total number of permutations possible ($B_{\max}$) is $\binom{2n_i}{n_i}$. For the one group time trend case with an equal sample size of n_t at each time point t, where $1 < t \leq t_{\max}$, $B_{\max} = t_{\max}!$. And for the two group time trend case, with equal sample sizes $n_{1t} = n_{2t} = n_{it}$ at each time point t, columns of gene expression data were permuted for each time point independently. Thus, $B_{\max}$ is $\binom{2n_{it}}{n_{it}}$ for each time point. For $t_{\max}$ time points, each with $\binom{2n_{it}}{n_{it}}$ permutations, there are $\binom{2n_{it}}{n_{it}}^{t_{\max}}$ possible unique combinations from which to calculate the null distribution of test statistics.

For each of the three cases considered, the number of permutations was chosen to achieve a balance between having a sufficient number of null test statistics to avoid the potential problem of granularity in the associated p-value distribution, as previously discussed, while simultaneously maintaining a manageable computing time. Values of n and $t_{\max}$ for the two groups single time point and one group time trend cases, respectively, were such that it was feasible to use the maximum possible number of permutations. For the two group time trend case, $B = 100$ was used. See Table 2 for

the exact number of permutations used in each case.

3.2.3 Significance test

FDR control was achieved using Storey's (2002) positive FDR procedure, as was used in Jung (2005), along with the assumption that $P(R > 0) \approx 1$ for large m, so the pFDR is equivalent to the FDR. A p-value was calculated for each of the j genes by the formula

$$p_j = \frac{\#\{T_b : |T_b| \geq |T_j|\}}{B}$$

where B is the total number of permutations used to derive the null distribution of test statistics, and T_b represents the values of the the null test statistics. The number of truly null genes was estimated by

$$\hat{m}_0(\lambda) = \frac{\#\{p_j : p_j > \lambda\}}{1 - \lambda},$$

with the tuning parameter $\lambda = 0.5$, as in Jung (2005). The q-value for each gene was found by the algorithm given in Storey (2002). This involves ordering the p-values from smallest to largest, $p_{(1)} \leq \ldots \leq p_{(m)}$. The corresponding q-value for each p-value is calculated by first estimating $\hat{q}(p_{(m)})$,

$$\hat{q}(p_{(m)}) = \widehat{\mathrm{FDR}}(p_{(m)}) = \frac{\hat{m}_0 p_{(m)}}{\#\{p_j : p_j \leq p_{(m)}\}},$$

and then, in order to ensure that $\hat{q}(p_{(1)}) \leq \ldots \leq \hat{q}(p_{(m)})$, setting each $\hat{q}(p_{(i)})$ equal to $\min\{\widehat{\mathrm{FDR}}(p_{(i)}), \hat{q}(p_{(i+1)})\}$ for $i = m-1, m-2, \ldots, 1$. The null hypothesis was rejected for genes with $\hat{q}_j \leq f$, where f is the pre-specified FDR level.

3.2.4 Power calculation

Power for each iteration of the simulation was determined by calculating the proportion of truly differentially expressed genes that were discovered:

$$\text{Power} = \frac{s}{m_1} = \frac{\sum_{j \in \mathcal{M}_1} I_{\{\hat{q}_j \leq f\}}}{m_1}.$$

The values of s, r, estimated FDR and true FDR were also obtained for each iteration. These were then averaged after iterating the simulation 1000 times with each set of fixed parameter values. Calculations were done for desired FDR (f) equal to 0.05 and again for 0.10. For a given set of parameters including true mean difference (Δ), proportion of differentially expressed genes (π_1), and number of time points ($t_{\max}$) where applicable, sample size was increased until average power was at least 80% in most cases.

Simulations were coded in R version 3.0.0 (R Core Team, 2013). See Appendix 6.3 for R code.

3.3 Parameter values

Table 2: Simulation parameters

Parameter	Values
For all cases	
Total number of genes (m)	5000
Gene expression variance (σ_j^2)	0.1
Difference in mean expression:	
Fold change ($FC = 2^{\Delta}$)	1.50, 2.00, 2.50, 3.00
Absolute effect size ($\Delta = \log_2 FC$)	0.58, 1.00, 1.32, 1.58
Standardized effect size (Δ/σ_j)	1.85, 3.16, 4.18, 5.01
FDR control level (f)	0.05, 0.1
Tuning parameter for pFDR procedure (λ)	0.5
Number of iterations for each simulation	1000
Two groups, single time point	
Sample size per group (n_i)	$3, 4, 5$
Proportion of differentially expressed genes (π_1)	0.01, 0.05, 0.1, 0.2
Number of permutations ($B = \binom{2n_i}{n_i}$)	20, 70, 252
One group, time trend	
Sample size at each time point (n_t)	4 to $16, 20, 25, 30, 40, 50$
Number of time points ($t_{\max}$)	5
Proportion of differentially expressed genes (π_1)	0.1
Number of permutations ($B = t_{\max}!$)	120
Two groups, time trend	
Sample size per group at each time point (n_{it})	2 to 15
Number of time points ($t_{\max}$)	$5, 10$
Proportion of differentially expressed genes (π_1)	0.1
Number of permutations (B)	100

4 Results

4.1 Two groups, single time point

Examples of the distributions of null test statistics and the actual test statistics for the two groups single time point case are presented in Figure 2. The p-value and q-value distributions corresponding to this particular iteration of the given parameters are presented in Figure 3. Plots of sample size per group versus average power over 1000 simulation iterations for chosen values of π_1, fold change (FC), and desired FDR ($f = 0.05, 0.10$) are shown in Figure 4. This data is also summarized in Tables 10 and 11 in Appendix 8.2 along with the average values for s, r, estimated and true FDR.

Sample size per group of 3, 4, and 5 were simulated for most combinations of parameters. Sample size of 3 resulted in zero average power for all of the parameter combinations with $f = 0.05$, and very low power in nearly all of the cases with $f = 0.10$ with the exception of the cases where $\pi_1 = 0.2$ and $FC = 2.5$ or 3 (corresponding to effect sizes 4.18 and 5.01), where average (standard deviation) power was 31.2 (8.7) and 65.9 (5.5) percent, respectively. None of the combinations with $FC = 1.5$ (effect size 1.85) achieved 80% power at $n_i = 5$, with the maximum average power for $FC = 1.5$ occurring at $f = 0.10$, $\pi_1 = 0.2$ resulting in 42.9 (2.9) percent average power. Table 3 shows minimum sample size necessary to achieve 80% power for all cases where this was achieved by $n_i = 5$.

As shown in Tables 10 and 11, the average true FDR was below desired FDR for all combinations of simulation parameters. In many cases the average true FDR was in fact less than the average estimated FDR. (Please note that average FDR values are omitted for cases where some simulation iterations returned no significant genes, i.e. for parameter combinations where at least one iteration had $r = 0$.)

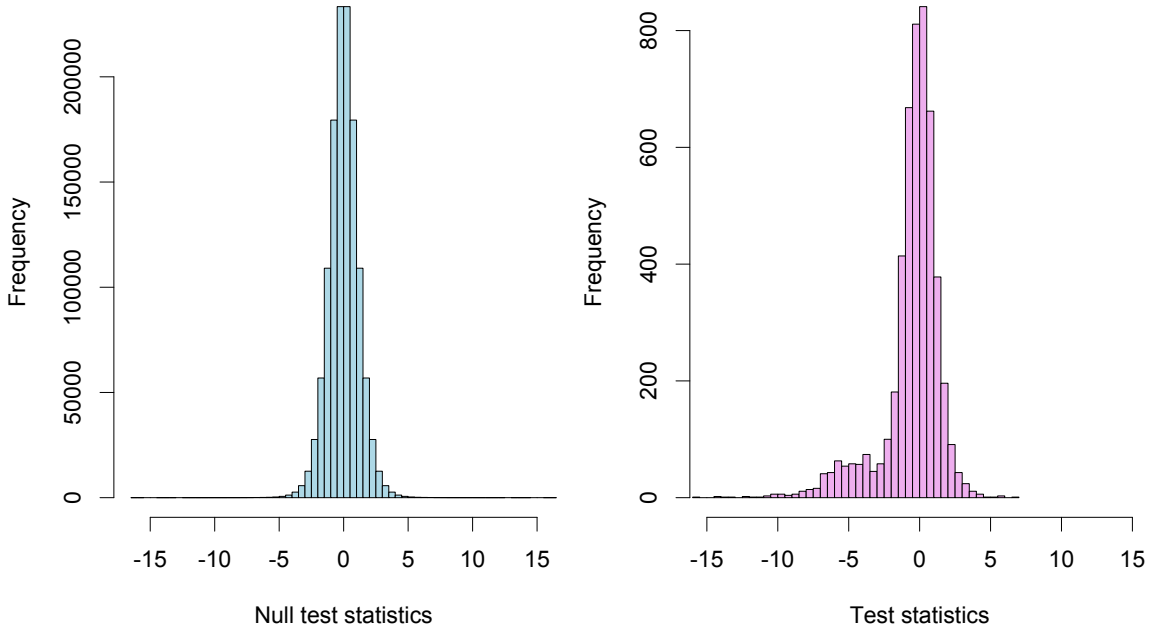

Figure 2: Two groups, single time point: Example of distributions of null test statistics and test statistics. Parameters: $m = 5000$, $n_i = 5$, Fold change $= 2$, $B = 252$. Total number of pooled null test statistics is $m \times B = 1,260,000$

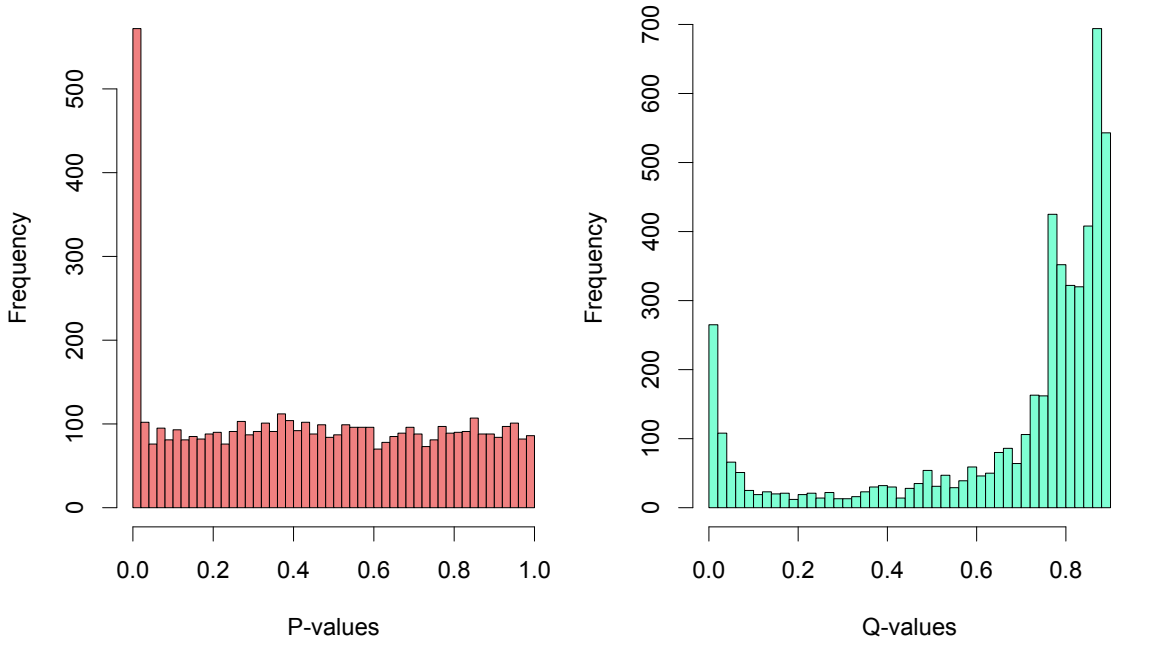

Figure 3: Two groups, single time point: Example of p-value and q-value distributions, using the same parameters as in Figure 2. The total number of p-values and q-values is $m = 5000$.

