CORRELATION AND VARIANCE STABILIZATION IN THE TWO GROUP COMPARISON CASE IN HIGH DIMENSIONAL DATA UNDER DEPENDENCIES

by

DILAN C. PARANAGAMA

B.S., University of Colombo, 2005

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2011

Abstract

Multiple testing research has undergone renewed focus in recent years as advances in high throughput technologies have produced data on unprecedented scales. Much of the focus has been on false discovery rates (FDR) and related quantities that are estimated (or controlled for) in large scale multiple testing situations. Recent papers by Efron have directly addressed this issue and incorporated measures to account for high-dimensional correlation structure when estimating false discovery rates and when estimating a density. Other authors also have proposed methods to control or estimate FDR under dependencies with certain assumptions. However, not much focus is given to the stability of the results obtained under dependencies in the literature. This work begins by demonstrating the effect of dependence structure on the variance of the number of discoveries and the false discovery proportion (FDP). A variance of the number of discoveries is shown and the density of a test statistic, conditioned on the status (reject or failure to reject) of a different correlated test, is derived. A closed form solution to the correlation between test statistics is also derived. This correlation is a combination of correlations and variances of the data within groups being compared. It is shown that these correlations among the test statistics affect the conditional density and alters the threshold for significance of a correlated test, causing instability in the results. The concept of performing tests within networks, Conditional Network Testing (CNT) is introduced. This method is based on the conditional density mentioned above and uses the correlation between test statistics to construct networks. A method to simulate realistic data with preserved dependence structures is also presented. CNT is evaluated using simple simulations and the proposed simulation method. In addition, existing methods that controls false discovery rates are used on t-tests and CNT for comparing performance. It was shown that the false discovery proportion and type I error proportions are smaller when using CNT versus using t-tests and, in general, results are more stable when applied to CNT. Finally, applications and steps to further improve CNT are discussed.

CORRELATION AND VARIANCE STABILIZATION IN THE TWO GROUP COMPARISON CASE IN HIGH DIMENSIONAL DATA UNDER DEPENDENCIES

by

DILAN C. PARANAGAMA

B.S., University of Colombo, 2005

A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2011

Approved by:

Major Professor
Prof. Gary L. Gadbury

Copyright

Abstract

Multiple testing research has undergone renewed focus in recent years as advances in high throughput technologies have produced data on unprecedented scales. Much of the focus has been on false discovery rates (FDR) and related quantities that are estimated (or controlled for) in large scale multiple testing situations. Recent papers by Efron have directly addressed this issue and incorporated measures to account for high-dimensional correlation structure when estimating false discovery rates and when estimating a density. Other authors also have proposed methods to control or estimate FDR under dependencies with certain assumptions. However, not much focus is given to the stability of the results obtained under dependencies in the literature. This work begins by demonstrating the effect of dependence structure on the variance of the number of discoveries and the false discovery proportion (FDP). A variance of the number of discoveries is shown and the density of a test statistic, conditioned on the status (reject or failure to reject) of a different correlated test, is derived. A closed form solution to the correlation between test statistics is also derived. This correlation is a combination of correlations and variances of the data within groups being compared. It is shown that these correlations among the test statistics affect the conditional density and alters the threshold for significance of a correlated test, causing instability in the results. The concept of performing tests within networks, Conditional Network Testing (CNT) is introduced. This method is based on the conditional density mentioned above and uses the correlation between test statistics to construct networks. A method to simulate realistic data with preserved dependence structures is also presented. CNT is evaluated using simple simulations and the proposed simulation method. In addition, existing methods that controls false discovery rates are used on t-tests and CNT for comparing performance. It was shown that the false discovery proportion and type I error proportions are smaller when using CNT versus using t-tests and, in general, results are more stable when applied to CNT. Finally, applications and steps to further improve CNT are discussed.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank the faculty in the Department of Statistics at Kansas State University for the opportunity given to me to pursue a degree in Statistics and their guidance and supervision throughout the years. I would like to thank Dr. John Boyer and Dr. Jim Neill for their leadership and guidance in the department during my stay.

I like to extend my gratitude to my committee members, Dr. Gary Gadbury, Dr. Paul Nelson, Dr. Suzanne Dubnicka, Dr. Karen Garrett and Dr. Donald Saucier for the support given to not only for this dissertation work but for guidance given to me outside of academia and classrooms.

I would like to thank the faculty in the Department of Statistics at University of Colombo, Sri Lanka, for their support and special thank you goes to Dr. Nimal Wickremasinghe for guiding me to pursue further education.

A hearty appreciation must be given to my advisor, Dr. Gary Gadbury for his countless efforts in helping me during my research work. This work would not be a possibility without his guidance and support.

I would also like to thank the staff of the Department of Statistics at Kansas State University, especially Pam, for the support given to me and making my life much easier on daily basis.

I would like to thank my parents for the support given to me and being there for me whenever I needed assistance. Finally I would like to thank my wife Thilanka, whose support and encouragement made this work possible and enjoyable, and my son Nevin, for simply being wonderful.

Chapter 1 - Introduction

High-dimensional (HD) data refers to sets of data in which the number of features or variables exceed the number of samples in the dataset. Microarray data and lipidomics data are two examples of HD data. Microarray experiments typically seek to identify genes (later more generally referred to as features) that are under or over expressed in a test tissue sample relative to a control sample. This is determined by the detection of messenger RNA (mRNA) that is present in the tissue samples and the data quantify an expression level for each of possibly thousands of genes for each sample. Lipidomics experiments seek to identify the composition of lipid species in tissue (plant or animal) samples and how the composition varies across treatment conditions. The lipid compounds are detected with mass spectrometry instruments. The majority of this report will focus on the context of microarray experiments, but the analysis techniques discussed apply to any situation where a large number of hypotheses are tested simultaneously. Technologies for "omics" platforms (genomics, lipidomics, metabolomics, etc.,) have undergone new and rapid developments in recent years. The advancements in both technologies used to collect the data and availability of powerful computers to analyze large datasets have contributed to this evolution. Since the technologies have recently become more affordable, vast amounts of HD data are being produced and analyzed. In addition, there have been major developments in the field of HD data analysis parallel to the advancements in technologies. Most of the common multivariate data analysis methods are not applicable to HD data since they suffer from the "curse of dimensionality" (Donoho 2000, Johnstones & Titterington 2009). Therefore, techniques for analyzing HD data have developed in novel ways and represent a combining of classical and Bayesian statistics, bioinformatics, computer science, and new paradigms for thinking about multiple testing. The number of features (variables) in a lipidomics dataset may range in the hundreds, and the number of features in a microarray dataset ranges in the many thousands. Two microarray examples are used in this report. The lung cancer data set consists of 12687 genes, and the multiple myeloma dataset consists of 3970 genes. Both experiments evaluate the expression of genes in tissue samples from diseased and healthy subjects.

Two treatment comparison studies have been common in microarray experiments (e.g., Thimmulappa et al, 2002 and Tanaka et al, 2000), but more complex experiments have been

done as well (e.g., Garrett et al, 2006). One of the common interests in HD data is to identify the features that are significantly different between two groups of individuals (e.g., the expression of genes in diseased vs. healthy, wild type vs. mutated etc.). When two groups are being compared, a t-test is commonly used to test for differential expression of genes between two groups. A challenge with high-dimensional data is that, at conventional α levels, these tests produce a large number of type I errors. For example, at a 0.05 level, the analysis of lung cancer data would be expected to make 634 type I errors even if there were no genes (out of 12,687) that were differentially expressed. At 0.01 level this number is 126. These numbers provide a quick insight to the extent of the problem of large scale testing. The two group comparison usually acts as an initial analysis to identify significant features. Further research then follows to study the factors affecting these features and having a large number of false positive results lead to waste of resources. The problem of a large number of false discoveries does not only pertain to the t-tests but is a problem in all situations where a large number of hypotheses are tested simultaneously.

Methods for the analysis of 'omics' data tend to develop down separate but overlapping paths. One path for methods development involves background correction to remove random noise present in the raw signal generated by the high-throughput technology (e.g., Cope et al, 2004). Another path involves normalization techniques to remove biases and noise that are interjected into the experiment because of characteristics in the technologies (e.g., Irizarry 2008). Yet another path involved the development of meaningful metrics to quantify the differential expression of a gene across two or more groups (cf., Pepe et al., 2003). Another research path, and the one considered herein, are methods to accommodate the multiple testing issues present in HD experiments.

A large body of work exists in the literature addressing multiple comparisons and entire texts have been written on the topic (e.g., Hochberg & Tamhane 1987, Westfall & Young 1993). One of the widely known and commonly used methods is the Bonferroni control method for multiple testing. Holm (1979) also introduced a sequential testing method for controlled multiple testing. There are variations of these methods introduced by different authors with different improvements. These methods attempt to control the family wise error rate (FWER) which controls the probability of making at least one type I error in simultaneous testing of multiple

hypotheses. These methods are quite conservative and, at small α levels these methods may fail to make any significant discoveries, resulting in a large number of type II errors.

Benjamini & Hochberg (BH, 1995) presented their landmark work in simultaneous testing. BH presented their paper in the contexts of large experimental designs where multiple treatment comparisons may be made subsequent to an omnibus test of treatment differences. They might not have imagined at the time how their method would be built upon and, combined with needs presented by emerging high-throughput technologies, revolutionize the area of multiple testing research. More details of the literature review for multiple testing techniques will be given in Chapter 2, but some overview is given here to set the stage for introduction to the research presented in this dissertation. BH introduced the concept of false discovery rate (FDR) in terms of an expected value of a ratio. The false discovery rate is defined as the expected proportion of false findings among the total number of discoveries (i.e., statistically significant results) made by simultaneous testing. It can be thought of as the expected proportion of false findings when the experiment is repeated a large number of times. In this work they introduced a method to control the false discovery rate and also a sequential testing method similar to what Holm (1979) used to control the FWER. Extensions to BH were given in Benjamini and Yekutieli (2001) who introduced a method similar to BH (1995) to control FDR under a dependence structure. The problem of false discoveries was also addressed by Storey (2002), who introduced an estimate of the false discovery rate (and called it the positive false discovery rate, pFDR) and the q-value which is the Bayesian counterpart to the p-value. The latter, loosely stated, is the probability of the data given the null hypothesis and the former, also loosely stated, is the probability of the null given the data. This statement about the q-value is sometimes how a p-value is "mis"-interpreted by subject matter scientists (cf. Berger and Selke, 1997). FDR control generally works analogously to FWER control in that the significance level is adjusted. Controlling the FDR at a level of 0.05 means that among a set of discoveries from significance testing, it is expected that 5 percent will be false leads. The recognition of the value of this approach in HD settings was that a certain proportion of false leads could be tolerated in favor of not missing any important findings with overly conservative methods of adjustment.

The work by Benjamini and colleagues, as well as others, focused on control of the FDR where control was defined in an "expectation sense," that is, the control of false discoveries is

controlled at a certain level (say, 0.05), over repeated realizations from the high-dimensional experiment. Or perhaps more interest is the proportion of false discoveries among a list of "findings" from a single experiment. This has sometimes been called a false discovery proportion (FDP) to distinguish it from FDR as a means of error rate control as discussed above. Thus, many methods have been proposed to estimate the FDP using data from a single experiment. These methods generally require estimation of the proportion of "true null hypotheses" which can be thought of as a prior probability on the proportion of nulls. It is interesting to note that estimation of the proportion of true nulls was considered much earlier in the context of a large ANOVA type of experiment by Schweder and Spjøtvoll (1982) using their p-value plot. Some methods for estimating FDP will be reviewed in the next section but many involve a mixture of cumulative distribution functions (CDFs), one for a null distribution and the other for the distribution of a test statistic under the alternative hypothesis. Efron (2001) introduced the local false discovery rate which is estimated by a mixture of densities and interpreted as a posterior probability that a particular test is a false discovery, given that it was declared significant, rather than an estimated proportion of false discoveries among significant results. Many techniques for estimating FDP and related quantities have modeled the distribution of p-values from multiple tests (cf. Allison et al., 2002).

Two issues emerged concerning the validity of estimates of FDP (or its local version) and their usefulness for multiple testing adjustments. These were discussed by Hu et al., (2010), and dealt with by others in various ways. One issue is the choice of the null distribution of the test statistic. Many of the methods for FDR control and estimation of FDP operate on the distribution of p-values from multiple tests, and these methods generally assume that the distribution of a p-value under the null hypothesis is uniform on the interval 0 to 1. This assumes that the correct reference distribution was used in computing the p-value. Efron (2004) presented a method to empirically estimate the reference distribution. Others have considered permutation tests for computing p-values; however, such tests produce p-values with a discrete distribution, making their use problematic in HD settings where often very small p-values are needed to declare significance (cf., Gadbury et al., 2003). A second issue concerns the correlation structure of test statistics from multiple tests.

Benjamini & Yekutielli (2001) considered the issue of dependence in FDR control where, again, FDR control is considered in an expectation sense. Early papers discussing the importance of correlation structure in actual estimates of quantities of interest from high dimensional experiments (HDE) were Klevanov and Yakovlev (2007) who stressed that the variance in estimators can be unpredictably high. Owen (2005) introduced a method to estimate the variance of the number of false discoveries where the correlation was due to the expression of multiple genes being tested against a common phenotype.

Efron, in his 2007 and 2010 publications, introduced a method to estimate the false discovery rate and adjustments to the variance of the density estimates taking the correlation structures into consideration. These works suggest that the results under independence are not suitable and often yield high variability when the independence assumption is violated, which is likely in gene expression since many genes are co-regulated.

Efron's work demonstrates that not only FDR but also density estimates of the test statistics become highly variable under dependence. This work showed that the correlations can be accounted for in these estimates by introducing a penalty for the dependence structure. Efron's work is one of the small amount of work that has been done on the problem of dependence structure in HDE. The handful of work that addresses this issue estimates the correlation structure by calculating the correlations of the data between the features (i.e., genes) that are being tested. Efron (2010) obtained a measure for the dependence structure by computing the variance of the correlations for the two groups being tested separately. A measure is obtained by calculating a weighted average of the two variances. Other authors have assumed that both groups being tested have the same dependence structure (Kim and Wiel, 2008).

Southworth et al (2009) demonstrated that the correlations among genes in mice change as they age. This introduces the idea that the correlation structures in the two groups being tested may be different. It can also be the case that a treatment applied to one of the groups not only changes the mean structure but also alters dependence structure as well. This is not an issue that is limited to microarray data. The assumption that the two groups being compared have the same covariance structure may be a strong assumption for any type of data. This indicates that the assumption of the same correlation structure in both groups being tested may often be violated.

One of the challenges faced by researchers in the field of high dimensional data analysis is being able to simulate realistic data. Realistic data are needed to evaluate the performance of statistical methods that analyze high-dimensional data sets. The difficulty in simulating realistic data is mainly due to the lack of ability to simulate data with arbitrary covariance matrices. The covariance matrices that are estimated by observed data often do not meet the requirement of covariance matrices for simulations. While pairwise covariances can be estimated from the data, a covariance matrix with these pairwise covariance estimates as elements could not be used for simulating data. Many data simulation algorithms require positive definite covariance matrices and these estimates often do not meet that requirement. The other issue is that the dimensions of these matrices are too large that regular personal computers are unable to handle them. There are numerous publications that present methods for high-dimensional data and that address different issues but, of those that even consider correlated genes, most only use basic block diagonal covariance matrices for simulations (Hu et al 2010). Simple comparisons between correlation distributions show that these block diagonal matrices do not correctly represent dependence structure in real data. Gadbury et al (2008) suggested a method to simulate data that are closer to real world data. However this method does not allow for a different dependence structure in different groups being compared.

The research described herein addresses the issue of correlations in large scale two group comparisons while attempting to relax some of the assumptions used in earlier work. The work previously done in this area does not directly address the possibility that the two groups may have different dependence structures or that the treatment applied to the subjects alters the dependence structure. The main goal of this work is to develop a two group comparison method that adjusts for the changes in the dependence structure while stabilizing the variance of the number of discoveries. In particular, the method adjusts p-values for dependence structure so that the many methods that have been developed in recent years that operate on the distribution of p-values can use the adjusted distribution as an alternative.