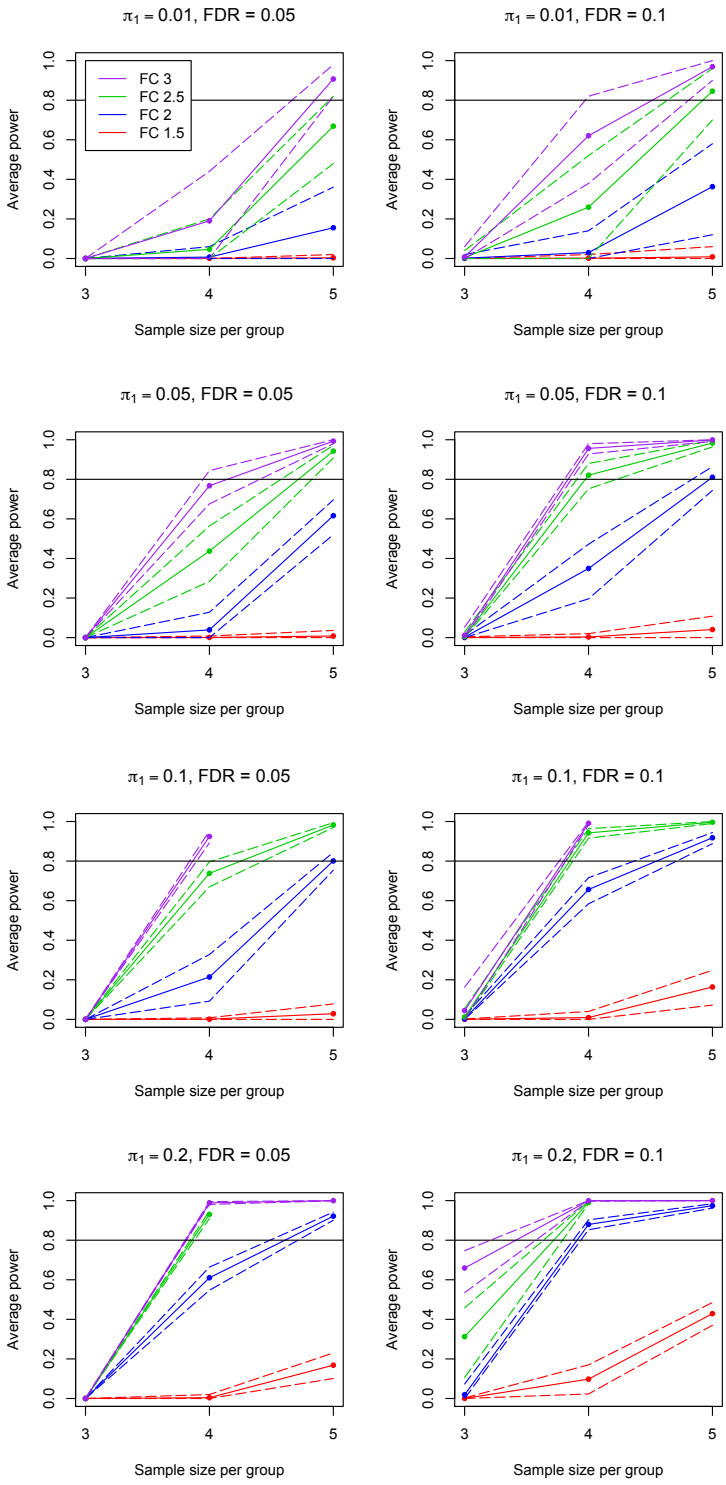

Figure 4: Two groups, single time point: Average power versus sample size per group. FDR = 0.05 (left) and FDR = 0.10 (right). Dashed lines are 2.5 and 97.5 percentiles.

Table 3: Two groups, single time point. Sample size necessary to achieve at least 80% average power. Values given are sample size at each time point (average percent power over 1000 simulations (standard deviation)).

		$f = 0.05$	$f = 0.10$
π_1	FC (effect size)	n_i (Power (sd))	n_i (Power (sd))
0.01	1.5 (1.85)		
	2.0 (3.16)		
	2.5 (4.18)		5 (84.5 (6.5))
	3.0 (5.01)	5 (90.7 (4.6))	5 (96.9 (2.6))
0.05	1.5 (1.85)		
	2.0 (3.16)		5 (81.0 (3.1))
	2.5 (4.18)	5 (94.2 (1.6))	4 (82.1 (3.3))
	3.0 (5.01)	5 (99.3 (0.5))	4 (95.6 (1.5))
0.10	1.5 (1.85)		
	2.0 (3.16)	5 (80.1 (2.3))	5 (91.8 (1.4))
	2.5 (4.18)	5 (98.3 (0.6))	4 (94.3 (1.2))
	3.0 (5.01)	4 (92.4 (1.5))	4 (99.1 (0.4))
0.20	1.5 (1.85)		
	2.0 (3.16)	5 (92.2 (1.0))	4 (88.0 (1.3))
	2.5 (4.18)	4 (93.1 (1.0))	4 (99.1 (0.3))
	3.0 (5.01)	4 (98.9 (0.4))	4 (99.9 (0.1))

4.2 One group, time trend

Figure 5 shows an example of simulated data with fitted regression line, as well as one example permutation of the time points and the resulting fitted regression line. Plots of sample size per time point versus average power over 1000 simulation iterations are shown in Figure 6, for $t_{\max} = 5$, $\pi_1 = 0.1$, the various chosen values for fold change, and desired FDR ($f = 0.05, 0.10$). This data is also summarized in Table 12 in Appendix 8.2 along with the average values for s, r, estimated and true FDR.

Table 4 shows minimum sample size necessary to achieve 80% power for all cases considered. When $f = 0.05$, the two larger effect sizes, i.e. $FC = 2.5$ and 3 (effect sizes 4.18 and 5.01), achieved over 80% power at the smallest sample size per time

point that was simulated, $n_t = 4$. For $FC = 1.5$ (effect size 1.85), sample size $n_t = 50$ was sufficient to achieve average percent power of 84.2 (2.1), but it is likely that a slightly smaller sample size would work as the next largest sample size simulated was $n_t = 40$ with average percent power of 78.0 (2.4). When $f = 0.10$, each of $FC = 2$, 2.5, and 3 achieved over 80% power at the smallest sample size, $n_t = 4$.

As shown in Table 12, the average true FDR was controlled well below the desired level in all cases, with a maximum average percent true FDR of only 0.13% when desired FDR was set at 5% and a maximum average true FDR of 2.73% when desired FDR was set at 10%. So for the one group time trend case, increasing the desired FDR from 5% to 10% resulted in an increase in power given a particular sample size without a large increase in the actual FDR. Meanwhile the average estimated FDR tended to be rather close to the desired FDR levels. Thus, FDR tended to be overestimated for the one group time trend simulations.

Table 4: One group, time trend. Sample size necessary to achieve at least 80% average power, for $\pi_1 = 0.1$, $t_{\max} = 5$. Values given are sample size at each time point (average percent power over 1000 simulations (standard deviation)).

	$f = 0.05$	$f = 0.10$
FC (effect size)	n_t (Power (sd))	n_t (Power (sd))
1.5 (1.85)	50 (84.2 (2.1))	15 (81.1 (2.0))
2.0 (3.16)	9 (80.5 (2.3))	4 (82.9 (2.0))
2.5 (4.18)	4 (82.8 (2.0))	4 (97.6 (0.8))
3.0 (5.01)	4 (94.5 (1.2))	4 (99.7 (0.3))

4.3 Two groups, time trend

Figure 7 shows an example of simulated data for the two group time trend case both before and after one permutation step, with fitted regression lines according to group assignment. Plots of sample size at each time point versus average power are shown

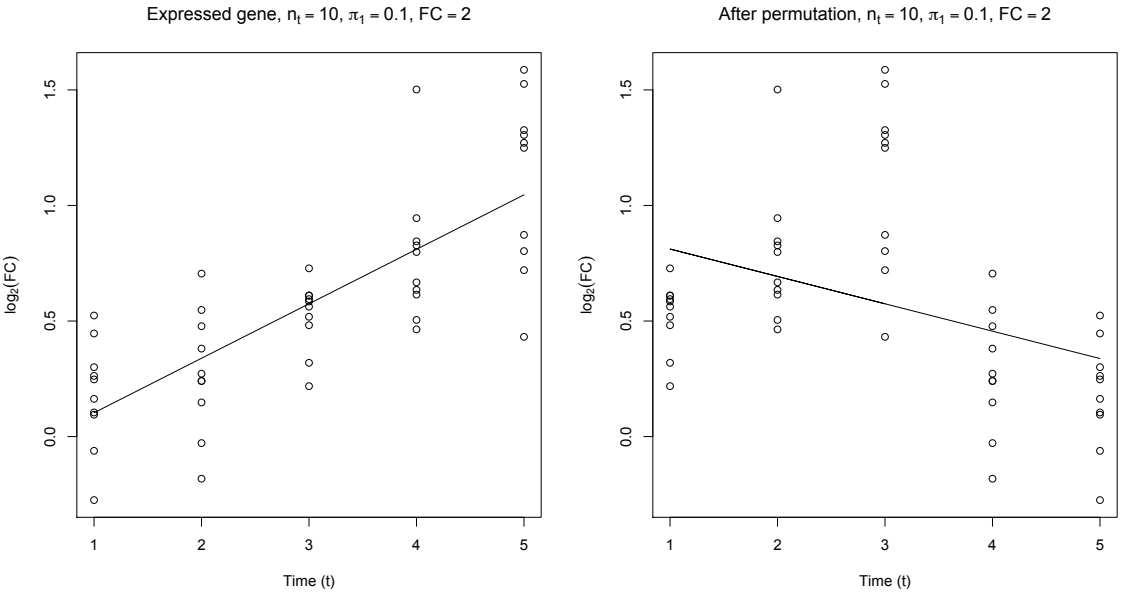

Figure 5: One group, time trend: Example of simulated data with fitted regression line. Before (left) and after (right) one permutation.
Parameters: $n_t = 10$, $t_{\max} = 5$, $\pi_1 = 0.1$, $FC = 2$.

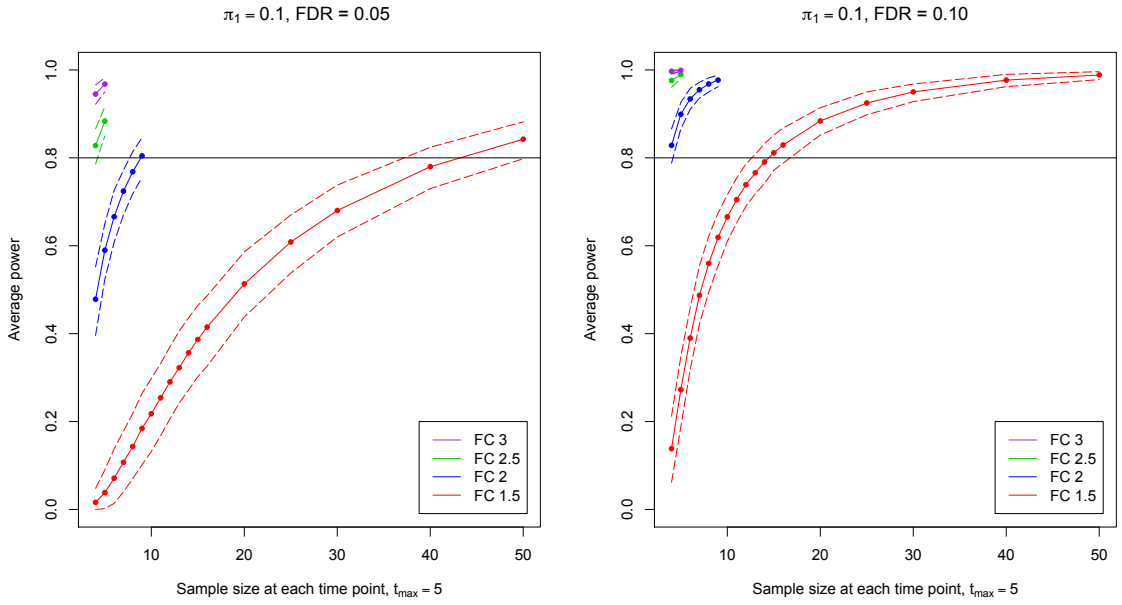

Figure 6: One group, time trend: Average power versus sample size. FDR = 0.05 (left) and FDR = 0.10 (right). Dashed lines are 2.5 and 97.5 percentiles.

in Figure 8 for $t_{\max} = 5$ and in Figure 9 for $t_{\max} = 10$. All simulations were done with $\pi_1 = 0.1$, at the various chosen values for fold change, and as in the other cases, calculations were done for desired FDR 0.05 and 0.10. The data for average s, r, power, estimated and true FDR is given in Tables 13 and 14 in Appendix 8.2 for $t_{\max} = 5$ and $t_{\max} = 10$, respectively. The sample size per group at each time point that was necessary to achieve at least 80% power is summarized in Table 5.

Necessary sample size per group at each time point in the two group time trend case tends to be less than the necessary sample size per time point for the one group time trend case. This may be in part because the permutation method in the two group time trend case, i.e. permuting the group assignment at each time point, is less likely to result in a significant interaction term, i.e. results in a narrower distribution of null test statistics. It is also notable that increasing the number of time points from 5 to 10 results in a slight increase in power for a given sample size. In contrast to the one group time trend case, for the two group time trend case the estimated FDR tended to be very close to the true FDR for both values of desired FDR, $f = 0.05$ and $f = 0.10$, as can be seen in Tables 13 and 14. The increase in both power and FDR with respect to the one group time trend case seems to exemplify the trade-off between type I and type II error which is typical of statistical hypothesis tests.

Table 5: Two groups, time trend. Sample size necessary to achieve at least 80% average power, for $\pi_1 = 0.1$. Values given are sample size per group at each time point (average percent power over 1000 simulations (standard deviation)).