Chapter 2 presents a literature review that provides more detail on multiple testing research related to the topic of this dissertation. Then, the initial work presented in Chapter 3 illustrates the effect of the dependence structure on the variance of the number of discoveries. Specifically, the increment of variance under dependence structures of the data is shown. A

closed form expression for the variance of the number of discoveries is derived. These initial results show that the variability of the number of discoveries increases as the dependence structure increases. A closed form for the correlation between the test statistics is derived in Chapter 4 and its sampling distribution is derived in Chapter 5. Chapter 4 investigates the behavior of conditional densities of test statistics under different conditions. Chapter 6 proposes a method to build networks of features that are defined by the correlations between test statistics and test them within the networks using the conditional density derived in Chapter 4. The method is referred to as *Conditional Network Testing*. Chapter 7 puts forward a new method to simulate data with the preserved dependence structure of the observed data. Chapter 8 illustrates the effects of using conditional network testing and the reduction in variance of the number of discoveries. Initial results show that the variance of the number of discoveries is controlled by Conditional Network Testing while providing more stability to false discovery proportions and type I error proportions. Chapter 9 investigates the behavior of false discovery rate control methods applied to the p-values obtained by conditional testing and compares them with the t-test results. Chapter 10 discusses possible applications of conditional network testing and suggests further improvements.

The initial results show that conditional network testing is capable of reducing the variance of the number of discoveries which was the main intention. In addition, it is shown that this method produces smaller proportion of false discoveries and achieves a lower rate of type I errors. However, the downside is that this results in an increment in type II error rate. The variance of the false discovery proportion and the type I errorproportion are also reduced by the suggested method but results an increased variance for the type II error rate. The proposed method has the potentioal to be a strong competitor among the methods used for testing when the emphasis is on the features that are declared as statistically significant.

Chapter 2 - Literature Review

A considerable amount of work has been done in the area of HD data analysis addressing different issues. This section briefly introduces the sources of HD data, specifically microarray data without going deep into the biological aspects. Some of the common data analyses used in this field are briefly introduced and followed by a more detailed description of the two-group comparison problem and false discovery rate procedures. Additional material related to the topic addressed in this dissertation is also discussed.

2.1 Sources of HD Data

While commonly produced in biological applications, HD data can be produced in any field. Advancements in biology have allowed mass production of high quality HD data that is rapidly becoming more affordable with advancing technologies. Microarray data, one of the common types of HD data, are obtained by the process of DNA-RNA hybridization. The biology behind the microarrays is introduced in Allison et al (2006) and additional background information is given in McLachlan et al (2004). Analysis of compounds such as lipids or metabolytes use a different technology but also produce similar HD data. These latter data are produced by analyzing samples in a mass spectrometer. An introduction to mass spectrometer technologies is given in Silverstein (2002). Statistical analysis for these HD data usually are similar but the dimensions in microarray data are much higher than the data produced by some other technologies.

There are numerous organizations that collect and manage these data. Gene Expression Omnibus (GEO, *www.ncbi.nlm.nih.gov/geo/*) is an online repository of microarray data. A large number of microarray data sets are available for public use in this website along with useful information and tools to manage these data. Lipid Maps (*www.lipidmaps.org*) is a similar online resource that manages lipidomics data. Two real datasets will be used for illustrations in this dissertation. They are introduced in section 2.6

2.2 Preprocessing Methods

Microarray data are obtained by scanning gene chips for probe intensities. The mass spectrometer similarly produces compound abundance intensities. These machines introduce

both variance and bias into data which must be dealt with before the data are used for any analysis. Techniques for correcting for bias and sources of variance introduced by the high-throughput technologies are referred to as background adjustment and normalization. Background adjustment is known as baseline correction in mass spectrometer data. SAM (Tusher et al 2001) and limma eBayes (Smyth 2004) are two commonly used methods that include techniques for background adjustment. Ritchie et al (2007) presented a comparison of these methods.

Normalization methods focus on reducing systematic variance in HD data. MAS-5, RMA and MBEI are commonly used methods for normalizing microarray data. These methods are discussed in detail in McLachlan (2004) and Allison et al (2006). Bolstad et al (2003) and Irizarray et al (2003) presented comparison of these methods. Common normalization methods of mass spectrometer data are built into the software used in the machine, such as MZMine (Katajamaa et al, 2006).

In addition to these methods, there are other measures for controlling the quality of the data. Some of these methods include the introduction of quality control probes in microarrays and inserting known compounds into mass spectrometer runs. The information from these probes and compounds are used in background correction and normalizing of data, and then later removed in the data cleaning process.

2.3 Common Analyses for HD Data

Once a HD data set is background corrected and normalized it can be used for planned analyses. A regular t-test for all features across two treatment groups (or two at a time for multiple treatment groups) is one of the most basic analyses performed. Another approach is to start with a principal components analysis as a measure of reducing the dimensionality. Typical multivariate analyses can then be performed on these dimensionality-reduced data. Analyses such as clustering or multidimensional scaling are also used for analysis of these datasets. However, one of the drawbacks of these methods is that it is not always possible to obtain meaningful principal components and, when this is the case, all information about features is lost in principal components. Classification and Regression Trees (CART) is a common method used for classification. Introduced in Brieman et al (1984), this method has been shown to work well

in regular multivariate environments. CART provides a list of important variables that lead to correct classifications. However, this method may not be directly applicable for HD data. Random forests, which is an extension of the CART tree concept, is used as a common classification method for HD data. Random forests get around the problem of high dimensionality by using only subsets of features at a time (Breiman, 2001). Classification methods including random forests, k-means method etc., are targeted to classify samples in the study to some phenotypic group, based on HD data

It is sometimes of interest to investigate the groups of features (genes, lipids, etc.) that function together. There are known genes, lipids or other compounds in biological systems that are linked together by their functions, physical positions (in genes) or the way they react to a treatment or a stress condition. Identified classes of genes are defined in gene ontology (GO) (Ashburner et al, 2000) and KEGG databases (Kanehisa and Susumu, 2000). Lipids, metabolites, proteomics etc. are also classified and studied in terms of groups of compounds that function together. In microarray data, gene sets defined by GO classification are often tested to see if they are affected by a treatment or other phenotypic difference as a group, among all of the genes. Gene set enrichment analysis is one of these analyses where a set of genes belonging to the same class are tested against the rest of the genes on the chip for differential expression (Subramanian et al, 2005). In lipidomics (and metabolimics, proteomics), this translates to presence or absence of a set of compounds that are known to associate with each other.

Pathway analysis is another type of analysis performed on HD data that concerns multiple features at the same time. In genetics, pathway analysis investigates whether all or several genes in a known genetic pathway are affected by a treatment (or other phenotypic condition) (Schilling et al, 1999). In lipidomics, a pathway is usually translated into a series of reactions occurring in a biological system. One can identify if a pathway is occurring by following the reactant to the end product on a known pathway or by investigating leftover compounds in intermediate reactions. These known pathways are documented and available for public use via the internet. Lipid maps (*www.lipidmaps.org*) is an interactive online website that is widely used for analysis of lipids. KEGG (*genome.jp/KEGG*) is an online repository for genetic information and KEGG Pathway is an online tool to map and investigate genetic pathways. These tools are widely used for scientific research and are being regularly updated.

2.4 Two Group Comparison Tests and the False Discovery Rate

The two group comparison is a commonly used method in HD data analysis (e.g. between treatment vs. control / wild type vs. mutant type). Often the interest is to investigate which features are significantly different between the two groups being compared. A t-test is commonly used for these comparisons, and an individual test is performed for each feature. As the number of comparisons increase, so does the family-wise error rate, FWER, which is the probability of making a type I error. This is an issue addressed by many authors, and there are a number of methods available to handle the increment in the type I error rate, such as Bonferroni, Dunett (Westfall & Young 1993), etc. These methods work reasonably well when the number of comparisons is relatively small.

In HD data analysis, a very large number of comparisons is done simultaneously, and this presents a different perspective on the traditional multiple testing problem. Since the number of hypotheses being tested is large, traditional α levels result in a large number of falsely declared significant results. These lead to a waste of resources in follow-up investigations, and attempts to employ conventional methods to control FWER results in overly conservative tests. For example, assuming that the interest is in comparing K features between two groups, the Bonferroni method suggests performing each comparison at an α/K level. When the number of comparisons K is large, these tests become too conservative and may fail to identify important differences between the two groups.

An early work addressing this issue was given in Holm (1979) who introduced a sequential procedure to control the FWER at a desired level. This method starts by ordering the hypotheses by their p values. Let $p_{(1)}, p_{(2)}, \ldots, p_{(K)}$ be the ordered list of p-values obtained from a series of t-tests for the K features and $H_{(1)}, H_{(2)}, \ldots, H_{(K)}$ be the corresponding hypotheses. The criteria is to reject $H_{(1)}, \ldots, H_{(t)}$ where for all $i = 1, \ldots, t$

$$p_{(i)} \leq \frac{\alpha}{K - i + 1} \tag{2.1}$$

Benjamini and Hochberg (1995) introduced a new way of thinking about the multiple testing problem. Their proposed method sought to control an expected false discovery rate. Their definition of FDR is as follows.

Assume that K hypothesis tests were conducted and R were declared significant at some α level. In addition, assume the null hypotheses for V tests are true among the R that were declared statistically significant. Then the FDR is given by

$$FDR = \begin{cases} E(V/R) & R > 0 \\ 0 & R = 0 \end{cases} \tag{2.2}$$

Although R would be known, the number V or the indices of true null hypothesis would not be known. They introduced a sequential testing method, similar to Holms, but to control the false discovery rate in multiple testing. BH method also begins by ordering the p-values obtained from multiple tests. Assume that these ordered p-values are denoted by $p_{(1)}, p_{(2)}, \ldots, p_{(K)}$ and their corresponding hypotheses are $H_{(1)}, H_{(2)}, \ldots, H_{(K)}$. The criteria is to reject the hypotheses $H_{(1)},\ldots,H_{(t)}$ where

$$t \text{ is the largest } i \text{ such that } p_{(i)} < \frac{\alpha}{K - i + 1} \tag{2.3}$$

This work also showed that use of this sequential method has a gain in power compared to FWER control methods such as Holm and Bonferroni methods. This makes the methods controlling the FDR preferable over methods that attempt to control FWER in HD data, when one is willing to tolerate a small proportion of Type I errors in favor of not missing important results.

Storey (2002) also contributed to estimating and controlling the false discovery rate. In this publication he defines the false discovery rate as

$$FDR = E\left(\frac{V}{R} \mid R > 0\right) P(R > 0) \tag{2.4}$$

and denotes the "positive false discovery rate" as

$$pFDR = E\left(\frac{V}{R} \mid R > 0\right) \tag{2.5}$$

where, again, K hypothesis tests were performed, R were declared statistically significant, and V is the number of true null hypothesis that were among the R rejected tests. Let $H_i = 0$ if the ith

null hypothesis is true and $H_i = 1$ if the ith null hypothesis is not true, and assume that these events on hypotheses are *a priori* identically distributed. Define the probabilities $P(H_i = 0) = \pi_0$ and $P(H_i = 1) = \pi_1$. If P is a random p-value from a test, the pFDR at a threshold of significance α is defined as

$$pFDR(\alpha) = \frac{\pi_0 \Pr(P \le \alpha \mid H = 0)}{\Pr(P \le \alpha)} = \frac{\pi_0 \alpha}{\Pr(P \le \alpha)} \tag{2.6}$$

The issue with definitions (2.4) and (2.5) is that neither the number of true null hypotheses that were rejected (V) nor the proportion of true null hypotheses (π_0) is known in real applications. Storey also presented a method to estimate the proportion of true null hypotheses subject to a tuning parameter λ, given by

$$\hat{\pi}_0(\lambda) = \frac{\#\{p_i > \lambda\}}{(1-\lambda)K} \tag{2.7}$$

Algorithms to calculate FDR and pFDR and evaluation of the performance of these estimates are also given in the same publication.

The works presented above rely on the assumption that individual tests are independent of each other. Hu et al., (2010) discussed the issues that arise when p values are correlated. They adopted the idea presented by Schweder and Spjøtvoll (1982) on modeling the p-values from large number of comparisons. In particular, they showed the expected number of discoveries denoted by N_0 when all the null hypotheses are true as,

$$E(N_0) = K(1 - \alpha)$$

and, assuming all pairwise correlations are equal, the variance of the number of discoveries

$$Var(N_0) = K\alpha(1-\alpha) + K(1-K)Cov(D_i, D_j) \tag{2.8}$$

where D_i is defined as

$$D_i = \begin{cases} 0 & p_i \le \alpha \\ 1 & p_i > \alpha \end{cases}$$

This accommodates dependence among tests but assumes that the dependence structure is the same across the two groups and among all pairwise tests. The effect of dependence on the

variance of the number of discoveries was also illustrated for different dependence structures in this work.

A series of publications by Efron (2004, 2007, and 2010) addressed some of the issues that arise in analyzing HD data. Efron (2004) introduced the idea of local false discovery rate. lFDR at particular value of a test statistic, z, as

$$lFDR(z) = \frac{\pi_0}{\pi_0 + (1-\pi_0)\,{}^{f_1(z)}\!/_{f_0(z)}} \tag{2.9}$$

where $f_1(z)$is the density of the non-null test statistics, $f_0(z)$is the density of the null test statistics computed at z and π_0 is the proportion of null hypotheses. This publication also proposed methods to estimate or adjust the null reference distribution of a test statistic rather than assuming a commonly used one (i.e., such as the normal or the t-distribution). It also provided methods to estimate each term in (2.9) so *lFDR* is a quantity that can be computed for an observed set of data, and is interpreted as a posterior probability that a feature with a particular test statistic value, z, is a false discovery.

Efron (2007) addressed the problem of correlated features. This work starts by transforming the test statistics to ensure the normality assumption under the null hypothesis.

$$z_i = \Phi^{-1}(p_i) \tag{2.10}$$

where z_i is the transformed test statistic for the ith feature tested, p_i is the p value obtained for the ith feature, and $\Phi^{-1}(.)$ is the inverse normal CDF. The theory is developed considering the bin counts of a histogram drawn to these transformed test statistics. Under the null hypothesis and independence, these counts should match the standard normal distribution. The covariance of the bin counts, denoted by y, can be written in the form

$$\mathrm{cov}(y) = C_0 + C_1$$

where C_0 is the covariance matrix under independence and C_1 accounts for the covariance structure, later referred to as the penalty for the correlations. The forms of C_0 and C_1 are given in his publication. This work also showed that the covariance of y (density estimate of the test statistics) is dominated by the correlations among the features, by comparing the first eigen

vector of the covariance matrix with the suggested form of the variance which accounts for the dependence structure. This verifies that the density of the test statistics is affected by the correlations in the data and the effect can be quantified by the root mean square correlation, given by

$$\alpha = \alpha_2^{1/2} = \left[\int_{-1}^{1} \rho^2 g(\rho) d\rho \right]^{1/2}$$

where $g(\rho)$ is the density of the correlations. Efron suggested an alternative estimate of the false discovery rate, adjusted for the correlation structure, given by

$$FDR(x \mid \hat{A}) = FDR_0 (1 + \hat{A} \frac{x\varphi(x)}{\sqrt{2}[1 - \Phi(x)]}) \tag{2.11}$$

where each hypothesis is rejected if the test statistic value is greater than x. FDR_0 is an initial estimate of the false discovery rate, and this work uses the estimate given in (2.6) under the assumption that π_0 is equal to 1. $\hat{A}$ is an estimate of the dispersion variate. This is given by

$$\hat{A} = \frac{2\Phi(x_0) - 1 - Y_0 / K}{\sqrt{2} x_0 \varphi(x_0)} \tag{2.12}$$

where Y_0 out of a total of K test statistics fall within the central limits of ($-x_0$, x_0). Φ and φ are the cumulative probability and density function of a standard normal distribution. The choice of x_0 is explained in this publication, but it was not clear how one would choose this value in general cases. The positive values of A indicate that the distribution of the test statistics is wider than the standard normal density (heavier tails) and negative values of A indicate the distribution of the test statistics is narrower than the standard normal density (lighter tails). This work also demonstrates that the false discovery proportion increases with increasing A. And also the variance of the density of test statistics has a direct relationship to the second moment of the density of the correlations, which is estimated empirically by computing all possible pairwise correlations among genes, after removing any mean differences due to treatment. The publication further explains that the null cases individually follow the standard normal density, but the variance of the test statistics for the alternative hypotheses may be larger than 1. The increment in the variance may be due to the effect of the correlations among the features.

Efron (2010) further investigates the effect of the correlations on the estimates of the distribution of the test statistics. The techniques start by performing the transformation in (2.10) on the p-values obtained from a series of independent t-tests. The "survival curve" of z_is, i.e., the test statistic transformed from t to standard normal, is defined as

$$\hat{F}(x) = \#\{z_i > x\} / N \tag{2.13}$$

where N is the total number of tests conducted, and x is as defined above. Similar to the work in Efron (2007), a discrete version of this density is obtained by considering the bin counts in the histogram drawn for z_is. The bin counts are denoted by y, and it is shown that the variance of the distribution of y can be decomposed into two parts. Similar to Efron (2007), the first part explains the variance under independence and the second part explains the variance due to the correlations. Assuming that C bins were used in the histogram of the test statistics, denote p_c as the proportion of z_is in the c^{th} bin. Let π_c be the vector of π_{kc} defined as

$$\pi_{kc} = prob_c(z_i \in Z_k)$$

where Z_k denotes the k^{th} bin in the histogram. The first part of the decomposition of the covariance matrix can be written as a sum over bins given by

$$\mathrm{cov}_0 = N \sum_c p_c \{diag(\pi_c) - \pi_c \pi_c'\}$$

And the form of the second part is given by

$$\mathrm{cov}_1 = (N\Delta\alpha)^2 \bar{\phi}^{(2)} \bar{\phi}^{(2)'} / 2$$

which is also known as the *rms* (root mean square) approximation. Δ is the common bin width and the function $\bar{\phi}^{(j)}$ (j^{th} derivative of the function $\bar{\phi}$) is given by

$$\bar{\phi}^{(j)} = \sum_c p_c \varphi_c^{(j)} / \sigma_c$$

where $\bar{\phi}^{(2)}$ can be obtained by

$$\varphi^{(2)}(u) = \varphi(u)(u^2 - 1)$$

where $\varphi(\cdot)$ is the standard normal density function and α is the room mean square correlation defined above. Covariance terms above are evaluated at each bin midpoint x_c. This work suggested that an estimate of α can be obtained through the variance of the all $N(N\text{-}1)/2$ pairwise correlations of the data. Several alternatives to estimating α are also presented. In addition, this work also describes that the variance of the *lFDR* estimates can also be decomposed into two parts, variance under independence and the penalty accounting for the covariance.

All the work done by Efron in HD data analysis is based on the histogram of the test statistics. If the histogram has K divisions, the space of test statistics can be represented by

$$Z = \bigcup_{k=1}^{K} Z_k$$

(Z_k is defined above) The y vector described earlier denotes the bin count of the histogram which acts as a discrete alternative to the density of the test statistics. Often a smooth curve fitted to the bin counts is used as the empirical density. Efron suggested that the new theories can be formulated based on the bin counts instead of the actual density of the test statistics.

An interesting point was made by Westfall in the discussions of Efron 2010. Westfall suggested that any correlation penalty introduced in terms of a multiple comparison procedure to account for the correlation structure makes tests more conservative. Therefore, some loss in power of the tests is to be expected under adjustments made to the tests.

Owen (2005) derived an approximation of the variance of the number of discoveries in a HD data analysis. The distribution of the square of the correlation coefficient of the data is given by

$$\hat{\rho}_j^2 \sim Beta\left(\frac{1}{2}, \frac{n-2}{2}\right) \tag{2.14}$$

The square of the correlations limits the density to its positive side. The approximation of the variance of the number of discoveries is derived to be

$$\text{var}(N^A) = d\alpha(1-\alpha) + d(d-1)\left(C(0) - \alpha^2 + \frac{C''(0)}{2(n-1)}\right) \tag{2.15}$$

for d features tested at a level α. $C(.)$ is the probability of rejecting two correlated test statistics as a function of the correlations between their features defined by

$$C(\rho_{jj'}) = P(Z_j = 1, Z_{j'} = 1 \mid \mathrm{Y}) \tag{2.16}$$

and *C''* denote the second derivative of *C*(.). $\rho_{jj'}$ denotes the correlation between jth and j'th feature. Z_j = 1 denotes the event that the test for the jth feature is rejected (Null hypotheses for both features are true). Y denote the data. C(0) is the above probability when the features are not correlated.

2.5 Other Related Topics

This section reviews the literature for a collection of topics that appear later in this report, and that are relevant to the results presented herein. Topics include what has been done on simulating HD data, different dependence structures in two groups, distributional properties of a correlation coefficient, testing covariance matrices, estimating overlap of distributions for the purpose of assessing similarity, and a description of the datasets used in this dissertation.

2.5.1 Simulating HD Data

There are challenges in simulating HD data for simulation studies. Researchers are often faced with the problem of simulating data that contain properties of real life data. Even though the distributions can be assumed about the data (e.g. multivariate distribution), simulations require additional information. The main issue is to build the covariance matrix for the simulations. For example, in order to simulate multivariate normal data, the covariance matrix is required to be a positive definite matrix. Although the covariance matrices can be estimated using the pairwise correlations in observed data, they do not result in positive definite matrices in HD situations. Therefore, most of the simulations that exist in the literature are limited to block diagonal matrices with positive covariance values. These matrices do not reflect the more complex dependence structures that are present in real datasets.

Gadbury et al. (2008) presented a method to simulate data that are more realistic than the data that can be generated with statistical distributions. They presented an algorithm to manipulate a real dataset in a way that the true number of null hypotheses is known and the covariance structure is preserved. The basic idea of the algorithm is as follows,

- Record the standardized effect sizes between the two groups

- Randomly divide the control group into two equal size groups (plasmode treatment groups and plasmode control group)
- Add a selected number of effects to the plasmode treatment group.

The exact number of hypotheses for which the null is true and their indices are known in these plasmode datasets. These datasets can be used to evaluate statistical methods, and since the true number of hypotheses for which the alternative is true is known, quantities like false discovery proportions can be directly computed.

2.5.2 Changes in Dependence Structure in Different Treatment Groups

Southworth et al (2009) investigated the changes in the dependence structures in groups of mice and presented that the correlation structure of the gene expression levels may change with the age of the mice. The only difference between the two groups of mice in their experiment was their age and no treatment was given. They showed that the strength of the relationship between genes in older mice is weaker compared to younger mice. They used a differential co-expression network to identify complex co-regulation of the genes as the mice age. This method itself is different from that regularly used to identify co-expression networks known as the differential clustering algorithm (Ihmels et al, 2005). Using a differential co-expression network analysis they showed that there is modular loosening of the correlation difference network in the older mice compared to the younger mice.

This suggests that the correlation structures of groups of samples that are subjected to different treatment conditions may be different. These can be differences between two phenotypic groups like in this case or differences due to a treatment applied to one group. This emphasizes that the assumption of equal dependence structure in two (or more) groups may be a strong assumption that is often violated.

2.5.3 Distribution of the Correlation Coefficient

The distribution of the sample correlation coefficient is discussed in detail in Stuart and Ord (2009). If x and y have a bivariate normal distribution with correlation coefficient ρ, then the sample correlation coefficient between x and y, r is given by

$$f(r \mid \rho) = \frac{(1-\rho^2)^{\frac{1}{2}(n-1)}}{\pi\,\Gamma(n-2)}(1-r^2)^{\frac{1}{2}(n-4)}\frac{d^{n-2}}{d(\rho r)^{n-2}}\left(\frac{\arccos(-\rho r)}{\sqrt{1-\rho^2 r^2}}\right) \qquad -1 \le r \le 1 \qquad (2.17)$$

When x and y are independent ($\rho = 0$), then the distribution simplifies to

$$f(r \mid \rho = 0) = \frac{1}{\mathrm{B}\left\{\frac{1}{2},\frac{1}{2}(n-2)\right\}}(1-r^2)^{\frac{1}{2}(n-4)} \qquad -1 \le r \le 1 \qquad (2.18)$$

This density can be used to derive the density of the combined correlation under independence introduced in Chapter 4.

2.5.4 Testing Equality of Correlation Matricess in HD Data

Srivastava and Yanagihara (2010) and Schott (2007) suggested methods to test the equality of covariance matrices in the HD data. The test statistic suggested by Schott is

$$t_{mn} = \sum_{i<j} tr\{(S_i - S_j)^2\}$$

where S_i *and* S_j are estimated covariance matrices and the test is based on the normal distribution. Srivastava and Yanagihara presented multiple testing procedures that perform well under different conditions of dimensionality of the data. Both of these methods rely on calculating the covariance matrix from the data. Although elements of the covariance of the matrix can be estimated from the data, they rarely result in positive definite matrices. For the purposes of this work, positive definiteness of the covariance matrices is not needed.

2.5.5 Estimations of Overlap of Distributions

Stine and Heyse (2001) suggested a non-parametric alternative to the measures of proportion of similar responses (PSR) introduced in Bradley (1985) and the area between curves (ABC) introduced in Bohning et al (1992). Assuming that the density functions of the two distributions being compared are denoted by f and g, these measure are given by

$$PSR(f,g) = \int \min[f(x), g(x)]dx \qquad (2.19)$$

and

$$ABC(f,g) = \int \left|f(x) - g(x)\right| dx \qquad (2.20)$$

The relationship between the proportion of similar responses and the area between the curve is given by

$$PSR(f,g) = 1 - \tfrac{1}{2} ABC(f,g) \tag{2.21}$$

Parametric estimates of these quantities were derived assuming normal distributions for f and g. The non-parametric estimates were obtained by using kernel density estimates of unknown densities f and g through observed data given by

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - X_i}{h}\right) \tag{2.22}$$

where $K(.)$ is a kernel and h is the bandwidth. The publication only considered symmetric kernels and these estimates were assessed using simulated data.

2.6 Datasets

Two microarray datasets are used for illustrating the methods discussed in this dissertation. A brief introduction to the background of the datasets is given below.

2.6.1 Lung Cancer Dataset

A microarray dataset of 192 individuals used in Spira et al (2007) is used for the two group comparison. The original dataset consists of 97 individuals with cancer, 90 individuals without cancer and 5 individials suspected of having cancer. Spira et al used a repetitive internal cross validation method to select the genes that are differentially expressed between cancer and no cancer groups to develop a classification method. Gene profiles were taken from Affymetrix HG-U133A microarrays. There are 12687 gene probes after normalization and background correction. Probe level data were summarized using the RMA algorithm, mentioned earlier.

2.6.2 Multiple Myeloma and Bone Lesion Dataset

A microarray dataset of 173 individuals used to compare the gene expression levels in multiple myeloma patients with and without bone lesions and was introduced in Tian et al (2003). There are 137 subjects with bone lytic lesions and 36 subjects without bone lytic lesions. U95Av2 microarrays were used in obtaining probe level data. Data were summarized using the

MAS-5 method. Tian et al used a logistic regression analysis to model presence or absence of bone lytic lesions in the subjects. Initial filtration applied to the data in this work reduced the number of probes to 3970. These data will be used for illustrations in this dissertation.

These first two chapters introduced the problem large scale simultaneous testing and some of the work done to address issues that arise in this area. The following chapters characterize the problem of high variance in large scale simultaneous testing under dependencies and devise a possible method to reduce the effects of the correlations of the data when conducting two group mean comparisons.

Chapter 3 - Two Group Comparisons

3.1 Overview of Large-scale Two Group Comparisons

Two group mean comparison is a common and basic analysis that is used to test the equality of two population means. This method is commonly applied to analyze large scale datasets with some assumptions about the data and the dependence structures. Methods for controlling or estimating type I errors, FWER or FDR are also employed in applications with assumptions. These methods were discussed earlier and many assume independence of data. The methods that allow dependencies include certain restrictions. The main purpose of this chapter is to gain further understanding into the problem of large scale testing under different dependence structures. The effect of the dependencies on the number of discoveries and the false discovery rate is first illustrated using a series of simulated data. A derivation of the variance of the number of discoveries is given in Chapter 4 along with an examination of the properties of the conditional density of correlated test statistics.

Some of the following results can be analytically proved and further verified by simulations while some of the results have to rely on the results of simulations. A proof is provided where possible and followed by a verifying simulation. For the cases where an analytical proof is impossible, simulations or computational methods are used for validation. The t-test used throughout this dissertation is the unequal variance t-test with Welch correction.

3.2 Mean and Variance of the Number of Discoveries

The notation used in this dissertation is briefly described below. The number of features under investigation is denoted by K. The features can be any observations in a HD dataset (e.g. genes in a microarray study, compound abundance levels in a mass spectrometer run). K is the final number of features in the study after the preprocessing and cleaning. In general, assume that subjects in the study fall into G distinct non-overlapping groups. The theory will be developed for the two group comparison case (i.e., $G = 2$). The number of individuals in each groups is denoted by n_g ($g = 1, \ldots, G$). The matrix of feature levels (normalized, background/baseline corrected) is denoted by X. The observed value for the i^{th} feature of the k^{th} individual in the g^{th} group is denoted by x_{igk}. The level of significance used for each individual test is denoted by α. Let p_i denote the p-value obtained by a performing a two-group comparison test for the i^{th} feature

and let x_{ig} denote the vector of expression levels obtained for the ith feature of all individuals in the gth group.

The effect of dependencies among features on the two group comparison test is demonstrated below. Assume that K independent size α tests are conducted, one for each feature. A notation similar to the one used in Hu et al., (2010) will be used below, and that is similar to that used in Schweder and Spjotvoll (1982). Assume that the global null hypothesis is true. Throughout this dissertation the global null hypothesis is defined as none of the features being tested are different between the two groups. Let D_i be the indicator variable that takes the values 0 or 1 such that

$$D_i = \begin{cases} 1 & if\ p_i \le \alpha \\ 0 & if\ p_i > \alpha \end{cases} \tag{3.1}$$

The number of significant discoveries, N_α^1 is given by

$$N_\alpha^1 = \sum_{i=1}^{K} D_i \tag{3.2}$$

Under the true global null hypothesis, the expected number of discoveries and the variance of the number of discoveries can be derived using the distribution of N_α^1 which is the sum of K Bernoulli trials with α probability of success. The expected number of discoveries is given by

$$E_{H_0}(N_\alpha^1) = E_{H_0}\left(\sum_{i=1}^{K} D_i\right) = \sum_{i=1}^{K} E_{H_0}(D_i) = \sum_{i=1}^{K} E_{H_0}\left[I(p_i \le \alpha)\right] = \sum_{i=1}^{K} \alpha = K\alpha \tag{3.3}$$

The variance of the number of discoveries is given by

$$V_{H_0}(N_\alpha^1) = V_{H_0}\left(\sum_{i=1}^{K} D_i\right) = K\alpha(1-\alpha) + \sum_{i=1}^{K} \sum_{j=1(i\neq j)}^{K} Cov(D_i, D_j) \tag{3.4}$$

Note that subscript H_0 in (3.3) and (3.4) indicate the expectation and the variance are calculated under the true global null hypothesis. If the Bernoulli trials were independent, $Cov(D_i, D_j) = 0$ for all i and j. However, as (3.4) indicates, the dependencies among individual tests may cause the variance of the number of rejections to increase. In HD data K is generally large and the sum of all pairwise correlations can become large in value.

Hu et al., (2010) evaluated the expression in (3.4) using bivariate normal distribution as the joint density of the two test statistics of interest as a means to calculate $Cov(D_i, D_j)$. In addition, in their simulations they used the same correlation values between all pairs of test statistics within a block diagonal structure for a HD correlation matrix. The proof in section 4.1 illustrates the relationship between the correlation of two test statistics and the correlation in the data. The form in Hu et al (2010) matches this work under strong assumptions on equal variances and same correlation in both groups. The derivation in this work allows the two groups in the test to have different correlation structures. This covariance between the test statistics cannot be directly calculated using an observed set of data.

A simulation study is used to illustrate the increment of the variance under different correlation structures. This simulation will show that the variance of the number of rejections increases as the dependence structure of the data increases. A description of the simulation is given below.

Simulation 1

A series of simulated datasets will be used to demonstrate the effect of correlations on the variance of number of discoveries. Only positive correlations will be used in the simulation since the methods used to generate data require positive definite covariance matrices. It will be demonstrated in section 4.2 that the negative correlations between two features have a similar effect to a positive correlation of the same magnitude.