	$f = 0.05$		$f = 0.10$	
	$t_{\max} = 5$	$t_{\max} = 10$	$t_{\max} = 5$	$t_{\max} = 10$
FC (effect size)	n_{it} (Power (sd))	n_{it} (Power (sd))	n_{it} (Power (sd))	n_{it} (Power (sd))
1.5 (1.85)	14 (83.4 (1.9))	9 (86.5 (1.6))	12 (83.4 (2.0))	7 (80.6 (2.1))
2.0 (3.16)	5 (81.1 (2.1))	3 (81.4 (2.0))	5 (88.5 (1.6))	3 (88.6 (1.6))
2.5 (4.18)	4 (95.0 (1.0))	2 (84.0 (1.9))	3 (88.2 (1.6))	2 (91.4 (1.4))
3.0 (5.01)	3 (94.7 (1.1))		3 (97.5 (0.7))	

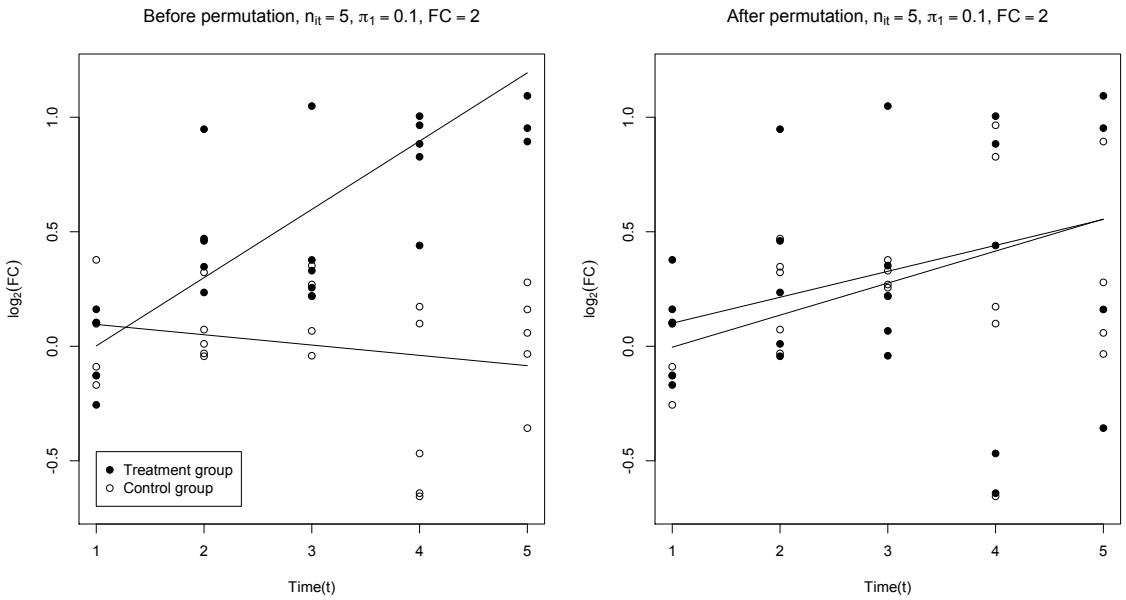

Figure 7: Two groups, time trend: Example of simulated data with fitted regression lines. Before (left) and after (right) one permutation.
Parameters: $t_{\max} = 5$, $n_{it} = 5$, $\pi_1 = 0.1$, $FC = 2$.

5 Discussion

Table 6 shows a comparison of the simulation results for the two group, single time point case with the calculation using Jung's method as given by Equation 7. In almost all cases the sample size estimate from Jung's method is slightly smaller than that derived from the simulation.

For the time trend cases, while Yang et al. (2003) had a slightly different model, we may still want to compare their sample size results. Their simulation used $m = 5000$ genes and a twofold change in gene expression at the end of the time series, which is closest to the $FC = 2$, effect size 3.16 case for the present study. Recall that the Yang et al. type I error control method does not estimate π_1, but rather uses a projection of the number of correctly identified significant genes (s) to estimate FDR. Table 7 shows the approximate sample sizes required to achieve 80% power for the Yang et al.

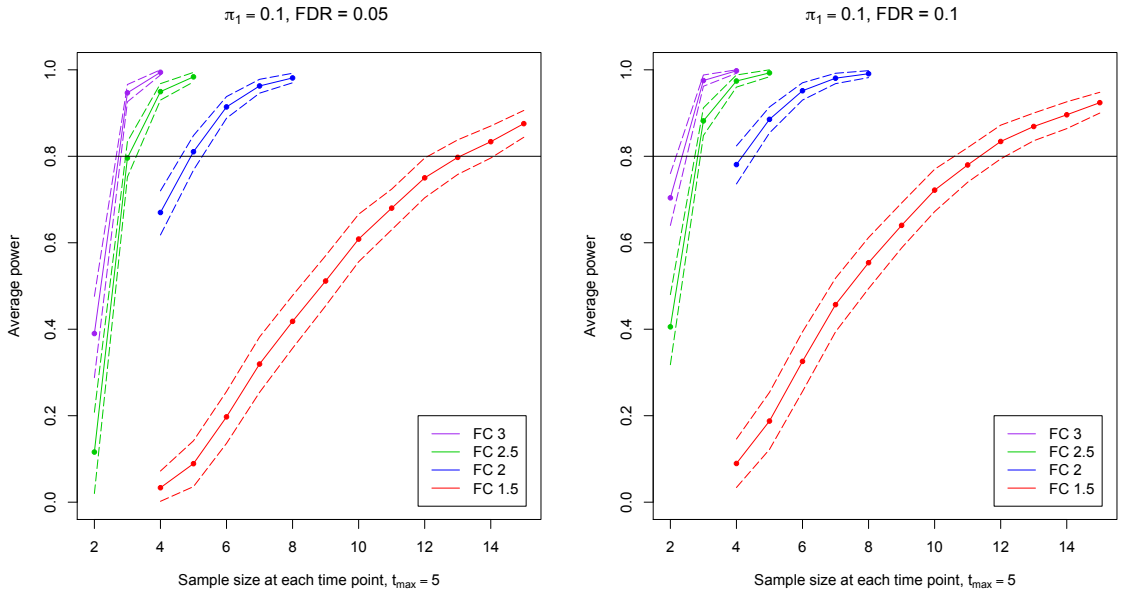

Figure 8: Two groups, time trend: Average power versus sample size, where $t_{\max} = 5$. FDR = 0.05 (left) and FDR = 0.10 (right). Dashed lines are 2.5 and 97.5 percentiles.

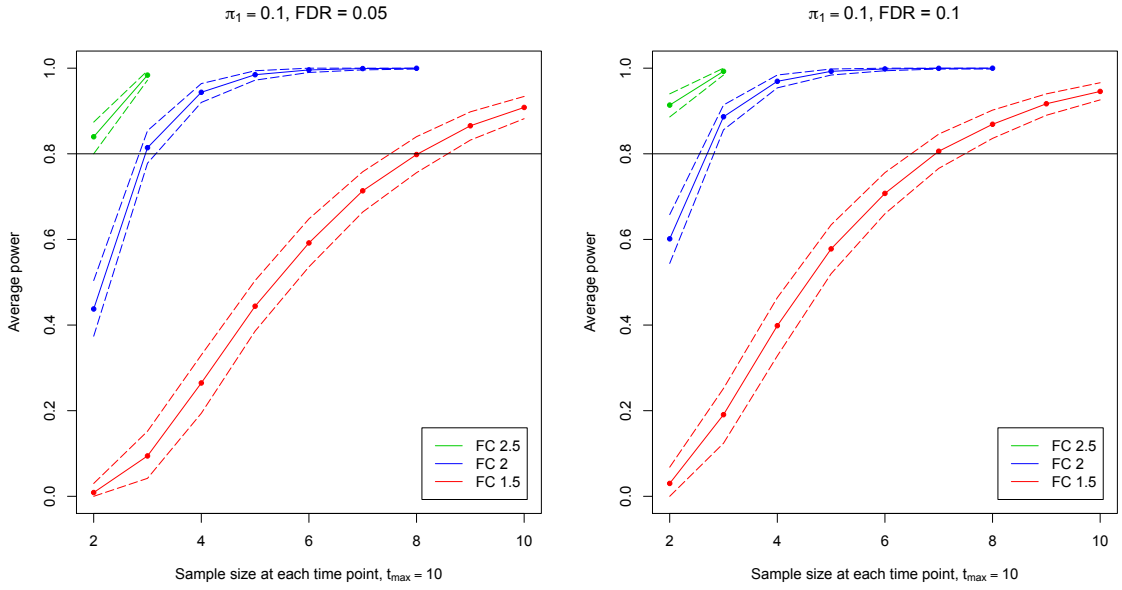

Figure 9: Two groups, time trend: Average power versus sample size, where $t_{\max} = 10$. FDR = 0.05 (left) and FDR = 0.10 (right). Dashed lines are 2.5 and 97.5 percentiles.

Table 6: Two group, single time point. Comparison of Jung's (2005) method using Equation 7 with simulation results. Sample size per group necessary to achieve 80% average power.

		$f = 0.05$		$f = 0.10$	
π_1	FC (effect size)	Jung's method	Simulation	Jung's method	Simulation
0.01	1.5 (1.85)	23		21	
	2.0 (3.16)	8		7	
	2.5 (4.18)	5		4	5
	3.0 (5.01)	4	5	3	5
0.05	1.5 (1.85)	18		16	
	2.0 (3.16)	7		6	5
	2.5 (4.18)	4	5	4	4
	3.0 (5.01)	3	5	3	4
0.10	1.5 (1.85)	16		14	
	2.0 (3.16)	6	5	5	5
	2.5 (4.18)	4	5	3	4
	3.0 (5.01)	3	4	2	4
0.20	1.5 (1.85)	14		12	
	2.0 (3.16)	5	5	4	4
	2.5 (4.18)	3	4	3	4
	3.0 (5.01)	2	4	2	4

method when desired FDR is 0.05, the number of replicates per sampling unit is equal to one, and the projected numbers of correctly identified significant genes are $s = 0$ (equivalent to the Bonferroni correction) and $s = 50$. Direct comparison is difficult however as it is not clear what the true value of π_1 was for their simulation. Smaller sample size for the present simulation study is expected to result partially from the more powerful method of type I error control. For Yang et al., increasing the number of time points from 5 to 10 also results in a slight increase in power, however for their method the required sample size per group in the two group case is larger than the required sample size per time point in the one group case, which is the opposite of the result for the present simulation.

Table 7: One and two group time trend cases. Comparison of Yang et al. (2003) method with simulation results. Sample size for each time point (and per group, for two group case) necessary to achieve 80% average power.

	Yang et al. $s = 0$	Yang et al. $s = 50$	Simulation
One group, time trend			
$t_{\max} = 5$	15	11	9
Two groups, time trend			
$t_{\max} = 5$	27	18	5
$t_{\max} = 10$	23	15	3

For this initial study, simple data and a simple linear model was chosen. Future studies involving more realistic microarray data would be interesting. This includes exploring the results of heterogeneity of true mean differences and variances for gene expression. The variance of 0.1 used in these simulations was perhaps less conservative than would be desirable for a sample size estimate, and it would be useful to try simulations with smaller effect sizes, e.g. $\Delta/\sigma = 1$.

Simulations were done assuming gene independence within and across time points. Another natural extension of this work would be to test the method under conditions of gene dependence. This may be carried out in a manner analogous to that used in Jung (2005), for example by using 500 blocks of 10 genes having a within group correlation coefficient of 0.6 simulated from the appropriate normal distributions. Alternatively, a more realistic correlation structure may be used according to the procedure proposed in Hardin et al. (2011). Longitudinal study designs incorporating correlation between time points could also be explored.

Finally, a linear trend is only one possible pattern of interest for gene expression over time. Many cellular processes are cyclical, for example, so their associated gene expression levels follow a periodic trend. The sample size method described here

can be generalized by simulating time course data with e.g. a periodic trend and fitting a more flexible model to time course data, e.g. a cubic spline or polynomial model. For example Storey et al. (2005) propose a regression goodness-of-fit statistic for time course microarray experiments using natural splines whose null distribution is estimated via a bootstrap method, while for the same statistic Sohn et al. (2009) use a permutation method for estimating the null distribution.

6 Summary and Conclusions

Determining sample size for time course microarray experiments is a challenging task since many parameters are involved and there may be few expectations regarding the temporal trend for each gene. Simulation is a useful approach for estimating power and sample size where there may be some expected trend, or at least to give some idea of the range of necessary sample size for a variety of different gene expression outcomes. The simulation method proposed here is flexible and could be adapted to a number of different settings.

7 References

Allison, D. B., Cui, X., Page, G. P., & Sabripour, M. (2006). Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews Genetics*, *7*(1), 55–65.
URL http://www.ncbi.nlm.nih.gov/pubmed/16369572

Androulakis, I. P., Yang, E., & Almon, R. R. (2007). Analysis of time-series gene expression data: methods, challenges, and opportunities. *Annual Review of Biomedical Engineering*, *9*, 205–28.
URL http://www.ncbi.nlm.nih.gov/pubmed/17341157

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *57*(1), 289–300.