Data were simulated using multivariate normal distributions. Data were generated with 40 individuals in each of two groups. 500 features are generated for each individual. ($K = 500$, $G = 2$, $n_1 = n_2 = 40$), and features were independent across individuals. The dependence structure between the features is varied to demonstrate the effect on the variance. It is not assumed that both groups have the same dependence structure among the features. Instead, the two groups are allowed to have different correlation values between pairs of features. Block diagonal correlation matrices were generated with all off-block elements equal to zero and the diagonal elements equal to one. The off diagonal elements within the blocks are set to different correlation values and the size of the blocks is also varied. Each correlation matrix is a 500×500 matrix. The block sizes chosen for this are 10, 20, 50 and 100. The correlation values varied from 0 to 0.9 with 0.1

increments. Data for the two groups were generated using all cross combinations of these values. Multivariate normal data are generated with a mean vector of zeros and these correlation matrices. A two-group t-test is performed for each feature and the number of discoveries is recorded. This process is repeated 100 times per each dependence structure and the variance of the 100 number of discoveries (significance level = 0.05) is calculated. The use of zero mean vectors imposes the condition that none of the features are different between the groups, thus the data are simulated under the global null hypothesis.

The colors in Figure 3.1 indicate the variance of the number of discoveries (Blue: smaller variance, Red: larger variance). This figure illustrates the increment in variance of number of discoveries as the dependence structure increases. As Figure 3.1 shows, when the features in a HD dataset are correlated, the discoveries from a series of regular t-tests become unreliable. Equation (3.4) shows that the increment of the variance is due to the increasing dependence structure among features. The relationship between the covariance and the probability of rejecting a test statistic under the global null hypothesis can be derived by the following.

$$\begin{aligned} Cov\ (D_i, D_j) &= E[D_i\ D_j] - E[D_i] \cdot E[D_j] \\ &= P[D_i = 1, D_j = 1] - P[D_i = 1] \cdot P[D_j = 1] \\ &= P[D_i = 1, D_j = 1] - \alpha^2 \end{aligned} \tag{3.5}$$

The expectation in (3.5) can be further simplified as

$$\begin{aligned} P[D_i = 1, D_j = 1] &= P[p_i < \alpha,\ p_j < \alpha] \\ &= P[p_i < \alpha \mid p_j < \alpha] \cdot P[p_j < \alpha] \\ &= P[p_i < \alpha \mid p_j < \alpha] \cdot \alpha \end{aligned} \tag{3.6}$$

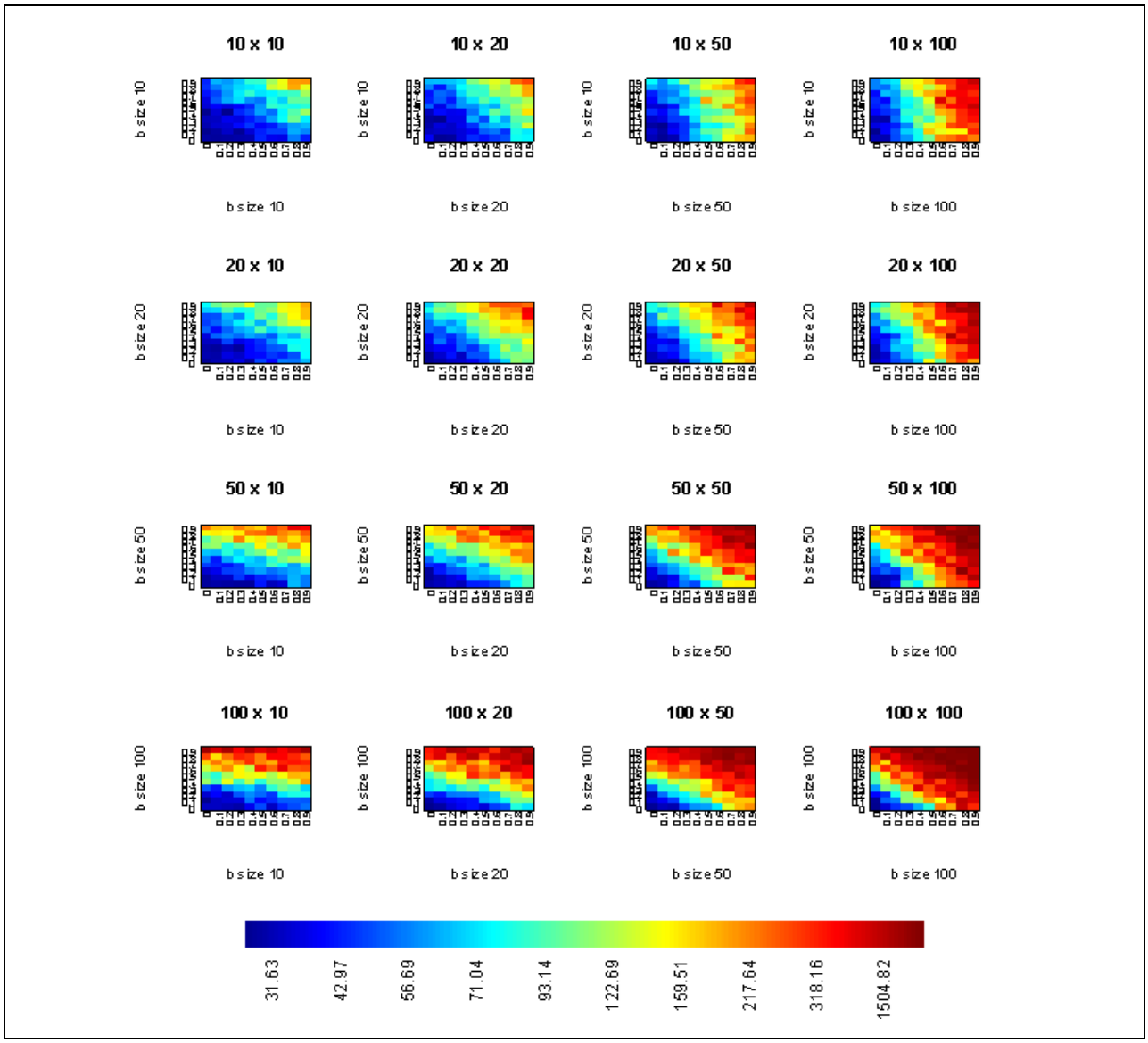

Figure 3.1: Effect of increasing dependence structure on the variance of the number of discoveries.
Each sub plot is for specific block sizes for the two groups. X and Y axes in each plot indicate the correlation values used in blocks. The colors of the plot indicate the magnitude of the variance with specific dependence structures. The plots show that the variance of the number of discoveries increases as the strength of the dependence structure increases.

Assuming that a series of test statistics with symmetric distributions about zero were used for the tests, the probability in (3.6) can be expressed as

$$P[p_i < \alpha \mid p_j < \alpha] = P[z_i < -z^* \ OR \ z_i > z^* \mid z_j < -z^* \ OR \ z_j > z^*]$$

$$= P\left(\frac{\bar{x}_i - \bar{y}_i}{\sqrt{\sigma_i^2 + \tau_i^2}} < -z^* \; OR \; \frac{\bar{x}_i - \bar{y}_i}{\sqrt{\sigma_i^2 + \tau_i^2}_i} > z^* \middle| \frac{\bar{x}_j - \bar{y}_j}{\sqrt{\sigma_j^2 + \tau_j^2}} < -z^* \; OR \; \frac{\bar{x}_j - \bar{y}_j}{\sqrt{\sigma_j^2 + \tau_j^2}} > z^* \right) \qquad (3.7)$$

Where $-z^*$ and z^* are lower and upper cutoff values for the test statistics for α level for two sided tests. It is worth noting that if the expression levels of i^{th} and j^{th} features are independent, the probability in (3.7) reduces to α, resulting in a $E[D_i = 1, D_j = 1] = \alpha^2$ in (3.5) and the covariance given in (3.5) will be zero, thus removing the effect of the covariance from the variance calculation in (3.4).

Chapter 4 further explores the conditional probability given in (3.7) and derives the conditional density of test statistics assuming the bivariate normality of the sample means. This distribution is used for testing features in the method introduced in Chapter 6.

Chapter 4 - Conditional Density of Test Statistics and the Variance of the number of Discoveries

4.1 Conditional Density of Test Statistics

In order to obtain an analytical form for the probability in (3.7) it is assumed that the sample means of the two features of interest within a group have a bivariate normal distribution. It is not assumed that the correlations between the two features are the same for the two groups. In addition, the variances are allowed to be different between features and groups. This allows the concept of different dependence structure between the two groups, or the treatment altering the dependence structure among features.

Let x_i and x_j be vectors of expression levels for features i and j in group 1. Let $\bar{x}_i$ and $\bar{x}_j$ denote their means and $\rho_{x_{ij}}$ denote the correlation between ith and jth sample means, so that the joint density can be written as

$$\begin{pmatrix} \bar{x}_i \\ \bar{x}_j \end{pmatrix} \sim N\left(\begin{bmatrix} \mu_i \\ \mu_j \end{bmatrix}, \begin{bmatrix} \sigma_i^2 & \rho_{x_{ij}}\sigma_i\sigma_j \\ \rho_{x_{ij}}\sigma_i\sigma_j & \sigma_j^2 \end{bmatrix} \right) \tag{4.1}$$

Similarly, let $\bar{y}_i$ and $\bar{y}_j$ be means of feature levels i_i and j of group 2, and

$$\begin{pmatrix} \bar{y}_i \\ \bar{y}_j \end{pmatrix} \sim N\left(\begin{bmatrix} \nu_i \\ \nu_j \end{bmatrix}, \begin{bmatrix} \tau_i^2 & \rho_{y_{ij}}\tau_i\tau_j \\ \rho_{y_{ij}}\tau_i\tau_j & \tau_j^2 \end{bmatrix} \right) \tag{4.2}$$

For the simplicity of notation the second level subscript ij is dropped from $\rho_{x_{ij}}$ and $\rho_{y_{ij}}$ and will be written as ρ_x and ρ_y. Technically these two correlations carry indices i, j, to indicate that they are specific to a particular pair and are thus allowed to be different for every pair. The suppression of the double subscripts helps to simplify notation in later derivations. It is assumed that expression levels x obtained for group 1 are independent of the expression levels y in group 2. This is a valid assumption since they are from non-overlapping groups of subjects. Using this, the joint distribution of the sample means with the simplified notation can be written as

$$\begin{pmatrix} \bar{x}_i \\ \bar{x}_j \\ \bar{y}_i \\ \bar{y}_j \end{pmatrix} \sim N\left(\begin{bmatrix} \mu_i \\ \mu_j \\ \nu_i \\ \nu_j \end{bmatrix}, \begin{bmatrix} \sigma_i^2 & \rho_x\sigma_i\sigma_j & 0 & 0 \\ \rho_x\sigma_i\sigma_j & \sigma_j^2 & 0 & 0 \\ 0 & 0 & \tau_i^2 & \rho_y\tau_i\tau_j \\ 0 & 0 & \rho_y\tau_i\tau_j & \tau_j^2 \end{bmatrix} \right) \tag{4.3}$$

A transformation on (4.3) is used to obtain the distribution of test statistics

Define

$$W = \begin{pmatrix} \bar{x}_i \\ \bar{x}_j \\ \bar{y}_i \\ \bar{y}_j \end{pmatrix} \qquad \delta = \begin{bmatrix} \mu_i \\ \mu_j \\ \nu_i \\ \nu_j \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma_i^2 & \rho_x\sigma_i\sigma_j & 0 & 0 \\ \rho_x\sigma_i\sigma_j & \sigma_j^2 & 0 & 0 \\ 0 & 0 & \tau_i^2 & \rho_y\tau_i\tau_j \\ 0 & 0 & \rho_y\tau_i\tau_j & \tau_j^2 \end{bmatrix}$$

Then

$$W \sim N(\delta, \Sigma)$$

By taking

$$b = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, bW \text{ results}$$

$$bW \sim N(b\delta, b\Sigma b^T)$$

$$W' = bW = \begin{pmatrix} \bar{x}_i - \bar{y}_i \\ \bar{x}_j - \bar{y}_j \end{pmatrix} \qquad \delta' = b\delta = \begin{bmatrix} \mu_i - \nu_i \\ \mu_j - \nu_j \end{bmatrix}$$

$$\Sigma' = b\Sigma b^T = \begin{bmatrix} \sigma_i^2 + \tau_i^2 & \rho_x\sigma_i\sigma_j + \rho_y\tau_i\tau_j \\ \rho_x\sigma_i\sigma_j + \rho_y\tau_i\tau_j & \sigma_j^2 + \tau_j^2 \end{bmatrix}$$

Then

$$W' \sim N(\delta', \Sigma')$$

By taking

$$c = \begin{bmatrix} 1/\sqrt{\sigma_i^2 + \tau_i^2} & 0 \\ 0 & 1/\sqrt{\sigma_j^2 + \tau_j^2} \end{bmatrix}, \; cW' \text{ results}$$

$$cW' \sim N(c\delta', c\Sigma' c^T)$$

$$W'' = cW' = \begin{pmatrix} z_i = \dfrac{\bar{x}_i - \bar{y}_i}{\sqrt{\sigma_i^2 + \tau_i^2}} \\ \\ z_j = \dfrac{\bar{x}_j - \bar{y}_j}{\sqrt{\sigma_j^2 + \tau_j^2}} \end{pmatrix} \qquad \delta'' = b\delta' = \begin{bmatrix} \dfrac{\mu_i - \nu_i}{\sqrt{\sigma_i^2 + \tau_i^2}} \\ \\ \dfrac{\mu_j - \nu_j}{\sqrt{\sigma_j^2 + \tau_j^2}} \end{bmatrix}$$

$$\Sigma'' = c\Sigma' c^T = \begin{bmatrix} \dfrac{\sigma_i^2 + \tau_i^2}{\sigma_i^2 + \tau_i^2} & \dfrac{\rho_x \sigma_i \sigma_j + \rho_y \tau_i \tau_j}{\sqrt{(\sigma_i^2 + \tau_i^2)(\sigma_j^2 + \tau_j^2)}} \\ \\ \dfrac{\rho_x \sigma_i \sigma_j + \rho_y \tau_i \tau_j}{\sqrt{(\sigma_i^2 + \tau_i^2)(\sigma_j^2 + \tau_j^2)}} & \dfrac{\sigma_j^2 + \tau_j^2}{\sigma_j^2 + \tau_j^2} \end{bmatrix} = \begin{bmatrix} 1 & \rho_z \\ \rho_z & 1 \end{bmatrix}$$

where
$$\rho_{z_{ij}} = \frac{\rho_x \sigma_i \sigma_j + \rho_y \tau_i \tau_j}{\sqrt{(\sigma_i^2 + \tau_i^2)(\sigma_j^2 + \tau_j^2)}} \tag{4.4}$$

and
$$W'' \sim N(\delta'', \Sigma'') \tag{4.5}$$

Similar to ρ_x and ρ_y above, the second level subscripts *ij* will be dropped from (4.4) to simplify the notation, that is ρ_z also carries subscripts *ij* indicating it is different for different pairs of features considered. The density given in (4.5) describes the distribution of two correlated test statistics with variances that are assumed known. The current work is based on the assumption that these covariances are known or can be accurately estimated from the data. Equation (4.4) gives the relationship between the dependence structure of the data and the correlation of the two

test statistics. Under the global null hypothesis, the i[th] and j[th] features are assumed to have a zero mean difference,

$$\mu_i - \nu_i = 0\,,\ \mu_j - \nu_j = 0 \text{ and } \delta'' = [0,0]^T = 0\,,$$

and

$$W'' \overset{H_0}{\sim} N(0, \Sigma'') \tag{4.6}$$

The probability given in (3.7) must be obtained through the conditional distribution of a test statistic conditioned on rejection of another correlated test statistic. Let $f_{z_j|z_i<-z^* \text{ or } z_i>z^*}$ denote the distribution of z_j, the test statistic for the j[th] test conditioned on the rejection of i[th] test. Then

$$f_{z_j|z_i<-z^* \text{ or } z_i>z^*}(z_j \mid z_i < -z^* \text{ or } z_i > z^*) = \frac{\int_{-\infty}^{-z^*} f_{z_i z_j}(z_i, z_j)dz_i + \int_{z^*}^{\infty} f_{z_i z_j}(z_i, z_j)dz_i}{\int_{-\infty}^{-z^*} f_{z_i}(z_i)dz_i + \int_{z^*}^{\infty} f_{z_i}(z_i)dz_i} \tag{4.7}$$

The joint density $f_{z_i z_j}$ is given in (4.6) and $f_{z_i}(z_i)$ is the marginal distribution of z_i which is a univariate standard normal distribution under the global null hypothesis. The integral in the denominator of (4.7) is a normal probability integral given by

$$\int_{-\infty}^{-z^*} f_{z_i}(z_i)dz_i + \int_{z^*}^{\infty} f_{z_i}(z_i)dz_i = \int_{-\infty}^{-z^*} \phi(z_i)dz_i + \int_{z^*}^{\infty} \phi(z_i)dz_i = \Phi(-z^*) + 1 - \Phi(z^*) = 2\Phi(-z^*) \tag{4.8}$$

where $\phi(z)$ and $\Phi(z)$ are standard normal probability and cumulative distribution functions evaluated at z, respectively.

The first term in the numerator of (4.7) can be simplified as

$$\begin{aligned}
\int_{-\infty}^{-z^*} f_{z_i z_j}(z_i, z_j)dz_i &= \int_{-\infty}^{-z^*} \frac{1}{2\pi\sqrt{1-\rho_z^2}} \exp\left\{\frac{-1}{2(1-\rho_z^2)}[z_i^2 - 2\rho_z z_i z_j + z_j^2]\right\}dz_i \\
&= \int_{-\infty}^{-z^*} \frac{1}{2\pi\sqrt{1-\rho_z^2}} \exp\left\{\frac{-1}{2(1-\rho_z^2)}[z_i^2 - 2\rho_z z_i z_j + (\rho_z z_j)^2 - (\rho_z z_j)^2 + z_j^2]\right\}dz_i \\
&= \int_{-\infty}^{-z^*} \frac{1}{2\pi\sqrt{1-\rho_z^2}} \exp\left\{\frac{-1}{2(1-\rho_z^2)}[(z_i - \rho_z z_j)^2 + z_j^2(1-\rho_z^2)]\right\}dz_i \\
&= \int_{-\infty}^{-z^*} \frac{1}{2\pi\sqrt{1-\rho_z^2}} \exp\left\{\frac{-1}{2(1-\rho_z^2)}(z_i - \rho_z z_j)^2\right\} \exp\left\{\frac{-1}{2(1-\rho_z^2)} z_j^2(1-\rho_z^2)\right\}dz_i \\
&= \int_{-\infty}^{-z^*} \frac{1}{2\pi\sqrt{1-\rho_z^2}} \exp\left\{\frac{-1}{2(1-\rho_z^2)}(z_i - \rho_z z_j)^2\right\} \exp\left\{\frac{-1}{2(1-\rho_z^2)} z_j^2(1-\rho_z^2)\right\}dz_i \\
&= \int_{-\infty}^{-z^*} \frac{1}{\sqrt{2\pi}\sqrt{1-\rho_z^2}} \exp\left\{\frac{-1}{2(1-\rho_z^2)}(z_i - \rho_z z_j)^2\right\}dz_i \frac{1}{\sqrt{2\pi}} \exp\left\{\frac{-1}{2} z_j^2\right\}
\end{aligned}$$

By a variable transformation of $x = \dfrac{z_i - \rho_z z_j}{\sqrt{1-\rho_z^2}}$

$$\int_{-\infty}^{-z^*} f_{z_i z_j}(z_i, z_j)dz_i = \int_{-\infty}^{\frac{-z^* - \rho_z z_j}{\sqrt{1-\rho_z^2}}} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\}dz_i \phi(z_j) = \phi(z_j)\Phi\left[\frac{-z^* - \rho_z z_j}{\sqrt{1-\rho_z^2}}\right] \qquad (4.9)$$

The second term in the numerator of (4.7) can be obtained in a similar fashion

$$\int_{z^*}^{\infty} f_{z_i z_j}(z_i, z_j)dz_i = \int_{\frac{z^* - \rho_z z_j}{\sqrt{1-\rho_z^2}}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\}dz_i \phi(z_j) = \phi(z_j)\left[1 - \Phi\left(\frac{z^* - \rho_z z_j}{\sqrt{1-\rho_z^2}}\right)\right] \qquad (4.10)$$

By combining (4.8), (4.9) and (4.10) the density in (4.7) can be written as

$$f_{z_j | z_i < -z^* \text{ or } z_i > z^*}(z_j \mid z_i < -z^* \text{ or } z_i > z^*) = \frac{\phi(z_j)\left[1 + \Phi\left(\dfrac{-z^* - \rho_z z_j}{\sqrt{1-\rho_z^2}}\right) - \Phi\left(\dfrac{z^* - \rho_z z_j}{\sqrt{1-\rho_z^2}}\right)\right]}{2\Phi(-z^*)} \qquad (4.11)$$

The probability given in (3.7) is an integral of the density given in (4.11) over the combined interval [(-∞, -z*), (z*,∞)]. A closed -form solution does not exist for these integrals but numerical methods can be employed to obtain the resulting values.

If the two features z_i and z_j are independent, the correlation given in (4.4) becomes zero ($\rho_z = 0$) and the density given in (4.11) simplifies to a standard normal density

$$f_{z_j|z_i<-z^* \text{ or } z_i>z^*}(z_j \mid z_i < -z^* \text{ or } z_i > z^*, \rho_z = 0) = \frac{\phi(z_j)\left[1+\Phi\left(\frac{-z^*-0\cdot z_j}{\sqrt{1-0}}\right)-\Phi\left(\frac{z^*-0\cdot z_j}{\sqrt{1-0}}\right)\right]}{2\Phi(-z^*)}$$

$$= \frac{\phi(z_j)\left[1+\Phi(-z^*)-\Phi(z^*)\right]}{2\Phi(-z^*)} = \frac{\phi(z_j)2\Phi(-z^*)}{2\Phi(-z^*)}$$

$$= \phi(z_j)$$

Thus, under independence the probability in (3.7) is equal to α, and the covariance in (3.5) is equal to zero.

Let $\overline{P}_{z^*}(i,j)$ denote the probability given in (3.7) for the ith feature conditioned on the jth feature. Then

$$\overline{P}_{z^*}(i,j) = \int_{-\infty}^{-z^*} f_{z_j|z_i<-z^* \text{ or } z_i>z^*}(z_j \mid z_i < -z^* \text{ or } z_i > z^*)\,dz_j + \int_{z^*}^{\infty} f_{z_j|z_i<-z^* \text{ or } z_i>z^*}(z_j \mid z_i < -z^* \text{ or } z_i > z^*)\,dz_j$$

The variance expression in (3.4) can be expressed as

$$V_{H_0}(N_\alpha^1) = V_{H_0}\left(\sum_{i=1}^{K} D_i\right) = K\alpha(1-\alpha) + \sum_{i=1}^{K}\sum_{j=1}^{K}[\alpha\,\overline{P}_{z^*}(i,j) - \alpha^2] \tag{4.12}$$

As discussed above, if the features are independent, then the conditional distribution given in (4.11) becomes a standard normal distribution, resulting $\overline{P}_{z^*}(i,j) = \alpha$. This will reduce the variance in (4.12) to

$$V_{H_0}(N_\alpha^1) = V\left(\sum_{i=1}^{K} D_i\right) = K\alpha(1-\alpha) + \sum_{i=1}^{K}\sum_{j=1}^{K}[\alpha\,\overline{P}_{z^*}(i,j) - \alpha^2]$$

$$= K\alpha(1-\alpha) + \sum_{i=1}^{K}\sum_{j=1}^{K}[\alpha^2 - \alpha^2] = K\alpha(1-\alpha)$$

which is the variance of the number of discoveries under independence of the features. A simulation can be used to demonstrate that the variance given in (4.12) is in fact the variance of the number of rejections under the global null hypothesis.

Simulation 2

Only six block sizes and correlations combinations from simulation 1, representing weak to strong correlations structures are selected for this simulation. All off diagonal elements within a block is set to a selected correlation value indicated in Table 4.1

Table 4.1 Dependence Structures for Simulation 2

Label	Group 1		Group 2	
	Block Size	Correlation	Block Size	Correlation
A	10	0	10	0
B	10	0.2	10	0.2
C	10	0.2	10	0.4
D	10	0.2	50	0.4
E	50	0.5	50	0.8
F	50	0.5	100	0.8

Data are generated for two groups with 40 individuals in each group ($n_1 = n_2 = 40$) and 500 features per individual ($K = 500$). Zero mean vectors are used for generating data, implying no difference between the two groups. Data for each correlation structure in Table 4.1 are simulated 200 times and the number of rejections (tested at $\alpha = 0.05$) of the null hypothesis is recorded. These values are used to obtain the variance of the number of rejections.

The procedure described above is repeated 100 times to obtain the sampling variability of the variance estimates. A triangle indicates the variance obtained by (4.12) for each dependence structure. A circle symbol indicates average variance based on 100 repetitions of above procedure for each correlation structure in Table 4.1. The two types of dotted lines indicate 1 standard deviation limits and 2 standard deviation limits for the estimation of variance obtained using the 100 repetitions of simulations. Figure 4.1 shows that for most cases the theoretical standard deviation stays within 1 standard deviation and all cases it stays well within 2 standard deviations. The limits shown above are simple symmetric one and two standard deviations above and below the mean variance. However, measures like quantiles of the 200 variances would likely capture the asymmetry in the sampling distribution of the variance estimator and, thus,

show that the theoretical variance is appears closer to the empirical average variance. This illustrates the behavior of the variance given in (4.12) under different dependencies. It also shows how empirical variances can be obtained via simulation in situations when theoretical derivations may not be tractable.

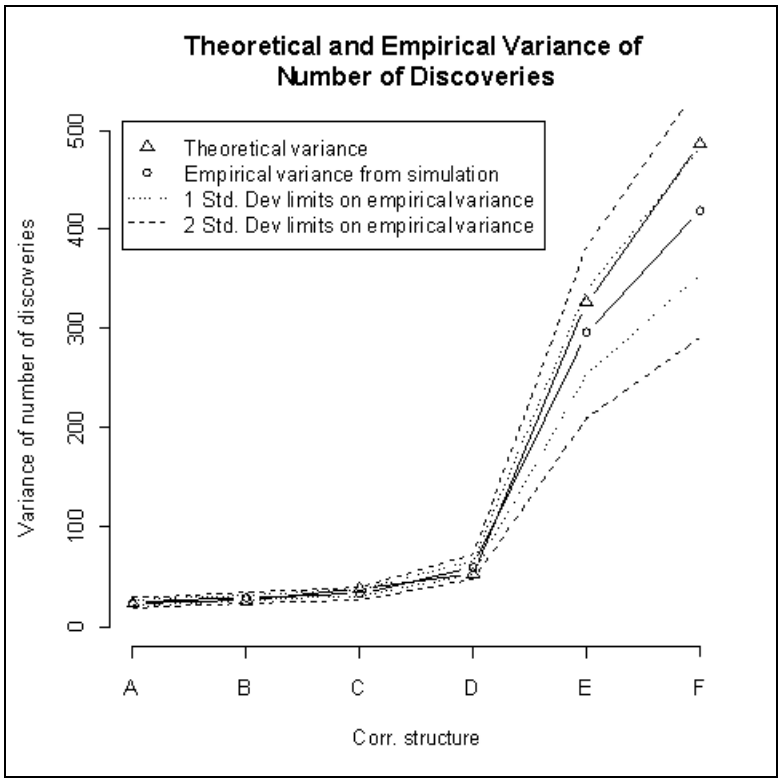

Figure 4.1: Theoretical and empirical variance of the number of discoveries.
Empirical variance is calculated from simulated data (circle symbol) and the theoretical variance (triangle symbol) is obtained by using (**4.12**) for each correlation structure given in Table **4.1**. The dashed line is the 2 standard deviation limits and the dotted line is one standard deviation limits estimated by repeated simulations. There is an agreement between the empirical and theoretical variances for weak dependence structures, and the estimates become more variable as the dependence structure increases, but are still within the simulation error. Tests were conducted at $\alpha = 0.05$.

According to (4.12) the contribution from the correlations to the increment in variance of number of rejections is through the conditional distribution given in (4.11). Although the variance of the number of discoveries is affected by the dependence structure, the expected number of discoveries remains unaffected as shown in (3.3)

4.2 Properties of the Conditional Density of Test Statistics

The distribution in (4.11) is the distribution of a test statistic conditioned on the rejection of another test statistic that is correlated with the first one. If both null hypotheses are assumed to be true, the distribution depends on two parameters, the cutoff values for the tests z^* and the correlation between the test statistics ρ_z, which is a function of correlations and variances of features levels given in (4.4). Figure 4.2 illustrates the density curves for varying values of ρ_z where z^* is fixed at 1.96 when both null hypotheses are true. (ρ_z is computed for each pair of features thus taciturnly carry subscripts *ij*)

Since a closed form for the integrals does not exist, numerical methods can be used to show that the distribution in (4.11) integrates to 1. Figure 4.2 shows the effect of the dependencies on the conditional density of the test statistic.

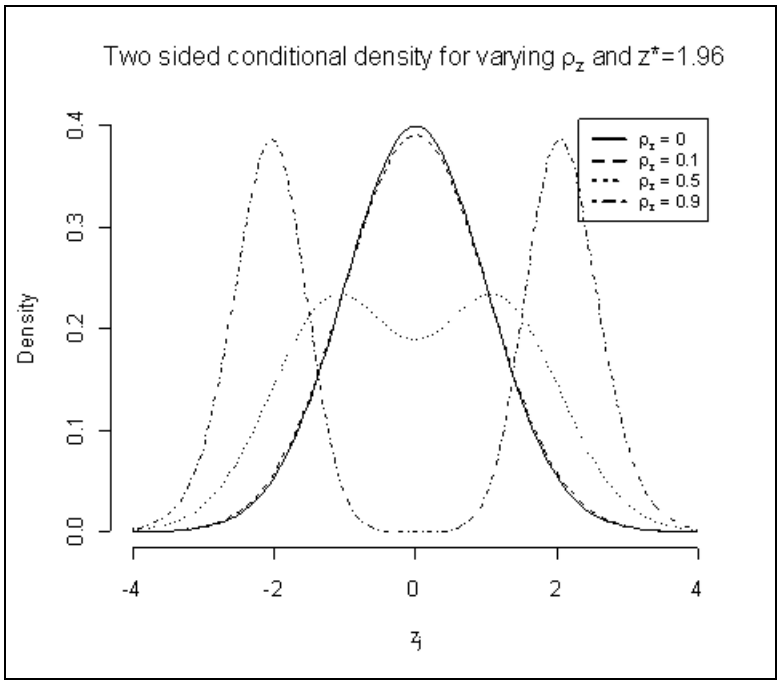

Figure 4.2: Conditional densities of test statistics.
The distribution of z_j conditioned on the rejection of z_i for varying values of ρ_z. The line for ρ_z = 0 is equivalent to the standard normal density curve.

In Figure 4.2, the curve for $\rho_z = 0$ is equivalent to the standard normal density. The deviation of the conditional distribution from the marginal density and the increment of the tail areas are visible in this figure.

Figure 4.3 shows the densities for two values of ρ_z and varying z^*. The solid line indicates the standard normal density and equivalent to the conditional density when $\rho_z = 0$. The deviation of the conditional density from the standard normal density is illustrated in this plot. The main interest is on the effect of the correlation structure on the test statistics for fixed values of z^* since the series of tests are normally performed under a fixed level of significance, α.

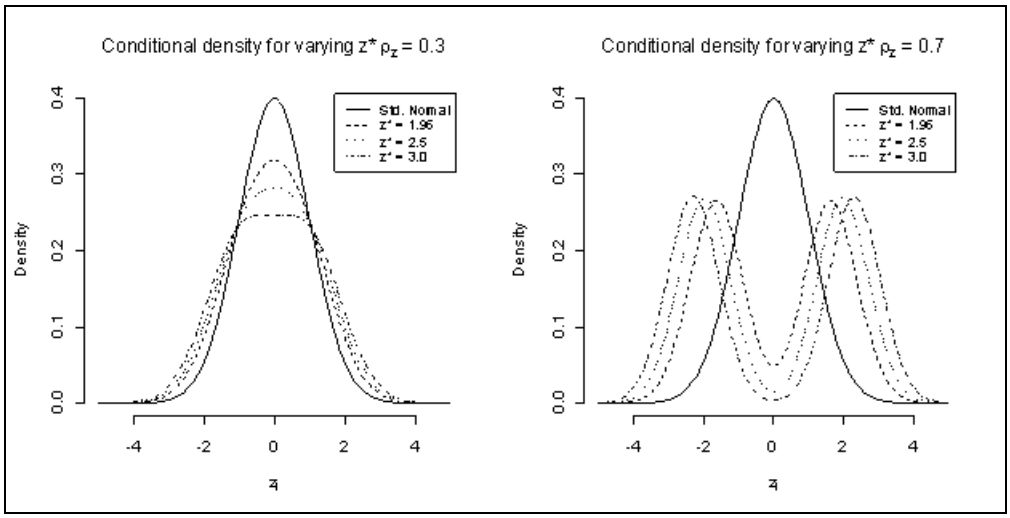

Figure 4.3: Density curves of conditional distributions given in (4.11).
Shows the density curves for different z^* values for two fixed values of ρ_z. These plots show the deviation of the conditional density from the marginal density and the increment of the tail probabilities.