Benjamini, Y., Krieger, A. M., & Yekutieli, D. (2006). Adaptive linear step-up procedures that control the false discovery rate. *Biometrika*, *93*(3), 491–507.
URL http://biomet.oxfordjournals.org/cgi/doi/10.1093/biomet/93.3.491

Diggle, P. J., Heagerty, P., Liang, K.-Y., & Zeger, S. L. (2002). *Analysis of Longitudinal Data*. New York, NY: Oxford University Press, 2nd ed.

Efron, B. (2004). Large-Scale Simultaneous Hypothesis Testing. *Journal of the American Statistical Association*, *99*(465), 96–104.
URL http://www.tandfonline.com/doi/abs/10.1198/016214504000000089

Efron, B. (2007). Correlation and Large-Scale Simultaneous Significance Testing. *Journal of the American Statistical Association*, *102*(477), 93–103.
URL http://pubs.amstat.org/doi/abs/10.1198/016214506000001211

Efron, B. (2012). *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. New York, NY: Cambridge University Press.

Fisher, R. A. (1966). *The Design of Experiments*. New York: Hafner, 8th ed.

Hardin, J., Garcia, S. R., & Golan, D. (2011). A method for generating realistic correlation matrices. *Unpublished manuscript*, (p. 36).
URL http://arxiv.org/abs/1106.5834

Hardin, J., & Wilson, J. (2009). A note on oligonucleotide expression values not being normally distributed. *Biostatistics Oxford England*, *10*(3), 446–50.
URL http://www.ncbi.nlm.nih.gov/pubmed/19276243

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer, 2nd ed.

Jung, S.-H. (2005). Sample size for FDR-control in microarray data analysis. *Bioinformatics (Oxford, England)*, *21*(14), 3097–104.
URL http://www.ncbi.nlm.nih.gov/pubmed/15845654

Kerr, M. K., & Churchill, G. A. (2001). Statistical design and the analysis of gene expression microarray data. *Genetical Research*, *77*, 123–128.
URL http://www.ncbi.nlm.nih.gov/pubmed/18976541

Kim, K. I., & van de Wiel, M. A. (2008). Effects of dependence in high-dimensional multiple testing problems. *BMC Bioinformatics*, *9*, 114.
URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2375137\&tool=pmcentrez\&rendertype=abstracthttp://www.biomedcentral.com/1471-2105/9/114/

Larson, N. B. (2011). *Investigation and development of statistical methods for gene expression data analysis*. Ph.D. thesis, Iowa State University.

Lee, M.-L. T., & Whitmore, G. A. (2002). Power and sample size for DNA microarray studies. *Statistics in Medicine*, *21*(23), 3543–70.
URL http://www.ncbi.nlm.nih.gov/pubmed/12436455

Li, S. S., Bigler, J., Lampe, J. W., Potter, J. D., & Feng, Z. (2005). FDR-controlling testing procedures and sample size determination for microarrays. *Statistics in Medicine*, *24*(15), 2267–80.
URL http://www.ncbi.nlm.nih.gov/pubmed/15977294

Lin, W.-J., Hsueh, H.-M., & Chen, J. J. (2010). Power and sample size estimation in microarray studies. *BMC Bioinformatics*, *11*, 48.
URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2837028\&tool=pmcentrez\&rendertype=abstract

Liu, P., & Hwang, J. T. G. (2007). Quick calculation for sample size while controlling false discovery rate with application to microarray analysis. *Bioinformatics (Oxford, England)*, *23*(6), 739–46.
URL http://www.ncbi.nlm.nih.gov/pubmed/17237060

Lockhart, D. J., & Winzeler, E. a. (2000). Genomics, gene expression and DNA arrays. *Nature*, *405*(6788), 827–36.
URL http://www.ncbi.nlm.nih.gov/pubmed/10866209

Nguyen, D. V., Senturk, D., Harvey, D. J., & Li, C.-S. (2008). Sample size calculation and power in genomics studies. In A. S. Russe (Ed.) *Computational Biology: New Research*, 530, chap. 1, (pp. 1–29). New York: Nova Science Publishers.
URL http://dnguyen.ucdavis.edu/.html/B3_2009Nguyen-P.pdf

Pounds, S., & Cheng, C. (2005). Sample size determination for the false discovery rate. *Bioinformatics (Oxford, England)*, *21*(23), 4263–71.
URL http://www.ncbi.nlm.nih.gov/pubmed/16204346

R Core Team (2013). *R: A Language and Environment for Statistical Computing, reference index version 3.0.0*. R Foundation for Statistical Computing, Vienna, Austria.
URL http://www.R-project.org/

Reiner, A., Yekutieli, D., & Benjamini, Y. (2003). Identifying differentially expressed genes using false discovery rate controlling procedures. *Bioinformatics*, *19*(3), 368–375.
URL http://bioinformatics.oxfordjournals.org/content/19/3/368.short

Sohn, I., Owzar, K., George, S. L., Kim, S., & Jung, S.-H. (2009). A permutation-based multiple testing method for time-course microarray experiments. *BMC Bioinformatics*, *10*, 336.
URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2772858\&tool=pmcentrez\&rendertype=abstract

Storey, J. D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *64*(3), 479–498.
URL http://doi.wiley.com/10.1111/1467-9868.00346

Storey, J. D. (2003). The positive false discovery rate: a Bayesian interpretation and the q-value. *The Annals of Statistics*, *31*(6), 2013–2035.
URL http://www.jstor.org/stable/10.2307/3448445

Storey, J. D., & Tibshirani, R. (2003a). SAM Thresholding and False Discovery Rates for Detecting Differential Gene Expression. In G. Parmigiani, E. S. Garrett, R. A. Irizarry, & S. L. Zeger (Eds.) *Statistics for Biology and Health: The Analysis of Gene Expression Data*, chap. 12, (pp. 272–290). New York: Springer.
URL http://dx.doi.org/10.1007/0-387-21679-0_12

Storey, J. D., & Tibshirani, R. (2003b). Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, *100*(16), 9440–5.
URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=170937\&tool=pmcentrez\&rendertype=abstract

Storey, J. D., Xiao, W., Leek, J. T., Tompkins, R. G., & Davis, R. W. (2005). Significance analysis of time course microarray experiments. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(36), 12837–42.
URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1201697\&tool=pmcentrez\&rendertype=abstract

Tai, Y. C., & Speed, T. P. (2005). Statistical Analysis of Microarray Time Course Data. In U. A. Nuber (Ed.) *DNA Microarrays*, chap. 20, (pp. 257–280). New York, NY: Taylor & Francis Group.

Tibshirani, R. (2006). A simple method for assessing sample sizes in microarray experiments. *BMC Bioinformatics*, *7*, 106.
URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1450307\&tool=pmcentrez\&rendertype=abstract`

Tsai, C.-A., Wang, S.-J., Chen, D.-T., & Chen, J. J. (2005). Sample size for gene expression microarray experiments. *Bioinformatics (Oxford, England)*, *21*(8), 1502–8.
URL `http://www.ncbi.nlm.nih.gov/pubmed/15564298`

Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(9), 5116–5121.
URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=33173\&tool=pmcentrez\&rendertype=abstract`

Wang, S.-J., & Chen, J. J. (2004). Sample size for identifying differentially expressed genes in microarray experiments. *Journal of Computational Biology*, *11*(4), 714–26.
URL `http://www.ncbi.nlm.nih.gov/pubmed/15579240`

Yang, M. C. K., Yang, J. J., McIndoe, R. A., & She, J. X. (2003). Microarray experimental design: power and sample size considerations. *Physiological Genomics*, *16*, 24–8.
URL `http://www.ncbi.nlm.nih.gov/pubmed/14532333`

Yang, Y. H., & Speed, T. (2002). Design issues for cDNA microarray experiments. *Nature Reviews Genetics*, *3*(8), 579–88.
URL `http://www.ncbi.nlm.nih.gov/pubmed/12154381`

8 Appendix

8.1 Example microarray dataset

8.1.1 Gene expression distributions

Table 8: Example microarray dataset: Gene expression distribution summary (after preprocessing)

Disease diagnosis	n	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Normal	7	1.184	3.386	4.829	4.873	6.177	14.750
Graves'	15	1.143	3.415	4.814	4.866	6.141	14.840
NSOI	34	1.058	3.358	4.823	4.873	6.195	14.740
Sarcoidosis	14	1.174	3.316	4.801	4.861	6.197	14.820
Wegener's	5	1.181	3.256	4.774	4.855	6.219	14.680

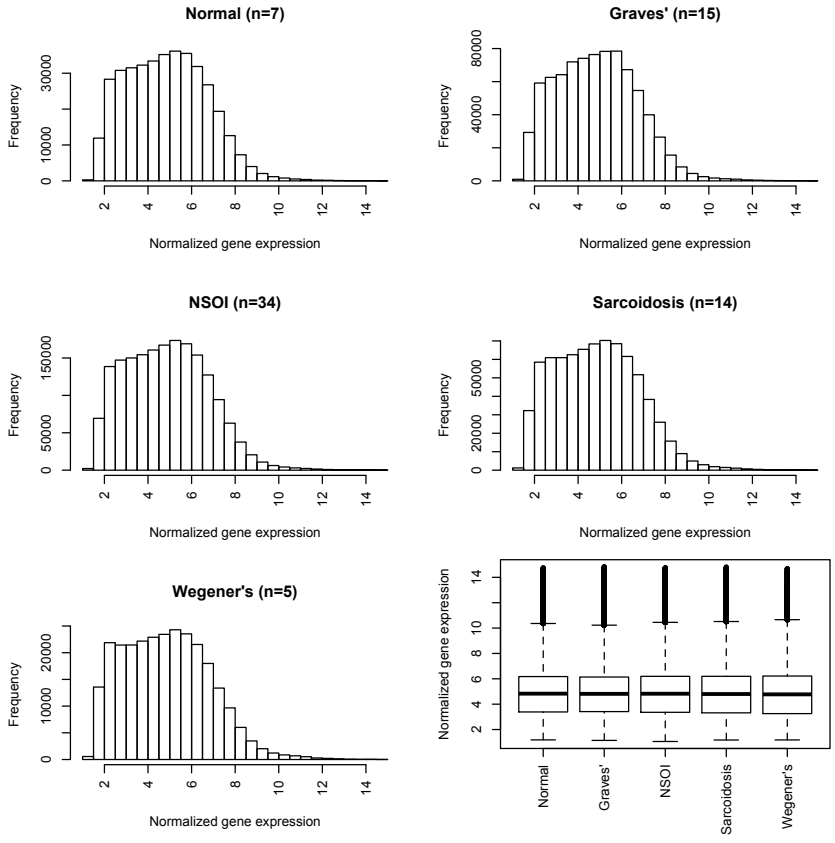

Figure 10: Normalized gene expression distributions

8.1.2 Variance distributions

Table 9: Example microarray dataset: Sample variance* distribution summary (after preprocessing)

Disease diagnosis	n	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Normal	7	0.000585	0.048050	0.084950	0.175300	0.155700	18.27
Graves'	15	0.006298	0.133200	0.266800	0.439800	0.530900	24.16
NSOI	34	0.004688	0.076190	0.116400	0.239500	0.201000	19.59
Sarcoidosis	14	0.003559	0.066090	0.109600	0.246300	0.208000	24.65
Wegener's	5	0.000033	0.023110	0.050800	0.180900	0.122600	23.87

*$s_j^2 = \sum_{i=1}^{n}(x_i - \bar{x})^2/(n-1)$

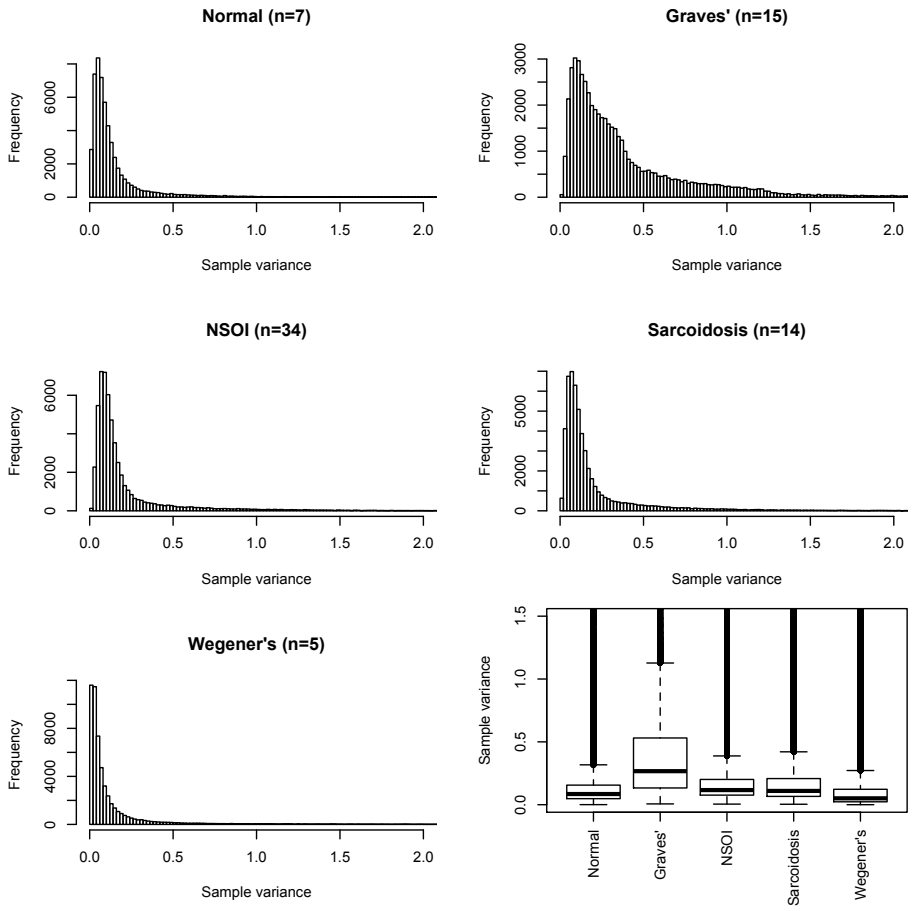

Figure 11: Sample variance distributions

8.2 Results tables

Table 10: Two groups, single time point, $m = 5000$, $m_1 = 50, 250$ ($\pi_1 = 0.01, 0.05$). Mean (sd) values after 1000 simulations, for desired FDR $f = 0.05$ and $f = 0.10$.