Figure 4.2 and Figure 4.3 show the increment of tail probabilities of the conditional density of the test statistics under different dependence structures. A close look at (4.11) shows that the conditional distribution does not depend on the sign of the correlation. For a positive correlation,

$$f_{z_j|z_i<-z^* \text{ or } z_i>z^*}(z_j \mid z_i < -z^* \text{ or } z_i > z^*, \rho_z = \rho > 0) = \frac{\varphi(z_j)\left[1+\Phi\left(\frac{-z^*-\rho z_j}{\sqrt{1-\rho^2}}\right)-\Phi\left(\frac{z^*-\rho z_j}{\sqrt{1-\rho^2}}\right)\right]}{2\Phi(-z^*)} \quad (4.13)$$

By substituting $\rho_z = -\rho$ in (4.13)

$$f_{z_j|z_i<-z^* \text{ or } z_i>z^*}(z_j \mid z_i < -z^* \text{ or } z_i > z^*, \rho_z = -\rho < 0) = \frac{\varphi(z_j)\left[1+\Phi\left(\frac{-z^*+\rho z_j}{\sqrt{1-\rho^2}}\right)-\Phi\left(\frac{z^*+\rho z_j}{\sqrt{1-\rho^2}}\right)\right]}{2\Phi(-z^*)}$$

$$= \frac{\phi(z_j)\left[1+\left\{1-\Phi\left(\frac{z^*-\rho z_j}{\sqrt{1-\rho^2}}\right)\right\}-\left\{1-\Phi\left(\frac{-z^*-\rho z_j}{\sqrt{1-\rho^2}}\right)\right\}\right]}{2\Phi(-z^*)}$$

$$= \frac{\phi(z_j)\left[1+1-\Phi\left(\frac{z^*-\rho z_j}{\sqrt{1-\rho^2}}\right)-1+\Phi\left(\frac{-z^*-\rho z_j}{\sqrt{1-\rho^2}}\right)\right]}{2\Phi(-z^*)}$$

$$= \frac{\phi(z_j)\left[1-\Phi\left(\frac{z^*-\rho z_j}{\sqrt{1-\rho^2}}\right)+\Phi\left(\frac{-z^*-\rho z_j}{\sqrt{1-\rho^2}}\right)\right]}{2\Phi(-z^*)}$$

$$= f_{z_j|z_i<-z^* \text{ or } z_i>z^*}(z_j | z_i < -z^* \text{ or } z_i > z^*, \rho_z = \rho > 0)$$

This can be further illustrated by the shape of the density curve of (4.11) for negative and positive values of ρ_z

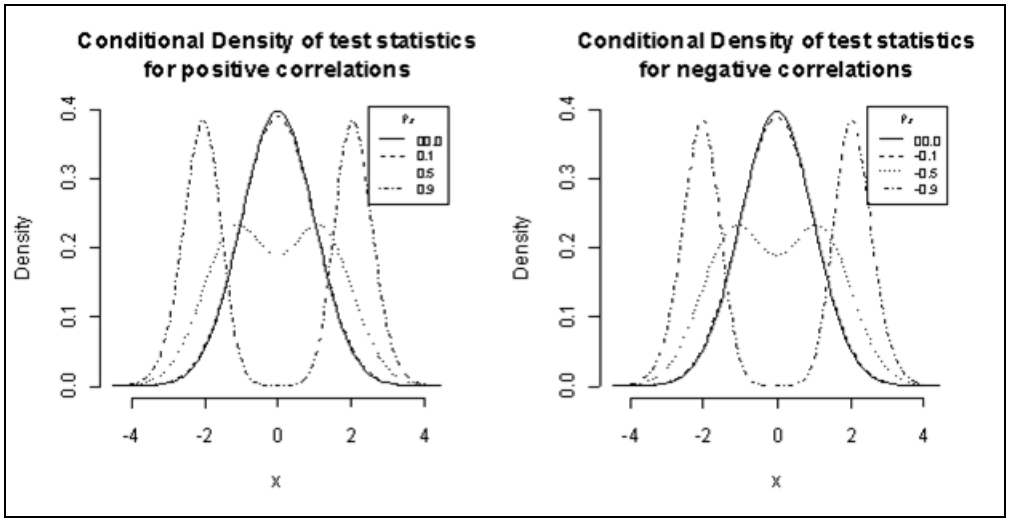

Figure 4.4: Conditional densities of test statistics under different correlations.
Figure illustrates that the negative values of ρ_z results the same density as the positive values of ρ_z. The curves on the left hand side which are for the positives correlations match the density curves for the negative correlations.

Figure 4.4 illustrates how the density given in (4.11) does not depend on the sign of the correlation but the magnitude of the correlation. This also shows that the density has the smallest tail area when the combined correlation is zero. The combined correlation can be zero for any

combination of correlations and variances that results in zero for (4.4). When the combined correlation is zero, the test statistics behave as if they were independent regardless of the correlation structure among the features. The tail areas increase for both positively and negatively correlated test statistics and the minimum tail probability above z^* is α. This implies that the probability $\overline{P}_{z^*}(i,j)$ in (4.12) is always greater than or equal to α which makes the covariance terms in (3.4) always positive. This illustrates that the dependencies in the data cause the variance of the number of discoveries to increase regardless of the type of relationship among the features, negative or positive. In addition, since the density does not depend on the sign of the correlation, the study of the effects of correlations can be limited to positive correlations. The results for the negative correlations will be implied by these results.

A derivation similar to deriving (4.11) can be used to obtain the density of a test statistic conditioned on the *failure to reject* a correlated test statistic (assuming both null hypotheses are true). The derivation results in,

$$f_{z_j|-z^*<z_i<z^*}(z_j|-z^*<z_i<z^*)=\frac{\phi(z_j)\left[\Phi\left(\frac{z^*-\rho_z z_j}{\sqrt{1-\rho_z^2}}\right)-\Phi\left(\frac{-z^*-\rho_z z_j}{\sqrt{1-\rho_z^2}}\right)\right]}{1-2\Phi(-z^*)} \qquad (4.14)$$

Figure 4.5 below shows the density curves of the distribution given in (4.14). The curve for ρ_z = 0 is equivalent to the standard normal distribution. In addition, it shows that the density has lighter tails compared to the standard normal density as the dependence structure between features increases. However, the deviation of the density from the normal distribution is subtle compared to conditioning on the rejection of a true null hypothesis.

Using a similar argument for (4.11), it can be shown that the density given in (4.14) also does not depend on the sign of the correlation but only on the magnitude of the correlation.

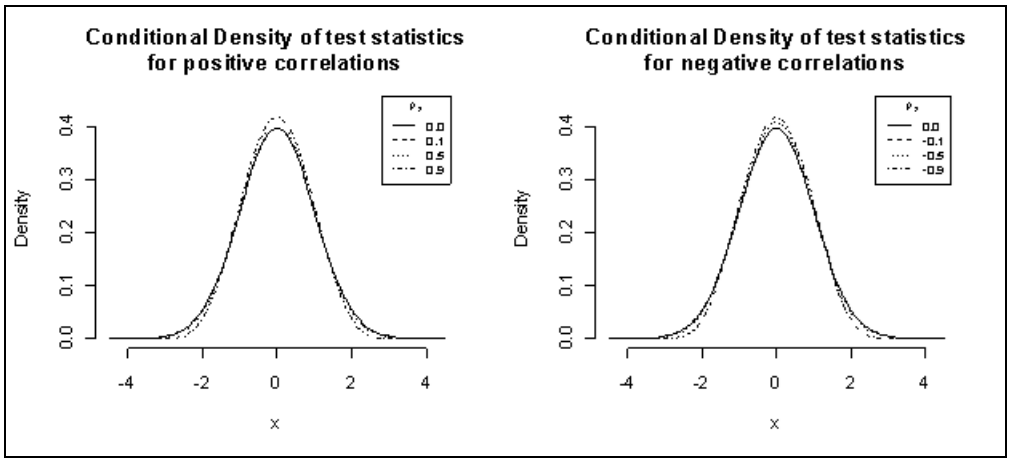

Figure 4.5: Density curves of the conditional distributions.
Curves of the density of test statistics conditioned on failure to reject a correlated test statistic when both null hypotheses are true. Both negative and positive correlations result the same density curve. The curves shown in solid lines correspond to correlation 0 and are equivalent to the standard normal density. Curves show that as the dependence structure increases, the density tend to have lighter tails.

4.3 Small Sample Results

When the sample sizes n_1 and n_2 are small in (4.1) and (4.2) the normality assumption in (4.5) maybe violated. A bivariate T-distribution may be more suitable to model the distribution of the test statistics. Then the joint distribution of the test statistics given in (4.5) can be rewritten as

$$\begin{pmatrix} t_i \\ t_j \end{pmatrix} \sim \text{bivariate T}(\nu, \rho) \tag{4.15}$$

where t_i and t_j denote the test statistics for ith and jth features being tested. ν is the degrees of freedom and ρ is the correlation, which may depend on the two features under consideration as before. Many representations exist for the form of the bivariate t distribution. This dissertation uses the form given by (4.16) below

$$f_{T_i,T_j}(t_i,t_j) = \frac{1}{2\pi\sqrt{1-\rho^2}}\left(1+\frac{t_i^2+t_j^2+2\rho t_i t_j}{\nu(1-\rho^2)}\right)^{-(\nu/2-1)} \tag{4.16}$$

$$-\infty \le t_i, t_j \le \infty$$
$$\nu > 0$$
$$-1 \le \rho \le 1$$

Assuming that the α level cutoff is t^*, the conditional density in (4.7) can be written using the joint density above

$$f_{T_j|T_i<-z^* \text{ or } T_i>z^*}(t_j \mid t_i < -t^* \text{ or } t_i > t^*) = \frac{\int_{-\infty}^{-t^*} f_{T_iT_j}(t_i, t_j)dt_i + \int_{t^*}^{\infty} f_{T_iT_j}(t_i, t_j)dt_i}{\int_{-\infty}^{-t^*} f_{T_i}(t_i)dt_i + \int_{t^*}^{\infty} f_{T_i}(t_i)dt_i} \tag{4.17}$$

However, a closed form for the integrals in (4.17) does not exist. They can be numerically evaluated to determine the shape of the distribution. The validity of the conditional density can be verified by a simulation with small samples.

Simulation 3

Data were generated using bivariate normal distribution for two features with 6 subjects per group. The correlation between the two features in the first group was set to 0.5 and it is set to 0.8 in the second group. Zero mean vectors were used implying the null hypothesis for both features is true. Data were repeatedly generated for 10^5 times and t-tests were used to test both features. The histogram below illustrates the simulated distribution of the test statistics for the second feature for which the null hypothesis was rejected for the first feature. The solid curve in the plot is the density curve given in (4.17) with 5 degrees of freedom for individual distributions being integrated. It indicates that the result in (4.17) holds for small samples.

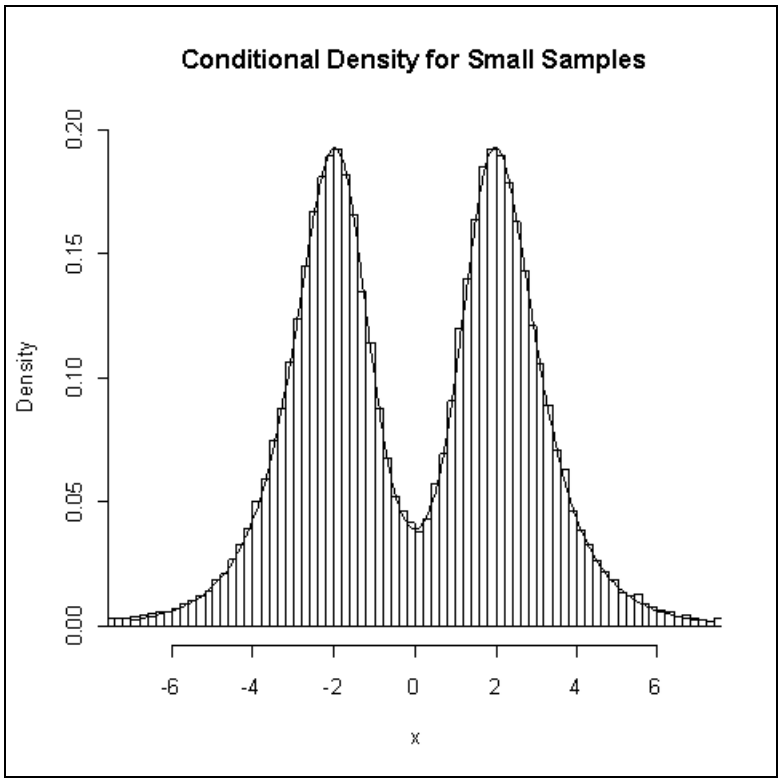

Figure 4.6: Conditional density for small samples.
Histogram of the test statistics for small samples. The solid curve indicates the density given in (**4.17**).

It will be discussed in section 6.4 that although the conditional density of the test statistics behaves as expected, the method proposed for controlling the variance of the number of discoveries may not be applicable for small samples.

Chapter 5 - Combined Correlation Coefficient of Test Statistics

5.1 Sampling Distribution of the Combined Correlation Coefficient

The correlations between the i[th] and the j[th] features will be estimated by the correlations of the data. Assume that r_x and r_y denote the sample estimates of ρ_x and ρ_y. Again, the double subscript notation for correlations is suppressed but it is noted that the correlations can be different depending on which pairs are under consideration. Let r_z denote the estimate of ρ_z. Assume that there are n_1 individuals in group 1 and n_2 individuals in group 2. The sampling distributions of r_x and r_y under the assumption of bivariate normality of the data for pairs of features, when the true correlations are equal to zero are given by (c.f. Stuart and Ord, 2009)

$$f(r_x) = \frac{1}{B(\frac{1}{2}, \frac{1}{2}(n_1 - 2))}\left(1 - r_x^2\right)^{\frac{1}{2}(n_1 - 4)} \tag{5.1}$$

and

$$f(r_y) = \frac{1}{B(\frac{1}{2}, \frac{1}{2}(n_2 - 2))}\left(1 - r_y^2\right)^{\frac{1}{2}(n_2 - 4)} \tag{5.2}$$

$$-1 \le r_x, r_y \le 1$$

The value of r_z will be calculated using (4.4) with sample statistics for each unique pair of features and its distribution will be used to identify the deviation of the dependence structure from independence (i.e., the two features are independent in both groups). Assume that σ_i^* and σ_j^* denote the standard deviation of the data for the i[th] and j[th] features in group 1 and τ_i^* and τ_j^* denote the variance of the data for i[th] and j[th] features in group 2. Then (4.4) can be written as

$$\rho_z = \frac{\rho_x \frac{\sigma_i^*}{\sqrt{n_1}}\frac{\sigma_j^*}{\sqrt{n_1}} + \rho_y \frac{\tau_i^*}{\sqrt{n_2}}\frac{\tau_j^*}{\sqrt{n_2}}}{\sqrt{\left(\frac{\sigma_i^{*2}}{n_1} + \frac{\tau_i^{*2}}{n_2}\right)\left(\frac{\sigma_j^{*2}}{n_1} + \frac{\tau_j^{*2}}{n_2}\right)}} = \frac{\rho_x \frac{\sigma_i^*\sigma_j^*}{n_1} + \rho_y \frac{\tau_i^*\tau_j^*}{n_2}}{\sqrt{\left(\frac{\sigma_i^{*2}}{n_1} + \frac{\tau_i^{*2}}{n_2}\right)\left(\frac{\sigma_j^{*2}}{n_1} + \frac{\tau_j^{*2}}{n_2}\right)}} \tag{5.3}$$

Let $a = \dfrac{\dfrac{\sigma_i^* \sigma_j^*}{n_1}}{\sqrt{\left(\dfrac{\sigma_i^{*2}}{n_1} + \dfrac{\tau_i^{*2}}{n_2}\right)\left(\dfrac{\sigma_j^{*2}}{n_1} + \dfrac{\tau_j^{*2}}{n_2}\right)}}$ and $b = \dfrac{\dfrac{\tau_i^* \tau_j^*}{n_2}}{\sqrt{\left(\dfrac{\sigma_i^{*2}}{n_1} + \dfrac{\tau_i^{*2}}{n_2}\right)\left(\dfrac{\sigma_j^{*2}}{n_1} + \dfrac{\tau_j^{*2}}{n_2}\right)}}$

Then (5.3) can be written as

$$\rho_z = a\rho_x + b\rho_y \tag{5.4}$$

It can be assumed that r_x and r_y are independent since they are computed from data on independent samples. Based on sample correlations r_x and r_y within the two groups, define an estimate of ρ_z as,

$$r_z = ar_x + br_y \tag{5.5}$$

The distribution of r_z is a linear transformation of r_x and r_y.