			$f = 0.05$					$f = 0.10$				
m_1	FC	n_i	s	r	Power (%)	Est. FDR (%)	True FDR (%)	s	r	Power (%)	Est. FDR (%)	True FDR (%)
50	1.5	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			<0.1 (0.1)	0.1 (0.4)	<0.1 (0.2)		
		4	<0.1 (0.2)	0.1 (0.3)	0.1 (0.4)			0.1 (0.3)	0.2 (0.5)	0.2 (0.7)		
		5	0.2 (0.5)	0.2 (0.6)	0.3 (0.9)			0.4 (0.8)	0.6 (1.1)	0.8 (1.7)		
	2.0	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			0.1 (0.3)	0.2 (0.5)	0.2 (0.6)		
		4	0.3 (0.8)	0.4 (1.0)	0.7 (1.7)			1.5 (2.1)	1.8 (2.5)	3.0 (4.2)		
		5	7.8 (5.1)	8.2 (5.4)	15.5 (10.2)			18.1 (6.0)	20.2 (7.1)	36.3 (12.1)		
	2.5	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			0.2 (0.5)	0.3 (0.7)	0.4 (1.1)		
		4	2.3 (2.9)	2.4 (3.0)	4.6 (5.8)			13.0 (7.0)	14.3 (7.9)	25.9 (14.0)		
		5	33.4 (4.7)	35 (5.2)	66.9 (9.3)	4.66 (0.34)	4.36 (0.03)	42.3 (3.2)	46.8 (4.5)	84.5 (6.5)	9.22 (0.68)	9.47 (4.30)
	3.0	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			0.4 (0.8)	0.5 (0.9)	0.8 (1.6)		
		4	9.5 (6.7)	9.8 (7.0)	19.0 (13.5)			31.0 (5.8)	33.8 (6.8)	62.0 (11.5)		
		5	45.4 (2.3)	47.5 (2.9)	90.7 (4.6)	4.51 (0.41)	4.36 (0.03)	48.5 (1.3)	53.6 (3.0)	96.9 (2.6)	8.83 (1.05)	9.43 (4.05)
250	1.5	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			0.1 (0.3)	0.2 (0.5)	<0.1 (0.1)		
		4	0.2 (0.6)	0.2 (0.7)	0.1 (0.2)			0.8 (1.5)	1.1 (1.9)	0.3 (0.6)		
		5	2.0 (2.7)	2.2 (2.9)	0.8 (1.1)			10.2 (7.7)	11.5 (8.9)	4.1 (3.1)		
	2.0	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			0.5 (0.9)	0.7 (1.0)	0.2 (0.4)		
		4	9.7 (9.0)	10.1 (9.5)	3.9 (3.6)			87.3 (17.5)	95.2 (20.0)	34.9 (7.0)	9.85 (0.16)	7.94 (2.96)
		5	154.0 (11.4)	161.2 (12.6)	61.6 (4.6)	4.92 (0.08)	4.42 (0.02)	202.6 (7.9)	224.3 (10.6)	81.0 (3.1)	9.85 (0.15)	9.62 (2.04)
	2.5	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			1.5 (1.9)	1.6 (2.0)	0.6 (0.8)		
		4	109.3 (18.0)	112.2 (18.8)	43.7 (7.2)	4.95 (0.06)	2.55 (0.02)	205.2 (8.3)	222.8 (10.9)	82.1 (3.3)	9.88 (0.11)	7.82 (1.91)
		5	235.6 (4.0)	246.6 (5.6)	94.2 (1.6)	4.85 (0.14)	4.45 (0.01)	245.9 (2.2)	272.5 (6.3)	98.4 (0.9)	9.72 (0.27)	9.73 (1.85)
	3.0	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			2.9 (3.7)	3.0 (3.8)	1.2 (1.5)		
		4	191.9 (10.5)	196.8 (11.4)	76.7 (4.2)	4.95 (0.05)	2.5 (0.01)	239.0 (3.6)	259.4 (7.0)	95.6 (1.5)	9.81 (0.19)	7.83 (1.80)
		5	248.2 (1.3)	259.9 (3.8)	99.3 (0.5)	4.72 (0.27)	4.48 (0.01)	249.7 (0.5)	277.0 (5.7)	99.9 (0.2)	9.66 (0.34)	9.79 (1.84)

m_1: total number of differentially expressed genes; FC: fold change; n_i: sample size per group; s: number of genes correctly declared significant; r: total number of genes declared significant; Power: $s/m_1 \times 100$; Est. FDR: estimated FDR; True FDR: true FDR

Table 11: Two groups, single time point, $m = 5000$, $m_1 = 500, 1000$ ($\pi_1 = 0.1, 0.2$). Mean (sd) values after 1000 simulations, for desired FDR $f = 0.05$ and $f = 0.10$.

			$f = 0.05$					$f = 0.10$				
m_1	FC	n_i	s	r	Power (%)	Est. FDR (%)	True FDR (%)	s	r	Power (%)	Est. FDR (%)	True FDR (%)
500	1.5	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			0.2 (0.5)	0.3 (0.6)	<0.1 (0.1)		
		4	0.5 (1.2)	0.6 (1.3)	0.1 (0.2)			4.6 (5.8)	5.3 (6.6)	0.9 (1.2)		
		5	14.3 (11.1)	15.1 (11.8)	2.9 (2.2)			81.8 (22.4)	90.6 (25.6)	16.4 (4.5)	9.82 (0.23)	9.42 (3.21)
	2.0	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			1.3 (2.0)	1.5 (2.1)	0.3 (0.4)		
		4	107.0 (30.5)	110.3 (31.7)	21.4 (6.1)	4.95 (0.05)	2.85 (0.02)	328.2 (17.5)	358.2 (21.8)	65.6 (3.5)	9.94 (0.06)	8.31 (1.59)
		5	400.4 (11.7)	419.8 (13.5)	80.1 (2.3)	4.95 (0.05)	4.60 (0.01)	458.9 (7.2)	508.2 (11.9)	91.8 (1.4)	9.90 (0.11)	9.68 (1.39)
	2.5	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			5.9 (7.6)	6.1 (7.9)	1.2 (1.5)		
		4	369.0 (16.5)	379.7 (18.1)	73.8 (3.3)	4.97 (0.03)	2.79 (0.01)	471.3 (6.0)	515.0 (11.0)	94.3 (1.2)	9.91 (0.08)	8.47 (1.33)
		5	491.3 (3.1)	515.6 (6.3)	98.3 (0.6)	4.89 (0.10)	4.72 (0.01)	498.1 (1.3)	553.0 (8.5)	99.6 (0.3)	9.85 (0.15)	9.90 (1.35)
	3.0	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			22.2 (23.6)	22.7 (24.2)	4.4 (4.7)		
		4	462.1 (7.7)	475.4 (9.4)	92.4 (1.5)	4.96 (0.04)	2.78 (0.01)	495.7 (2.2)	541.8 (8.4)	99.1 (0.4)	9.87 (0.13)	8.50 (1.29)
1000	1.5	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			0.5 (1.0)	0.6 (1.2)	0.1 (0.1)		
		4	4.0 (5.5)	4.2 (5.8)	0.4 (0.5)			97.9 (37.5)	107.7 (41.9)	9.8 (3.7)		
		5	168.3 (33.3)	176.4 (35.5)	16.8 (3.3)	4.96 (0.05)	4.51 (0.02)	428.8 (29.2)	474.4 (34.9)	42.9 (2.9)	9.96 (0.04)	9.57 (1.45)
	2.0	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			18.9 (21.2)	19.8 (22.4)	1.9 (2.1)		
		4	610.2 (29.4)	631.8 (32.1)	61.0 (2.9)	4.99 (0.01)	3.41 (0.01)	879.7 (13.4)	970.8 (20.2)	88.0 (1.3)	9.96 (0.04)	9.36 (1.01)
		5	921.6 (10.1)	967.3 (13.6)	92.2 (1.0)	4.97 (0.04)	4.72 (0.01)	974.2 (5.6)	1079.5 (14.0)	97.4 (0.6)	9.93 (0.07)	9.74 (0.97)
	2.5	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			312.3 (87.0)	322.0 (90.9)	31.2 (8.7)	9.98 (0.03)	2.89 (1.05)
		4	930.6 (10.1)	964.5 (13.4)	93.1 (1.0)	4.98 (0.02)	3.51 (0.01)	990.5 (3.1)	1097.7 (12.8)	99.1 (0.3)	9.94 (0.06)	9.75 (0.97)
	3.0	3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)			659.4 (54.6)	679.2 (58.6)	65.9 (5.5)	9.99 (0.01)	2.89 (0.74)
		4	988.8 (3.6)	1025.0 (8.1)	98.9 (0.4)	4.95 (0.05)	3.53 (0.01)	999.3 (0.8)	1109.6 (12.3)	99.9 (0.1)	9.93 (0.07)	9.93 (0.99)
		5	999.9 (0.3)	1054.0 (7.9)	100.0 (0)	4.91 (0.09)	5.13 (0.01)	1000.0 (0.1)	1113.0 (12.0)	100.0 (0)	9.91 (0.09)	10.14 (0.97)

m_1: total number of differentially expressed genes; FC: fold change; n_i: sample size per group; s: number of genes correctly declared significant; r: total number of genes declared significant; Power: $s/m_1 \times 100$; Est. FDR: estimated FDR; True FDR: true FDR

Table 12: One group, time trend, $t_{\max} = 5$, $m = 5000$, $m_1 = 500$ ($\pi_1 = 0.1$).
Mean (sd) values after 1000 simulations, for desired FDR $f = 0.05$ and $f = 0.10$.