The joint density of r_x and r_y, when $\rho_x = \rho_y = 0$, can be written as

$$f(r_x r_y) = \frac{1}{\mathrm{B}(\frac{1}{2},\frac{1}{2}(n_1-2))}\frac{1}{\mathrm{B}(\frac{1}{2},\frac{1}{2}(n_2-2))}\left(1-r_x^2\right)^{\frac{1}{2}(n_1-4)}\left(1-r_y^2\right)^{\frac{1}{2}(n_2-4)} \tag{5.6}$$

$$-1 \le r_x, r_y \le 1$$

Let $x = ar_x + br_y$ and $y = ar_x - br_y$

By performing the transformation above

$$f_{XY}(x,y;a,b) = \frac{1}{2ab\mathrm{B}(\frac{1}{2},\frac{1}{2}(n_1-2))\mathrm{B}(\frac{1}{2},\frac{1}{2}(n_2-2))}\left(1-\left[\frac{x+y}{2a}\right]^2\right)^{\frac{1}{2}(n_1-4)}\left(1-\left[\frac{x-y}{2b}\right]^2\right)^{\frac{1}{2}(n_2-4)} \tag{5.7}$$

$$-(a+b) \le x \le (a+b)$$

$$y \in \begin{cases} [-2a-x, 2b+x] & for[-1 \le x < a-b] \\ [-2a-x, 2a-x] & for[a-b \le x < b-a] \\ [-2b+x, 2a-x] & for[b-a \le x < 1] \end{cases}$$

The joint density given in (5.7) is derived under the assumption that features are independent ($\rho_x = 0$, $\rho_y = 0$) and it only depends on the sample sizes and the individual variances of the features within the two groups. Figure 5.1 below illustrates this joint density in (5.7) for two cases of different combinations of sample sizes and variances for two arbitrary features 1 and 2.

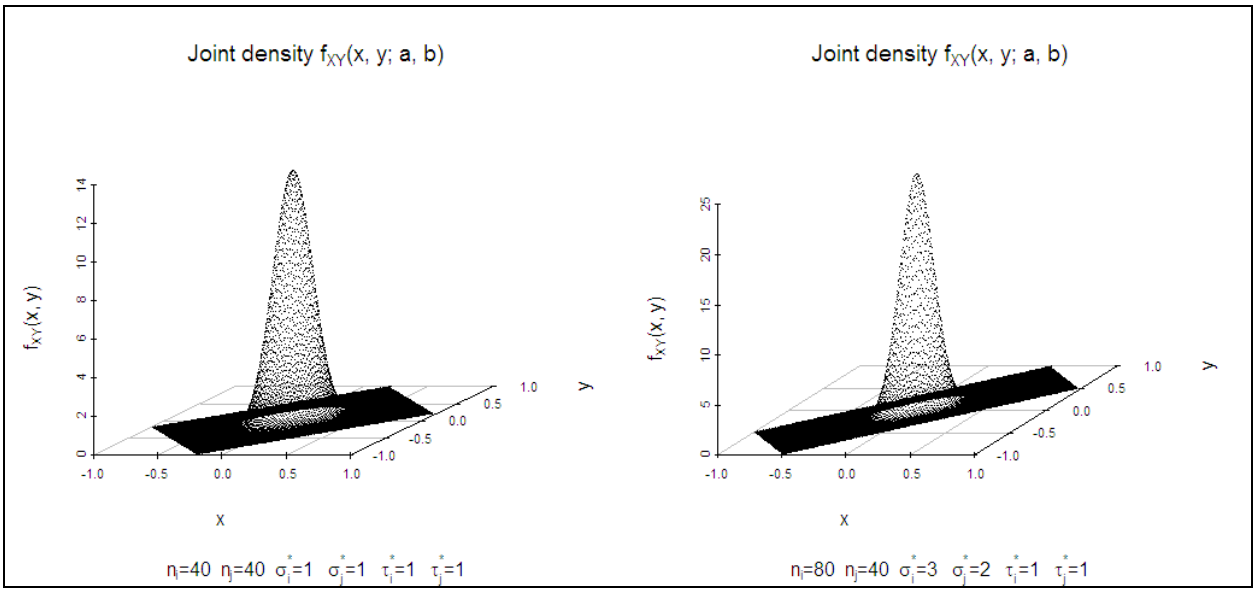

Figure 5.1: The joint density *f*(*x*, *y*).
The joint density of *f*(*x*, *y*) given in (5.7**)** for different combinations of sample sizes and standard deviations. The plots also show the changes in the domains of *x* and *y*.

In order to obtain the density of the estimator r_z, the variable y must be integrated out from (5.7). As shown above the domain for y must be selected with caution since it not only depends on the value of x but also the constants a and b in (5.4). Noting $r_z = ar_x + br_y = x$ in the parameterization above, the density of r_z can be written as

$$f(x) = \int_{lower\ bound(x,a,b)}^{upper\ bound(x,a,b)} f_{XY}(x, y)\, dy$$

$$= \frac{1}{2ab\mathrm{B}(\frac{1}{2}, \frac{1}{2}(n_1 - 2))\mathrm{B}(\frac{1}{2}, \frac{1}{2}(n_2 - 2))} \int_{lower\ bound}^{upper\ bound} \left(1 - \left[\frac{x+y}{2a}\right]^2\right)^{\frac{1}{2}(n_1-4)} \left(1 - \left[\frac{x-y}{2b}\right]^2\right)^{\frac{1}{2}(n_2-4)} dy \quad (5.8)$$

where the variable x in (5.8) corresponds to the random variable r_z. The above integration cannot be analytically solved. However, numerical methods can be used to obtain the density curve of $f(r_z)$. The validity of this density estimation can be verified by a simulation.

Simulation 4

For both cases shown in Figure 5.1, data were generated for 10,000 independent pairs of features in two groups using the bivariate normal distribution. The combined correlation for each pair is calculated. Histograms in Figure 5.2 show the empirical densities of combined correlations and the solid curves are for the theoretical densities of the combined correlations obtained by numerically evaluating (5.8). These show the agreement between the theoretical and empirical distributions of the combined correlation between the features.

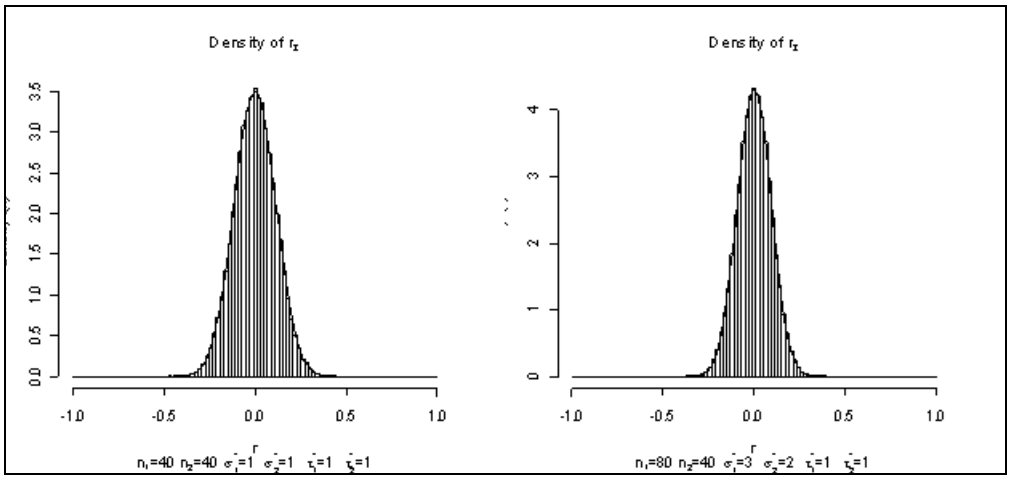

Figure 5.2: Empirical and theoretical densities of the combined correlation.
The solid lines in the plots show that the theoretical density given in (5.8**)** agrees with the densities in the histograms.

This can be further verified by the comparison of theoretical and empirical quantiles. Figure 5.3 below is the quantile-quantile plot for the above two cases. The plot shows every 5^{th} percentile from the 5^{th} percentile to 95^{th} percentile. They verify the agreement between the empirical density of the correlations and the density given in (5.8).

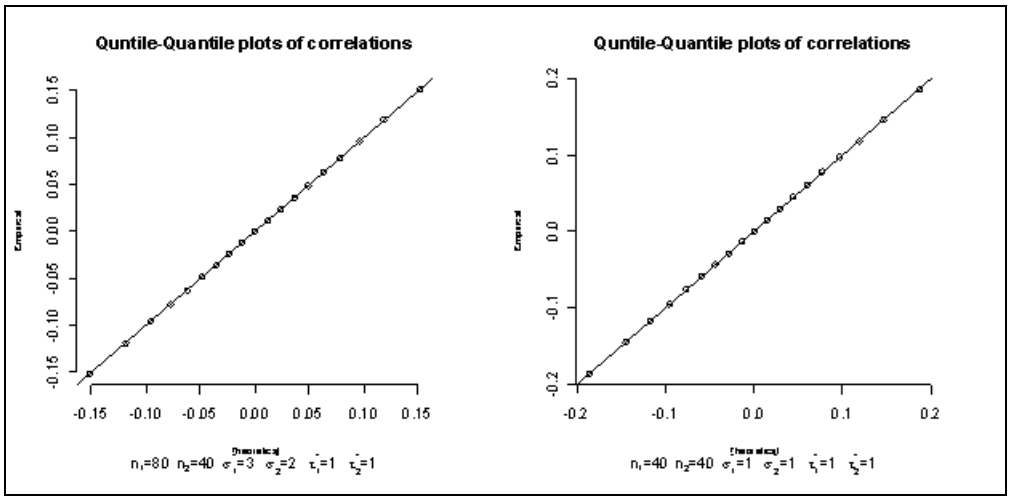

Figure 5.3: Quantile-quantile plots of combined correlation.
Quantile-quantile plots of the simulated data shown in Figure **5.2**. This illustrates the agreement between the densities of correlations of independent data and the theoretical density given in (5.8**)**. Every 5th percentile from 5th percentile to 95th percentile are shown in the plot.

Since the integration in (5.8) done numerically, its accuracy is evaluated against the subdivisions used in the *y* space. The settings in the first plot of Figure 5.2 will be used for this evaluation. Results for several subdivision sizes are listed in the table below. These sizes are used to divide the x and y ranges of the joint density in the integral. The table shows that the values of the density estimates change very little when subdivisions of size 0.005 or smaller are used. A size of 0.0005 will be used for the subdivisions in the following applications of correlation density estimates.

Table 5.1: Density estimation accuracy for *f*(*r*) using (5.8)

Subdivision size	Density estimation point				
	-0.8	-0.3	0	0.3	0.8
0.01	1.389532e-16	0.1023440	3.487013	0.1023440	1.389532e-16
0.005	1.119657e-16	0.1018563	3.489469	0.1018563	1.119657e-16
0.001	1.037186e-16	0.1017003	3.490255	0.1017003	1.037186e-16
0.0005	1.034639e-16	0.1016955	3.490279	0.1016955	1.034639e-16

The density in (5.8) can be used to check the degree of dependence among the features. If the data are independent, the density of the combined correlations should approximate this density. However, the values *a* and *b* may not be constants across all features since features have different variances. In order to make a valid comparison, the data will be scaled using the sample standard deviations so the variances of the features are equal to 1. This causes the constants *a*

and b to be equal for all the pairs of features. Then the distribution of the combined correlation for the data can be compared with the theoretical distribution of the combined correlation with standard deviations equaling to 1. The mean structure of the data will not be adjusted (centered) in the standardization since it would not have an effect in calculating the combined correlation in (4.4). Then the constants a and b become

$$a = \frac{\frac{1}{n_1}}{\left(\frac{1}{n_1}+\frac{1}{n_2}\right)} \text{ and } b = \frac{\frac{1}{n_2}}{\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}$$

The rest of the calculation and the integration in (5.8) do not change due to scaling the data. The scaled data will be used to obtain the density of the combined correlations in the simulations and applications in this dissertation.

5.2 Combined Correlation Coefficient in Real World Data

Section 5.1 illustrated the behavior of the combined pairwise correlation when all of the features being tested are independent in both groups. However, this is not the case in real life datasets and there are various dependencies existing in data that are both identified and unidentified by researchers. The distributions of the combined correlation coefficient for the two applications in this report, the lung cancer data and multiple myeloma data, were used to compare different correlation distributions. Figure 5.4 below shows the density curves of the combined correlations of the two datasets.

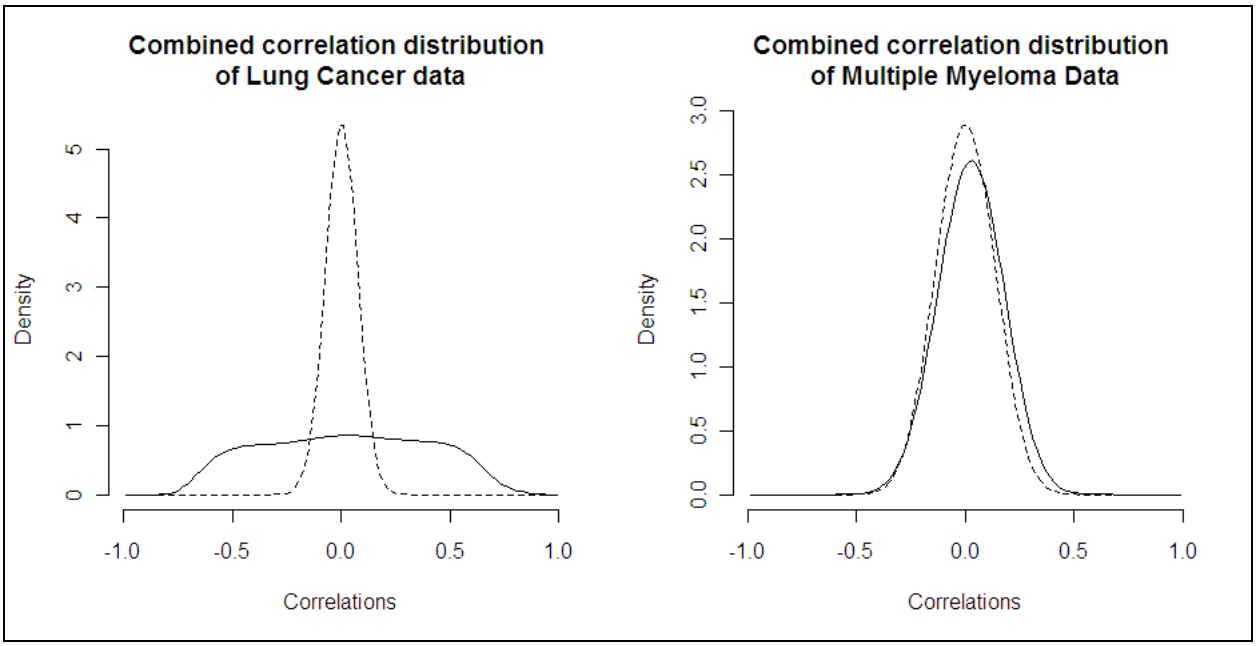

Figure 5.4: Correlation densities for Lung Cancer and Multiple Myeloma datasets.
The dotted lines indicate the density under independence for corresponding dimensions calculated using (5.8**)**. The solid lines show the empirical correlation densities for the two datasets.