		$f = 0.05$					$f = 0.10$				
FC	n_t	s	r	Power (%)	Est. FDR (%)	True FDR (%)	s	r	Power (%)	Est. FDR (%)	True FDR (%)
1.5	4	8.1 (7.2)	8.2 (7.2)	1.6 (1.4)			69.2 (18.7)	71.3 (19.5)	13.8 (3.7)	9.80 (0.21)	2.73 (2.07)
	5	19.1 (11.8)	19.2 (11.8)	3.8 (2.4)			136.3 (19.8)	138.8 (20.5)	27.3 (4.0)	9.90 (0.11)	1.77 (1.12)
	6	35.6 (16.2)	35.6 (16.2)	7.1 (3.2)			195.0 (18.1)	197.3 (18.6)	39.0 (3.6)	9.92 (0.08)	1.15 (0.80)
	7	53.6 (17.7)	53.7 (17.8)	10.7 (3.5)	4.89 (0.14)	0.05 (<0.01)	243.5 (17.1)	245.4 (17.4)	48.7 (3.4)	9.93 (0.07)	0.74 (0.55)
	8	71.6 (18.8)	71.6 (18.8)	14.3 (3.8)	4.92 (0.09)	0.02 (<0.01)	279.9 (16.2)	281.2 (16.4)	56.0 (3.2)	9.93 (0.06)	0.45 (0.40)
	9	92.3 (20.1)	92.3 (20.1)	18.5 (4.0)	4.94 (0.06)	0.01 (<0.01)	309.5 (15.3)	310.4 (15.4)	61.9 (3.1)	9.94 (0.06)	0.27 (0.30)
	10	108.9 (20.9)	108.9 (20.9)	21.8 (4.2)	4.95 (0.06)	0.01 (<0.01)	332.8 (13.8)	333.4 (13.9)	66.6 (2.8)	9.94 (0.06)	0.17 (0.24)
	11	127.0 (20.6)	127.0 (20.6)	25.4 (4.1)	4.95 (0.05)	<0.01 (<0.01)	352.3 (13.4)	352.7 (13.5)	70.5 (2.7)	9.94 (0.06)	0.10 (0.16)
	12	145.2 (21.0)	145.2 (21.0)	29.0 (4.2)	4.96 (0.05)	<0.01 (<0.01)	369.4 (12.4)	369.7 (12.4)	73.9 (2.5)	9.93 (0.07)	0.06 (0.13)
	13	161.3 (20.5)	161.3 (20.5)	32.3 (4.1)	4.96 (0.04)	<0.01 (<0.01)	382.9 (12.0)	383.1 (12.0)	76.6 (2.4)	9.93 (0.07)	0.03 (0.09)
	14	178.2 (20.4)	178.2 (20.4)	35.6 (4.1)	4.96 (0.03)	<0.01 (<0.01)	395.3 (11.2)	395.4 (11.3)	79.1 (2.2)	9.93 (0.06)	0.02 (0.07)
	15	193.3 (20.3)	193.3 (20.3)	38.7 (4.1)	4.97 (0.04)	<0.01 (<0.01)	405.7 (10.2)	405.7 (10.2)	81.1 (2.0)	9.93 (0.07)	0.01 (0.05)
	16	207.5 (20.4)	207.5 (20.4)	41.5 (4.1)	4.97 (0.03)	<0.01 (<0.01)	414.7 (10.5)	414.7 (10.6)	82.9 (2.1)	9.93 (0.06)	0.01 (0.04)
	20	256.6 (18.9)	256.6 (18.9)	51.3 (3.8)	4.97 (0.03)	<0.01 (<0.01)	442.0 (8.3)	442.0 (8.3)	88.4 (1.7)	9.92 (0.08)	<0.01 (0.02)
	25	304.3 (16.8)	304.3 (16.8)	60.9 (3.4)	4.97 (0.03)	<0.01 (<0.01)	462.4 (6.5)	462.4 (6.5)	92.5 (1.3)	9.90 (0.11)	<0.01 (<0.01)
	30	340.1 (15.4)	340.1 (15.4)	68.0 (3.1)	4.97 (0.03)	<0.01 (<0.01)	475.1 (5.0)	475.1 (5.0)	95.0 (1.0)	9.87 (0.13)	<0.01 (<0.01)
	40	389.9 (12.0)	389.9 (12.0)	78.0 (2.4)	4.97 (0.03)	<0.01 (<0.01)	488.4 (3.6)	488.4 (3.6)	97.7 (0.7)	9.80 (0.18)	<0.01 (<0.01)
	50	421.2 (10.6)	421.2 (10.6)	84.2 (2.1)	4.97 (0.03)	<0.01 (<0.01)	494.3 (2.4)	494.3 (2.4)	98.9 (0.5)	9.71 (0.26)	<0.01 (<0.01)
2.0	4	239.2 (20.2)	239.5 (20.2)	47.8 (4.0)	4.97 (0.03)	0.13 (<0.01)	414.3 (9.9)	418.9 (10.4)	82.9 (2.0)	9.92 (0.08)	1.10 (0.51)
	5	294.9 (16.2)	295.0 (16.2)	59.0 (3.2)	4.97 (0.03)	0.03 (<0.01)	449.6 (7.6)	451.4 (7.8)	89.9 (1.5)	9.91 (0.09)	0.40 (0.29)
	6	333.0 (14.9)	333.0 (14.9)	66.6 (3.0)	4.97 (0.03)	0.01 (<0.01)	467.0 (6.2)	467.6 (6.3)	93.4 (1.2)	9.88 (0.11)	0.13 (0.17)
	7	362.1 (13.4)	362.1 (13.4)	72.4 (2.7)	4.97 (0.03)	<0.01 (<0.01)	477.4 (4.9)	477.7 (4.9)	95.5 (1.0)	9.85 (0.14)	0.04 (0.10)
	8	384.2 (12.3)	384.2 (12.4)	76.8 (2.5)	4.97 (0.03)	<0.01 (<0.01)	484.0 (4.1)	484.1 (4.1)	96.8 (0.8)	9.82 (0.18)	0.02 (0.06)
	9	402.3 (11.3)	402.3 (11.3)	80.5 (2.3)	4.97 (0.03)	<0.01 (<0.01)	488.5 (3.5)	488.5 (3.5)	97.7 (0.7)	9.77 (0.21)	<0.01 (0.03)
2.5	4	413.9 (10.2)	414.1 (10.3)	82.8 (2.0)	4.97 (0.03)	0.05 (<0.01)	488.0 (3.8)	490.4 (4.2)	97.6 (0.8)	9.77 (0.21)	0.49 (0.32)
	5	441.8 (8.3)	441.8 (8.3)	88.4 (1.7)	4.96 (0.04)	0.01 (<0.01)	494.8 (2.3)	495.4 (2.5)	99.0 (0.5)	9.64 (0.33)	0.11 (0.15)
3.0	4	472.5 (5.8)	472.6 (5.8)	94.5 (1.2)	4.94 (0.06)	0.02 (<0.01)	498.5 (1.3)	499.8 (1.7)	99.7 (0.3)	9.37 (0.54)	0.27 (0.23)
	5	483.9 (4.1)	483.9 (4.1)	96.8 (0.8)	4.91 (0.08)	<0.01 (<0.01)	499.5 (0.7)	499.8 (0.8)	99.9 (0.1)	8.71 (0.86)	0.04 (0.09)

m_1: total number of differentially expressed genes; FC: fold change; n_t: sample size at each time point; s: number of genes correctly declared significant; r: total number of genes declared significant; Power: $s/m_1 \times 100$; Est. FDR: estimated FDR; True FDR: true FDR

Table 13: Two group, time trend, $t_{\max} = 5$, $m = 5000$, $m_1 = 500$ ($\pi_1 = 0.1$).
Mean (sd) values after 1000 simulations, for desired FDR $f = 0.05$ and $f = 0.10$.

		$f = 0.05$					$f = 0.10$				
FC	n_{it}	s	r	Power (%)	Est. FDR (%)	True FDR (%)	s	r	Power (%)	Est. FDR (%)	True FDR (%)
1.5	4	16.7 (9.4)	17.7 (10.1)	3.3 (1.9)			44.8 (14.6)	49.8 (16.8)	9.0 (2.9)		
	5	44.5 (13.5)	46.6 (14.4)	8.9 (2.7)	4.77 (0.27)	4.15 (0.03)	93.8 (17.2)	103.2 (19.8)	18.8 (3.4)	9.81 (0.19)	8.86 (2.94)
	6	98.7 (15.8)	103.6 (16.8)	19.7 (3.2)	4.89 (0.11)	4.74 (0.02)	162.8 (17.8)	180.1 (20.7)	32.6 (3.6)	9.87 (0.13)	9.50 (2.31)
	7	159.7 (16.1)	168.3 (17.6)	31.9 (3.2)	4.92 (0.08)	5.05 (0.02)	228.5 (15.9)	254.3 (19.7)	45.7 (3.2)	9.89 (0.11)	10.07 (2.07)
	8	209.1 (15.9)	219.8 (17.4)	41.8 (3.2)	4.93 (0.07)	4.82 (0.01)	276.9 (14.9)	306.6 (18.3)	55.4 (3.0)	9.90 (0.10)	9.62 (1.79)
	9	255.8 (14.8)	268.3 (16.4)	51.2 (3.0)	4.93 (0.06)	4.64 (0.01)	320.1 (13.2)	353.5 (16.5)	64.0 (2.6)	9.90 (0.10)	9.40 (1.63)
	10	304.3 (13.8)	319.8 (15.2)	60.9 (2.8)	4.93 (0.06)	4.85 (0.01)	360.9 (12.2)	400.1 (15.4)	72.2 (2.4)	9.90 (0.10)	9.77 (1.54)
	11	340.1 (12.2)	356.9 (13.9)	68.0 (2.4)	4.93 (0.07)	4.70 (0.01)	390.0 (10.7)	431.4 (14.4)	78.0 (2.1)	9.89 (0.10)	9.56 (1.48)
	12	375.1 (11.6)	395.0 (13.6)	75.0 (2.3)	4.93 (0.06)	5.01 (0.01)	417.1 (9.8)	464.1 (14.6)	83.4 (2.0)	9.89 (0.10)	10.08 (1.53)
	13	398.7 (10.3)	419.3 (12.3)	79.7 (2.1)	4.92 (0.08)	4.89 (0.01)	434.5 (8.4)	481.9 (12.7)	86.9 (1.7)	9.88 (0.11)	9.83 (1.41)
	14	416.9 (9.6)	436.9 (11.7)	83.4 (1.9)	4.92 (0.08)	4.57 (0.01)	448.0 (7.5)	494.3 (12.1)	89.6 (1.5)	9.87 (0.12)	9.35 (1.44)
	15	437.8 (8.1)	461.3 (10.4)	87.5 (1.6)	4.91 (0.09)	5.09 (0.01)	462.0 (6.2)	514.3 (11.1)	92.4 (1.2)	9.87 (0.13)	10.15 (1.44)
2.0	4	335.0 (13.4)	352.3 (15.1)	67.0 (2.7)	4.94 (0.06)	4.88 (0.01)	390.4 (11.3)	433.0 (15.1)	78.1 (2.3)	9.90 (0.10)	9.81 (1.49)
	5	405.3 (10.3)	422.3 (11.9)	81.1 (2.1)	4.93 (0.07)	4.01 (0.01)	442.7 (7.8)	483.7 (11.6)	88.5 (1.6)	9.87 (0.12)	8.46 (1.33)
	6	457.1 (6.6)	479.3 (8.8)	91.4 (1.3)	4.90 (0.09)	4.62 (0.01)	475.8 (4.9)	524.9 (9.9)	95.2 (1.0)	9.86 (0.14)	9.34 (1.34)
	7	481.3 (4.2)	507.5 (7.3)	96.3 (0.8)	4.88 (0.12)	5.15 (0.01)	490.3 (3.1)	546.3 (9.6)	98.1 (0.6)	9.84 (0.16)	10.22 (1.44)
	8	490.6 (3.0)	514.6 (6.4)	98.1 (0.6)	4.85 (0.14)	4.65 (0.01)	495.5 (2.2)	547.3 (8.5)	99.1 (0.4)	9.83 (0.17)	9.45 (1.33)
2.5	2	58.0 (24.1)	58.7 (24.5)	11.6 (4.8)			202.8 (21.2)	212.6 (23.2)	40.6 (4.2)	9.91 (0.09)	4.57 (1.47)
	3	398.0 (10.6)	415.3 (12.0)	79.6 (2.1)	4.93 (0.07)	4.14 (0.01)	440.9 (8.2)	483.6 (12.1)	88.2 (1.6)	9.89 (0.11)	8.79 (1.39)
	4	474.8 (5.1)	499.2 (7.4)	95.0 (1.0)	4.89 (0.11)	4.88 (0.01)	487.1 (3.5)	540.2 (9.2)	97.4 (0.7)	9.85 (0.15)	9.82 (1.37)
	5	491.9 (2.9)	511.0 (5.4)	98.4 (0.6)	4.83 (0.16)	3.74 (0.01)	496.4 (1.9)	539.9 (7.5)	99.3 (0.4)	9.80 (0.20)	8.03 (1.21)
3.0	2	195.0 (24.1)	197.1 (24.6)	39.0 (4.8)	4.95 (0.05)	1.02 (0.01)	352 (15.3)	368.7 (17.7)	70.4 (3.1)	9.93 (0.07)	4.52 (1.17)
	3	473.4 (5.3)	492.8 (7.2)	94.7 (1.1)	4.89 (0.11)	3.93 (0.01)	487.6 (3.7)	533.1 (8.8)	97.5 (0.7)	9.84 (0.15)	8.51 (1.31)
	4	497.1 (1.7)	522.7 (5.6)	99.4 (0.3)	4.82 (0.17)	4.87 (0.01)	498.9 (1.1)	553.2 (8.6)	99.8 (0.2)	9.82 (0.17)	9.81 (1.37)

m_1: total number of differentially expressed genes; FC: fold change; n_{it}: sample size per group at each time point; s: number of genes correctly declared significant; r: total number of genes declared significant; Power: $s/m_1 \times 100$; Est. FDR: estimated FDR; True FDR: true FDR

Table 14: Two group, time trend, $t_{\max} = 10$, $m = 5000$, $m_1 = 500$ ($\pi_1 = 0.1$).
Mean (sd) values after 1000 simulations, for desired FDR $f = 0.05$ and $f = 0.10$.

		$f = 0.05$					$f = 0.10$				
FC	n_{it}	s	r	Power (%)	Est. FDR (%)	True FDR (%)	s	r	Power (%)	Est. FDR (%)	True FDR (%)
1.5	2	4.4 (4.2)	4.7 (4.5)	0.9 (0.8)			15.1 (9.1)	16.7 (10.2)	3 (1.8)		
	3	47.2 (13.8)	49.5 (14.7)	9.4 (2.8)	4.80 (0.21)	4.55 (0.03)	95.3 (16.8)	105.3 (19.2)	19.1 (3.4)	9.79 (0.22)	9.32 (2.97)
	4	132.4 (17.0)	139.0 (18.5)	26.5 (3.4)	4.91 (0.08)	4.71 (0.02)	199.3 (17.2)	220.4 (20.4)	39.9 (3.4)	9.88 (0.12)	9.52 (2.11)
	5	222.0 (15.6)	233.4 (17.1)	44.4 (3.1)	4.93 (0.07)	4.84 (0.01)	289.0 (14.7)	320.2 (18.0)	57.8 (2.9)	9.90 (0.09)	9.70 (1.72)
	6	295.9 (14.3)	310.6 (15.7)	59.2 (2.9)	4.94 (0.07)	4.72 (0.01)	353.8 (12.4)	391.1 (16.0)	70.8 (2.5)	9.89 (0.10)	9.51 (1.56)
	7	356.8 (12.2)	375.2 (14.0)	71.4 (2.4)	4.93 (0.07)	4.89 (0.01)	403.0 (10.3)	446.7 (14.2)	80.6 (2.1)	9.89 (0.11)	9.75 (1.48)
	8	399.1 (10.8)	418.6 (12.9)	79.8 (2.2)	4.92 (0.08)	4.64 (0.01)	434.5 (8.5)	479.7 (12.8)	86.9 (1.7)	9.88 (0.12)	9.40 (1.39)
	9	432.7 (8.2)	454.6 (10.4)	86.5 (1.6)	4.92 (0.08)	4.81 (0.01)	458.4 (6.4)	508.0 (11.1)	91.7 (1.3)	9.88 (0.11)	9.73 (1.40)
	10	454.1 (6.9)	476.7 (9.1)	90.8 (1.4)	4.90 (0.10)	4.73 (0.01)	472.8 (5.4)	522.6 (10.1)	94.6 (1.1)	9.86 (0.13)	9.51 (1.36)
2.0	2	218.8 (16.5)	226.5 (17.5)	43.8 (3.3)	4.94 (0.07)	3.37 (0.01)	300.7 (14.5)	326.2 (17.5)	60.1 (2.9)	9.91 (0.09)	7.78 (1.51)
	3	407.1 (10.0)	425.5 (11.7)	81.4 (2.0)	4.93 (0.07)	4.30 (0.01)	443.2 (7.8)	487.2 (11.7)	88.6 (1.6)	9.88 (0.13)	9.00 (1.40)
	4	471.8 (5.5)	494.4 (7.7)	94.4 (1.1)	4.88 (0.12)	4.55 (0.01)	484.7 (3.9)	534.1 (9.2)	96.9 (0.8)	9.85 (0.15)	9.24 (1.34)
	5	492.4 (2.9)	516.7 (6.0)	98.5 (0.6)	4.85 (0.15)	4.70 (0.01)	496.3 (2.0)	548.4 (8.2)	99.3 (0.4)	9.83 (0.17)	9.48 (1.30)
	6	497.9 (1.4)	521.2 (5.4)	99.6 (0.3)	4.81 (0.20)	4.45 (0.01)	499.1 (1.0)	549.6 (8.1)	99.8 (0.2)	9.80 (0.20)	9.17 (1.33)
	7	499.5 (0.7)	524.0 (5.3)	99.9 (0.1)	4.80 (0.20)	4.65 (0.01)	499.8 (0.4)	552.3 (7.9)	100.0 (0.1)	9.81 (0.19)	9.48 (1.28)
	8	499.9 (0.3)	523.0 (5.3)	100.0 (0.1)	4.80 (0.20)	4.41 (0.01)	500.0 (0.2)	549.5 (7.7)	100.0 (<0.1)	9.80 (0.20)	9.00 (1.27)
2.5	2	419.9 (9.7)	433.3 (10.9)	84.0 (1.9)	4.93 (0.07)	3.09 (0.01)	456.9 (6.8)	493.3 (10.4)	91.4 (1.4)	9.87 (0.14)	7.36 (1.23)
	3	491.9 (2.9)	512.7 (5.7)	98.4 (0.6)	4.83 (0.16)	4.04 (0.01)	496.3 (2.0)	543.6 (8.1)	99.3 (0.4)	9.81 (0.18)	8.67 (1.30)

m_1: total number of differentially expressed genes; FC: fold change; n_{it}: sample size per group at each time point; s: number of genes correctly declared significant; r: total number of genes declared significant; Power: $s/m_1 \times 100$; Est. FDR: estimated FDR; True FDR: true FDR

8.3 Median computation times

Table 15: Computation times

	Median computing time	
	Seconds	Hours
Two groups, single time point		
$n_i = 3$, $B = 20$		
Amazon C1 HiCPU ExtraLarge	2758	0.77
MacBook Pro 2 GHz Intel Core i7	3120	0.87
$n_i = 4$, $B = 70$		
Amazon C1 HiCPU ExtraLarge	9482	2.63
Amazon M3 Double ExtraLarge	9500	2.64
MacBook Pro 2 GHz Intel Core i7	21060	5.85
$n_i = 5$, $B = 252$		
Dell Precision T7500 Intel Xeon 3.33 GHz ($\times 2$)	21365	5.93
One group, time trend		
$t_{\max} = 5$; $n = 4$ to $16, 20, 25, 30, 40, 50$; $B = 120$		
Dell Precision T7500 Intel Xeon 3.33 GHz ($\times 2$)	24774	6.88
Two groups, time trend		
$t_{\max} = 5$; $n = 2$ to 15; $B = 100$		
Dell Precision T7500 Intel Xeon 3.33 GHz ($\times 2$)	11138	3.09
$t_{\max} = 10$; $n = 2$ to 8; $B = 100$		
Dell Precision T7500 Intel Xeon 3.33 GHz ($\times 2$)	10593	2.94

8.4 R code

8.4.1 Two groups, single time point

```
# Two groups, single time point
# FDR and permutation-based simulation study
# For sample size and power estimation

# Parameters used:
# sample size n1 = n2 = ni = 3, 4, 5
# total genes m = 5000
# proportion differentially expressed genes pi1 = 0.01, 0.05, 0.1, 0.2
# fold change FC = 1.5, 2, 2.5, 3
# gene expression variance = 0.1
# pFDR level f = 0.05, 0.1
# tuning parameter for pFDR lambda = 0.5
# maximum number of permutations used B = 2ni choose ni = 20, 70, 252
# number of iterations for each set of parameters 1000

### Set working directory.
setwd("working directory")

### Load library snowfall (depends on snow) for parallel computing.
library(snowfall)

### Set values for global variables.
ni = 5
m = 5000
pi1 = 0.05
FC = 2.5

### Generate all possible permutations for given ni.
permutations = combn(2*ni, ni)
B = dim(permutations)[2]

### Create blocks for snowfall parallel execution (20 blocks of 50).
blocks = matrix(1:1000,nrow=20,byrow=TRUE)

### DEFINE FUNCTIONS ###
### Function to generate data.
generate.data = function(ni, pi1, FC, variance=0.1, m=5000){
    control.data = matrix(rnorm(m*ni, mean=0, sd=sqrt(variance)),
```

```
            nrow=m, ncol=ni)
    treatment.data = matrix(c(rnorm(pi1*m*ni, mean=log2(FC),
            sd=sqrt(variance)), rnorm((1-pi1)*m*ni, mean=0,
            sd=sqrt(variance))), nrow=m, ncol=ni, byrow=TRUE)
    data = cbind(control.data, treatment.data)
    return(data)
}

### Function to calculate t-statistics for two groups.
### Input is 2 gene expression matrices of dimension m*ni.
### Uses global variable ni.
t.stats = function(group.1,group.2){
    mean.1 = rowMeans(group.1)
    mean.2 = rowMeans(group.2)
    sd.1 = apply(group.1,1,sd)
    sd.2 = apply(group.2,1,sd)
    pooled.variance = ((ni-1)*sd.1^2+(ni-1)*sd.2^2)/(2*ni-2)
    pooled.sd = sqrt(pooled.variance)
    t = as.matrix((mean.1-mean.2)/(pooled.sd*sqrt(2/ni)))
    return(t)
}

### Function to apply positive FDR algorithm to obtain
### q-value for each gene. For 5000 genes.
qval = function(p.values, lambda=0.5) {
    ### Estimate number of null genes.
    m0hat = sum(p.values > lambda)/(lambda)
    pi0hat = m0hat/length(p.values)
    ### Sort p-values from largest to smallest.
    ordered.p.values =
        sort(p.values,decreasing=TRUE,index.return=TRUE)
    ### Value of largest p-value:
    p_m = ordered.p.values[[1]][1]
    ### Obtain q-value estimate for the largest p-value:
    q.p_m = ((m0hat*p_m)/sum(p.values <= p_m))
    q.values = matrix(NA,nrow=5000,ncol=1)
    q.values[1,] = q.p_m
    for (i in 2:m) {
        q.values[i,] = min(q.values[i-1,],
            ((m0hat*ordered.p.values[[1]][i])/
            sum(p.values <= ordered.p.values[[1]][i])))
    }
```

```
    return(q.values)
}

### Function to perform significance test and power calculation.
### Input q-values and desired FDR (f). Uses global variables pi1, m.
### Outputs s, r, power, estimated and true FDR.
test.power = function(q.values,p.values,f){
    q.val.index = which(q.values <= f)
    if (length(q.val.index)==0){
        s = 0
        r = 0
        power = 0
        estimated.FDR = 0
        true.FDR = 0
    } else {
        ordered.p.values =
            sort(p.values,decreasing=TRUE,index.return=TRUE)
        gene.index = ordered.p.values[[2]][q.val.index]
        s = sum(gene.index <= pi1*m)
        r = length(gene.index)
        power = s/(pi1*m)
        estimated.FDR = max(q.values[q.val.index])
        true.FDR = (r - s)/r
    }
    out = c(s,r,power,estimated.FDR,true.FDR)
    return(out)
}

### INITIALIZE SNOWFALL ###
### Enter number of cpus according to machine.
sfInit(parallel=TRUE,cpus=12,type="SOCK")
sfExportAll()   ### Export all objects in workspace to each node.

### Record start time.
time1 = proc.time()

### BEGIN ITERATIONS ###
output = sfApply(blocks,1,function(k){
    sapply(k,function(l){
        set.seed(l)
        ### Generate gene expression data.
        data = generate.data(ni, pi1, FC)
```

```
        ### Generate null distribution by permutation.
        perms.index = c(1:B)
        null.t.stats = sapply(perms.index,function(i){
            group.1 = data[,permutations[,i]]
            group.2 = data[,-permutations[,i]]
            return(t.stats(group.1,group.2))})
        ### Calculate the test statistics for the data.
        group.1 = data[,1:ni]
        group.2 = data[,(ni+1):(2*ni)]
        t = t.stats(group.1,group.2)
        ### Calculate raw p-values using pooled null test statistics.
        p.values = apply(t,1,function(x){(sum(abs(null.t.stats)
            >= abs(x)))/(B*m)})
        ### Positive FDR algorithm to obtain q-value for each gene.
        q.values = qval(p.values)
        ### Do significance test and power calculation
        ### for FDR levels f = 0.05 and f = 0.10.
        result = c(test.power(q.values,p.values,0.05),
            test.power(q.values,p.values,0.10))
        return(result)
    })
})      ### END ITERATIONS ###

sfStop()        ### Stop cluster.

### Record end time.
time2 = proc.time()

### Restructure output matrix.
output = matrix(output,nrow=1000,ncol=10,byrow=TRUE)
dimnames(output) = list(NULL,c("s.05","r.05","power.05",
        "estimated.fdr.05","true.fdr.05","s.10","r.10",
        "power.10","estimated.fdr.10","true.fdr.10"))

### Save output matrix in csv file.
filename = paste("pi1_",pi1,"_FC_",FC,"_ni_",ni,".csv",sep="")
write.csv(output, filename)

### Print out filename, time, means.
print(filename)
print(structure(time2 - time1, class = "proc_time"))
print(c("means",colMeans(output)))
```

8.4.2 One group, time trend

```
# One group, time trend
# FDR and permutation-based simulation study
# For sample size and power estimation

# Parameters used:
# sample size at each time point nt = 4 to 16, 20, 25, 30, 40, 50
# maximum number of time points t.max = 5
# total genes m = 5000
# proportion differentially expressed genes pi1 = 0.1
# fold change at t.max FC = 1.5, 2, 2.5, 3
# gene expression variance = 0.1
# pFDR level f = 0.05, 0.1
# tuning parameter for pFDR lambda = 0.5
# maximum number of permutations used B = t.max! = 5
# number of iterations for each set of parameters 1000

### Set working directory.
setwd("working directory")

### Load library snowfall (depends on snow) for parallel computing.
library(snowfall)

### Set values for global variables.
nt = 4
t.max = 5
m = 5000
pi1 = 0.1
FC = 1.5

### For t.max = 2 to 6: generate all possible permutations.
### For t.max > 6: generate 1000 unique permutations.
set.seed(42)
permutations = t(replicate(2000, sample(t.max, t.max)))
permutations = permutations[!duplicated(permutations),]
if(dim(permutations)[1] > 1000) permutations = permutations[1:1000,]
B = dim(permutations)[1]

### Create blocks for snowfall parallel execution (20 blocks of 50).
blocks = matrix(1:1000,nrow=20,byrow=TRUE)

### DEFINE FUNCTIONS ###
```

```
### Function to generate data.
generate.data = function(nt, t.max, pi1, FC, variance=0.1, m=5000){
    data = array(,dim=c(m,nt,t.max))
    for (t in 1:t.max){
        data[,,t] = matrix(c(rnorm(pi1*m*nt, mean=log2(FC)*((t-1)/
            (t.max-1)), sd=sqrt(variance)), rnorm((1-pi1)*m*nt, mean=0,
            sd=sqrt(variance))), nrow=m, ncol=nt, byrow=TRUE)
    }
    return(data)
}

### Function to fit OLS linear regression model to each gene.
### Input is one m*nt*t.max array.
### Output is vector of t-statistics.
lin.reg = function(data, m=5000){
    nt = dim(data)[2]
    t.max = dim(data)[3]
    t = matrix(rep(1:t.max, each=nt),
        nrow=m, ncol=(nt*t.max), byrow=TRUE)
    y.jt = t(apply(data,1,as.vector))
    sd.t = apply(t,1,sd)
    sd.y.jt = apply(y.jt,1,sd)
    r = sapply(seq.int(m), function(i) cor(t[i,], y.jt[i,]))
    beta.1.hat = r*(sd.y.jt/sd.t)
    beta.0.hat = sapply(seq.int(m), function(i){
        mean(y.jt[i,]) - beta.1.hat[i]*mean(t[i,])})
    fitted.values = beta.0.hat + t*beta.1.hat
    residuals.sqrd = (y.jt - fitted.values)^2
    s.beta.1.hat = sapply(seq.int(m), function(i){
        sqrt(sum(residuals.sqrd[i,])/
        ((nt*t.max-2)*(nt*t.max-1)*sd.t[i]^2))})
    Tj = beta.1.hat/s.beta.1.hat
    return(Tj)
}

### Function to apply positive FDR algorithm to obtain
### q-value for each gene. For 5000 genes.
qval = function(p.values, lambda=0.5) {
    ### Estimate number of null genes.
    m0hat = sum(p.values > lambda)/(lambda)
    pi0hat = m0hat/length(p.values)
    ### Sort p-values from largest to smallest.
```

```
    ordered.p.values =
        sort(p.values,decreasing=TRUE,index.return=TRUE)
    ### Value of largest p-value:
    p_m = ordered.p.values[[1]][1]
    ### Obtain q-value estimate for the largest p-value:
    q.p_m = ((m0hat*p_m)/sum(p.values <= p_m))
    q.values = matrix(NA,nrow=5000,ncol=1)
    q.values[1,] = q.p_m
    for (i in 2:m) {
        q.values[i,] = min(q.values[i-1,],
            ((m0hat*ordered.p.values[[1]][i])/
            sum(p.values <= ordered.p.values[[1]][i])))
    }
    return(q.values)
}

### Function to perform significance test and power calculation.
### Input q-values and desired FDR (f). Uses global variables pi1, m.
### Outputs s, r, power, estimated and true FDR.
test.power = function(q.values,p.values,f){
    q.val.index = which(q.values <= f)
    if (length(q.val.index)==0){
        s = 0
        r = 0
        power = 0
        estimated.FDR = 0
        true.FDR = 0
    } else {
        ordered.p.values =
            sort(p.values,decreasing=TRUE,index.return=TRUE)
        gene.index = ordered.p.values[[2]][q.val.index]
        s = sum(gene.index <= pi1*m)
        r = length(gene.index)
        power = s/(pi1*m)
        estimated.FDR = max(q.values[q.val.index])
        true.FDR = (r - s)/r
    }
    out = c(s,r,power,estimated.FDR,true.FDR)
    return(out)
}

### INITIALIZE SNOWFALL ###
```

```
### Enter number of cpus according to machine.
sfInit(parallel=TRUE,cpus=12,type="SOCK")
sfExportAll()   ### Export all objects in workspace to each node.

### Record start time.
time1 = proc.time()

### BEGIN ITERATIONS ###
output = sfApply(blocks,1,function(k){
    sapply(k,function(l){
        set.seed(l)
        ### Generate gene expression data.
        data = generate.data(nt, t.max, pi1, FC)
        ### Generate null distribution by permutation.
        perms.index = c(1:B)
        null.t.stats = sapply(perms.index,function(i){
            perm.data = data[,,permutations[i,]]
            return(lin.reg(perm.data))})
        ### Calculate the test statistics for the data.
        Tj = lin.reg(data)
        ### Calculate raw p-values using pooled null test statistics.
        p.values = sapply(Tj,function(x){(sum(abs(null.t.stats)
             >= abs(x)))/(B*m)})
        ### Positive FDR algorithm to obtain q-value for each gene.
        q.values = qval(p.values)
        ### Do significance test and power calculation
        ### for FDR levels f = 0.05 and f = 0.10.
        result = c(test.power(q.values,p.values,0.05),
            test.power(q.values,p.values,0.10))
        return(result)
    })
})      ### END ITERATIONS ###

sfStop()        ### Stop cluster.

### Record end time.
time2 = proc.time()

### Restructure output matrix.
output = matrix(output,nrow=1000,ncol=10,byrow=TRUE)
dimnames(output) = list(NULL,c("s.05","r.05","power.05",
        "estimated.fdr.05","true.fdr.05","s.10","r.10",
```

```
        "power.10","estimated.fdr.10","true.fdr.10"))

### Save output matrix in csv file.
filename = paste("t.max_",t.max,"_pi1_",pi1,"
         _FC_",FC,"_nt_",nt,".csv",sep="")
write.csv(output, filename)

### Print out filename, time, means.
print(filename)
print(structure(time2 - time1, class = "proc_time"))
print(c("means",colMeans(output)))
```

8.4.3 Two groups, time trend

```
# Two groups, time trend
# FDR and permutation-based simulation study
# For sample size and power estimation

# Parameters used:
# sample size per group at each time point n.it = 2 to 14
# maximum number of time points t.max = 5, 10
# total genes m = 5000
# proportion differentially expressed genes pi1 = 0.1
# fold change at end time FC = 1.5, 2, 2.5, 3
# gene expression variance = 0.1
# pFDR level f = 0.05, 0.1
# tuning parameter for pFDR lambda = 0.5
# maximum number of permutations used B = 100
# number of iterations for each set of parameters 1000

### Set working directory.
setwd("working directory")

### Load library snowfall (depends on snow) for parallel computing.
library(snowfall)

### Set values for global variables.
n.it = 4
t.max = 10
m = 5000
pi1 = 0.1
FC = 3
B = 100

### Generate group indicator variable for data.
I.group.data = rep(c(rep(0,n.it), rep(1,n.it)), t.max)
### Generate group indicator variable
### for B unique permutations of group assignment.
set.seed(42)
I.group.perms = matrix(replicate(1500,
    as.vector(replicate(t.max, sample(c(rep(0,n.it), rep(1,n.it)))))),
    nrow=1500, ncol=t.max*2*n.it, byrow=TRUE)
I.group.perms = I.group.perms[!duplicated(I.group.perms),]
I.group.perms = I.group.perms[1:B,]
```

```
### Create blocks for snowfall parallel execution (20 blocks of 50).
blocks = matrix(1:1000,nrow=20,byrow=TRUE)

### DEFINE FUNCTIONS ###
### Function to generate data.
generate.data = function(n.it, t.max, pi1, FC, variance=0.1, m=5000){
    data = array(,dim=c(m,2*n.it,t.max))
    for (t in 1:t.max){
        control.data = matrix(rnorm(m*n.it, mean=0,
                sd=sqrt(variance)), nrow=m, ncol=n.it)
        treatment.data = matrix(c(rnorm(pi1*m*n.it,
                mean=log2(FC)*((t-1)/(t.max-1)), sd=sqrt(variance)),
                rnorm((1-pi1)*m*n.it, mean=0, sd=sqrt(variance))),
                nrow=m, ncol=n.it, byrow=TRUE)
        data[,,t] = cbind(control.data, treatment.data)
    }
    return(data)
}

### Function to fit OLS linear regression model to each gene.
### Input is one m*2*n.it*t.max array
### and vector of indicator variables, length n.it*t.max.
### Output is vector of t-statistics for interaction term coefficient.
lin.reg = function(data, I.group){
    m = dim(data)[1]
    n = dim(data)[2]
    n.it = n/2
    t.max = dim(data)[3]
    ones = rep(1, t.max*n)
    t = rep(1:t.max, each=n)
    t.I.group = t*I.group
    y.jt = t(apply(data,1,as.vector))
    X = matrix(c(ones,t,I.group,t.I.group), nrow=t.max*n, ncol=4)
    beta.hat = t(apply(y.jt, 1, function(y){
        solve(t(X) %*% X) %*% t(X) %*% y}))
    y.jt.hat = t(X %*% t(beta.hat))
    sigma.j.hat.sqrd = rowSums((y.jt - y.jt.hat)^2)/(t.max*n - 4)
    ### Get variance of beta.3.j from the variance-covariance matrix
    ### of the least squares parameter estimates.
    var.beta.3.j = solve(t(X) %*% X)[4,4]*sigma.j.hat.sqrd
    Tj = beta.hat[,4]/sqrt(var.beta.3.j)
    return(Tj)
```

```
}

### Function to apply positive FDR algorithm to obtain
### q-value for each gene. For 5000 genes.
qval = function(p.values, lambda=0.5) {
    ### Estimate number of null genes.
    m0hat = sum(p.values > lambda)/(lambda)
    pi0hat = m0hat/length(p.values)
    ### Sort p-values from largest to smallest.
    ordered.p.values =
        sort(p.values,decreasing=TRUE,index.return=TRUE)
    ### Value of largest p-value:
    p_m = ordered.p.values[[1]][1]
    ### Obtain q-value estimate for the largest p-value:
    q.p_m = ((m0hat*p_m)/sum(p.values <= p_m))
    q.values = matrix(NA,nrow=5000,ncol=1)
    q.values[1,] = q.p_m
    for (i in 2:m) {
        q.values[i,] = min(q.values[i-1,],
            ((m0hat*ordered.p.values[[1]][i])/
            sum(p.values <= ordered.p.values[[1]][i])))
    }
    return(q.values)
}

### Function to perform significance test and power calculation.
### Input q-values and desired FDR (f). Uses global variables pi1, m.
### Outputs s, r, power, estimated and true FDR.
test.power = function(q.values,p.values,f){
    q.val.index = which(q.values <= f)
    if (length(q.val.index)==0){
        s = 0
        r = 0
        power = 0
        estimated.FDR = 0
        true.FDR = 0
    } else {
        ordered.p.values =
            sort(p.values,decreasing=TRUE,index.return=TRUE)
        gene.index = ordered.p.values[[2]][q.val.index]
        s = sum(gene.index <= pi1*m)
        r = length(gene.index)
```

```
        power = s/(pi1*m)
        estimated.FDR = max(q.values[q.val.index])
        true.FDR = (r - s)/r
    }
    out = c(s,r,power,estimated.FDR,true.FDR)
    return(out)
}

### INITIALIZE SNOWFALL ###
### Enter number of cpus according to machine.
sfInit(parallel=TRUE,cpus=12,type="SOCK")
sfExportAll()   ### Export all objects in workspace to each node.

### Record start time.
time1 = proc.time()

### BEGIN ITERATIONS ###
output = sfApply(blocks,1,function(k){
    sapply(k,function(l){
        set.seed(l)
        ### Generate gene expression data.
        data = generate.data(n.it, t.max, pi1, FC)
        ### Generate null distribution by permutation.
        perms.index = c(1:B)
        null.t.stats = sapply(perms.index,function(i){
            lin.reg(data, I.group.perms[i,])})
        ### Calculate the test statistics for the data.
        Tj = lin.reg(data, I.group.data)
        ### Calculate raw p-values using pooled null test statistics.
        p.values = sapply(Tj,function(x){(sum(abs(null.t.stats)
            >= abs(x)))/(B*m)})
        ### Positive FDR algorithm to obtain q-value for each gene.
        q.values = qval(p.values)
        ### Do significance test and power calculation
        ### for FDR levels f = 0.05 and f = 0.10.
        result = c(test.power(q.values,p.values,0.05),
            test.power(q.values,p.values,0.10))
        return(result)
    })
})      ### END ITERATIONS ###

sfStop()        ### Stop cluster.
```

```
### Record end time.
time2 = proc.time()

### Restructure output matrix.
output = matrix(output,nrow=1000,ncol=10,byrow=TRUE)
dimnames(output) = list(NULL,c("s.05","r.05","power.05",
            "estimated.fdr.05","true.fdr.05","s.10","r.10",
            "power.10","estimated.fdr.10","true.fdr.10"))

### Save output matrix in csv file.
filename = paste("t.max_",t.max,"_pi1_",pi1,"_FC_",FC,
            "_n.it_",n.it,".csv",sep="")
write.csv(output, filename)

### Print out filename, time, means.
print(filename)
print(structure(time2 - time1, class = "proc_time"))
print(c("means",colMeans(output)))
```