The solid curves indicate the empirical density of combined correlations obtained through data. The dotted curves in the plots indicate the density of the combined correlation under independence for corresponding dimensions and sample sizes. While the departure of the correlation distribution in multiple myeloma data from independence is subtle, there is a drastic difference between the correlation distribution in lung cancer data and the corresponding distribution under independence. These plots illustrate the diversity of the correlation structures existing in real data. These curves can also be used to measure the degree of dependencies in the data by comparing the empirical curves with the theoretical distributions under independence.

The quantiles of these correlation densities can be compared with the quantiles of the correlation density under independence. Figure 5.5 below lists these two densities in the same plot and it shows the degree of deviation from independence in the two datasets being considered. The density plots in Figure 5.4 are used to make inferences about the correlation density of the data and to quantify the deviation of the correlation structure from independence.

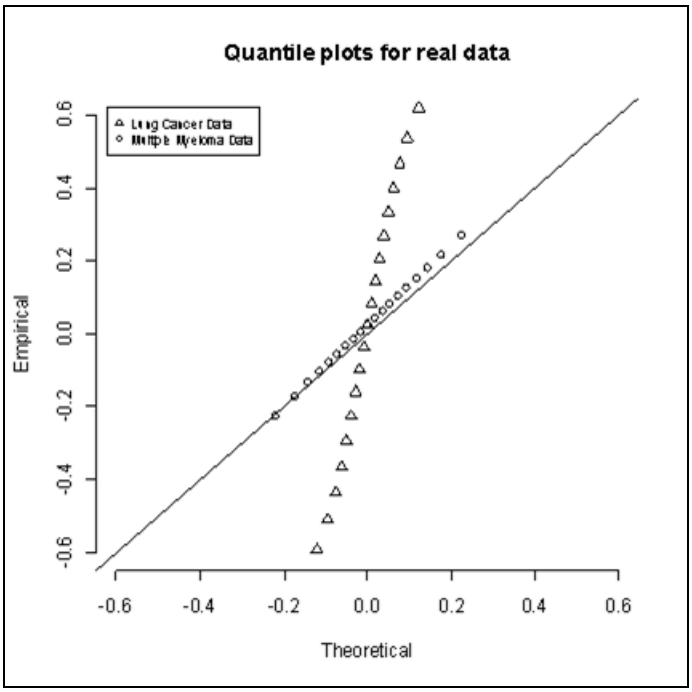

Figure 5.5: Quantile-quantile plots for correlations of the two applications.
The figure illustrates the drastic deviation of correlations from independence in Lung Cancer data (symbol: triangle) compared to the Multiple Myeloma Data (symbol: circle). Plot shows every .05 quantile from. .05 quantile to .95 quantile.

The conditional density given in 4.11 depends on the combined correlation between the test statistics, and this chapter explored some of the characteristics of the combined correlation and derived the density of the combined correlation under independence. This density can be compared with the empirical density of combined correlation of a dataset to determine the degree of departure from independence as shown in Figure 5.4. This will be used in the suggested process of conditionally testing features introduced in the next chapter.

Chapter 6 - Conditional Network Testing

Previous chapters illustrated the effect of dependencies in large scale two group comparison studies. This information can be combined to devise a method that controls the variability of the number of discoveries made in these analyses. This chapter introduces an adjustment to the regular two group comparisons, considering the conditional density of test statistics introduced in Chapter 4 - and the combined correlation of test statistics introduced in Chapter 5 - , to reduce the variability of the number of discoveries under dependencies.

6.1 Introduction to Conditional Network Testing

Chapter 3 - and Chapter 4 - illustrated the increment of variance of the number of discoveries due to the correlations of the data. Chapter 4 - illustrated the behavior of the density of a test statistic conditioned on the rejection or non-rejection status of another correlated test statistic. It was shown in Figure 4.2 that the tail areas of the conditional density are increased under dependencies. Chapter 5 - illustrated the behavior of the distribution of the correlations between test statistics. The empirical density of the correlations obtained from the data can be compared with the theoretical density of correlations under independence given by (5.8). These comparisons can be used to investigate the degree of dependencies present in the data and they can be used to build networks of features that are governed by the correlations of the data. Once a set of networks is constructed, the features can be tested within the networks using the conditional density. The objective of testing within networks is to further investigate the features that have test statistic values close to the cutoff level for a given size. The decisions for features with very large and very small test statistics values are not expected to be changed, however the decisions for features with borderline test statistics will be affected by network testing since the p-values are drawn from conditional densities. The method is referred to as Conditional Network Testing (CNT). The intuition and the procedure for network testing are described in the following sections.

6.2 Building Networks of Features

As discussed in the literature review section, the features of interest are often related to each other rather than being independent of each other. In genetics this can be referred to as co-regulated genes or genes in pathways. In lipidomics, this can be referred to as reactant and

product relationships between compounds. These concepts govern the correlation densities shown in Figure 5.4 that can be empirically estimated from observed data. The features can be networked together based on the correlations of the data. It was shown in (4.12) that the variance of the number of discoveries depends on the correlation of the test statistics. Therefore the combined correlations of test statistics can be used to determine the dependencies of the tests. The correlations referred to in this chapter are these combined correlations between test statistics.

The density curve given in (5.8) defines the correlation density under independence. Therefore, the empirical correlation density can be compared with this correlation density to determine the departure of the dependence structure from one of complete independence. Data for each feature are scaled within each group to ensure the comparison between the theoretical and empirical densities are valid as discussed at the end of section 5.1. The following paragraphs discuss some of the issues that need to be considered in building networks of features that are conditionally tested.

The intention of testing features within networks is to use the conditional density to determine an alternate p-value for testing differences between the two groups. The goal is to obtain a modified p-value for features with smaller test statistics conditioned on the features with a larger test statistic. A test statistic for each feature is obtained by conducting an initial independent t-test on each feature. Figure 4.4 illustrates that the larger (magnitude) correlations results in heavier tail areas in the conditional density. Assume that the interest is in attaching a feature, say feature A, to another feature that has a larger test statistic. Among the features with test statistics larger than that for feature A, the feature with the largest correlation with A is selected so that the effect of the tail area is the maximum compared to other test statistics.

In addition, sampling variability of correlations under independence must also be considered for network building. For example, Figure 5.4 illustrates that there are more pairs of features with correlations equal to 0.4 than expected under independence in the multiple myeloma data. Therefore, if a feature has a correlation of 0.4 with another feature, they can be attached to each other (attach the feature with the smaller test statistic to the one with the larger test statistic). On the other hand, if the correlation between two features in the same dataset has a correlation of -0.1, the same figure shows that there are less pairs with correlation -0.1 than

expected under independence. Therefore, features with -0.1 correlation will not be attached. If these data were independent, there would not be any difference between the empirical distribution of correlations and the theoretical distribution of correlations under independence. Then in this ideal situation, features will not be attached to each other and each feature would stand alone and is considered its own network. However, the sampling variability of the empirical density of the correlations allows some networks even under independence.

Assume that the interest is in testing two correlated features A and B, and the t-test statistic for feature A is larger than the test statistic value for B. Suppose that the correlation between A and B indicates that there is a relationship between A and B. Then the above description suggests that B should be attached to A in a network. However, an initial decision must be considered before constructing the network. Figure 4.4 shows that if A is already rejected, the tail area of the conditional density B is inflated. If the initial test for feature B failed to reject the null hypothesis (larger p-value), considering the conditional density does not make a difference in the decision. (The p-value drawn from the conditional density is larger than the x p-value.). This suggests that the features for which the null hypothesis failed to reject need not be mixed with features for which the null hypothesis was rejected in network building. On the other hand, consider network building among features that failed to reject the null hypothesis. Figure 4.5 illustrates that tail area of the conditional density, when conditioned on a features that failed to reject, shrinks. The intention would be to obtain a p-value for larger features (which failed to reject the null hypothesis) conditioned on features with smaller test statistics. Therefore, the network building within the features that failed to reject the null hypothesis starts with smaller test statistics and attaches smaller test statistics to the larger ones. The p-values are computed from the conditional density given in (4.14). The following summarizes the network building algorithm.

The algorithm for building networks is given below.

1. Scale the data within groups by dividing each feature by its standard deviation.
2. Perform a t-test for each feature and transform the test statistics using (2.10**)**
3. Compute all $K(K-1)/2$ combined correlations for the data using (4.4).
4. Among the features that were declared significant
    a. Start with the largest (absolute value) test statistic

b. Attach smaller (absolute value) test statistics to the larger test statistics by comparing correlation densities as described above
c. If no attachments can be made, select the largest remaining test statistic to build a new network
d. If no features can be attached to a selected feature, it is declared as its own network

5. Steps in 4 are repeated until no more features are left for networking
6. Among the features that were declared not significant
    a. Start with the smallest (absolute value) test statistic
    b. Attach larger (absolute value) test statistics to the smaller test statistics by comparing correlation densities as described above
    c. If no attachments can be made, select the smallest remaining test statistic to build a new network
    d. If no features can be attached to a selected feature, it is declared its own network
7. Steps in 6 repeated until no more features are left for networking.

Above algorithm was implemented in *R*. Additional measures are implemented to stabilize the process and program control methods were introduced. The program was developed to output the network structure to be visualized using the software *Pajek* (*http://pajek.imfm.si/doku.php?id=pajek*). The algorithm was applied to lung cancer and multiple myeloma datasets (testing was done at $\alpha = 0.05$ level). The summaries of networking for these two datasets are given in the table below.

Table 6.1: Summary of network Structure for two example datasets

Dataset	Grouping	Number of networks	Network Sizes	Summary
Lung Cancer	Networks For significant probes	1	3169	3169 initial discoveries were reduced to 1675
	Networks for non-significant probes	1	9517	Added 9 new discoveries
Multiple Myeloma	Networks for significant probes	4	251 225 6 32	514 initial discoveries were reduced to 298
	Networks for non-significant probes	3	2450 768 238	Added 11 new discoveries

Figure 6.1 below illustrates network number 4 for the multiple myeloma data which has 32 features that are designated as 32 labeled Affymetrix probes. The sizes of the dots represent the relative size of the test statistics. The head of the arrow indicates the probe being adjusted. Note that smaller (i.e., smaller test statistic value) probes are attached to larger (larger test statistic value) probes. Further explanation of how to read the graph will be given later on a simpler network.

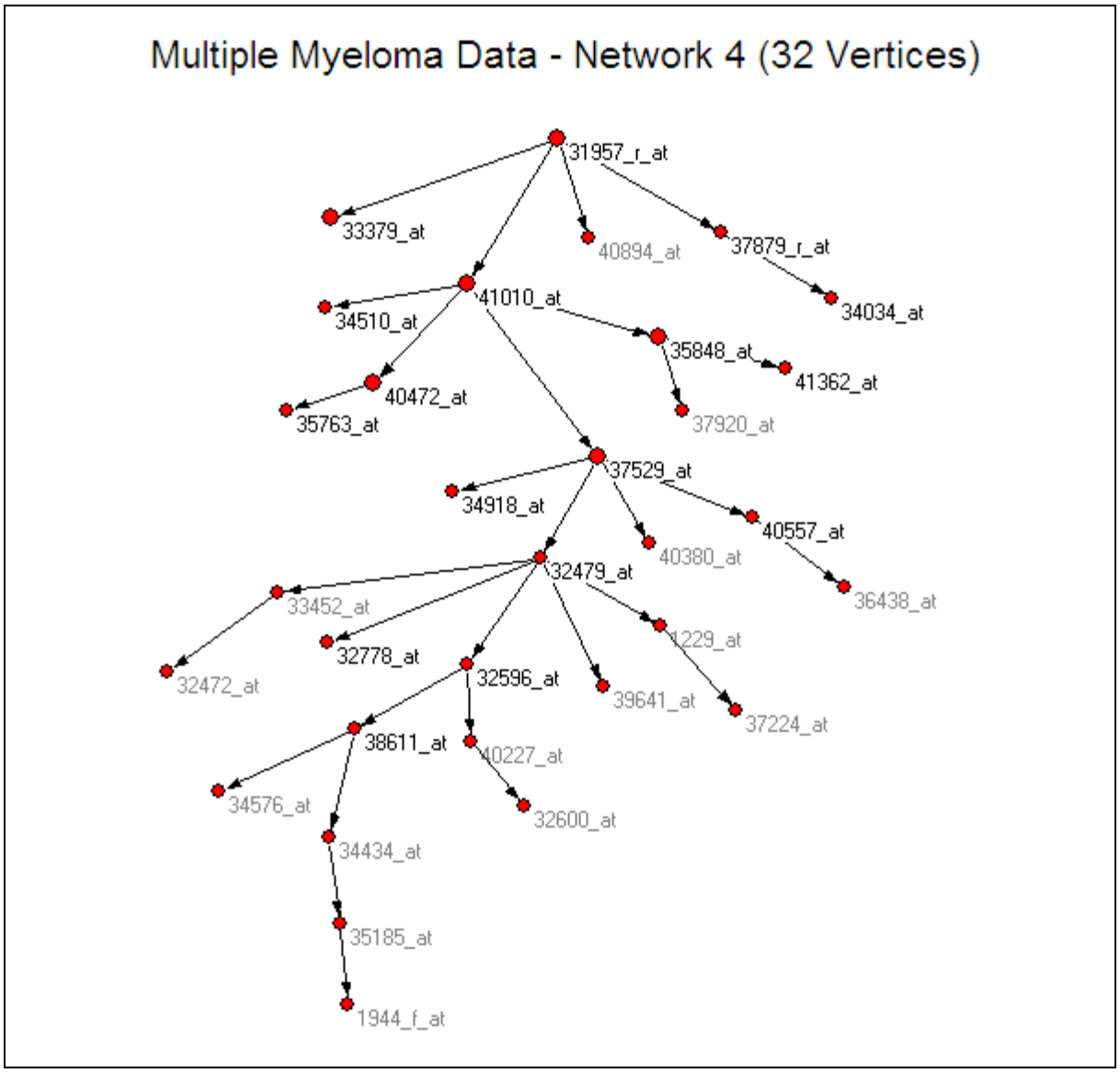

Figure 6.1: Network 4 for multiple myeloma dataset.
Network 4 obtained by applying networking algorithm on the set of probes declared significant in multiple myeloma dataset. The network included 32 probes. The size of the dots represents the relative size of the test statistic for each probe. The labels shown in the figure are Affymetrix probe IDs.

These networks are tested using the conditional densities derived in Chapter 4 - . The process of testing is discussed in the next section.

6.3 Testing Networks of Features

The densities given in (4.11) and (4.14) are used to obtain p-values for the features that are conditionally tested. Assume that a dataset resulted in the following network given in Figure 6.2 for the set of features that were declared as significant in the initial t-tests. The procedure obtains

a new p-value for each feature conditioned on the status of the most immediate feature to which it is attached.

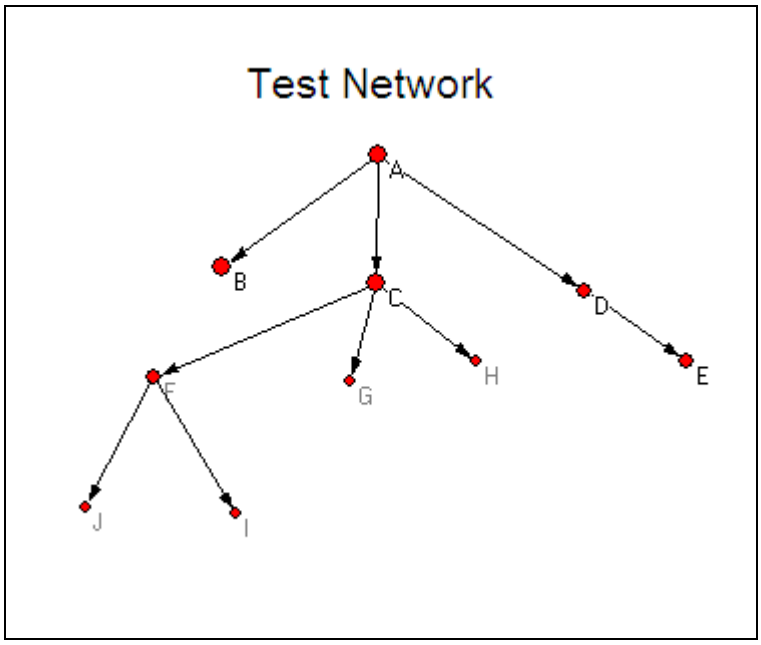

Figure 6.2: Example network of features.
Illustration of a network constructed by the proposed method.

The p-values for features B, C and D are obtained by drawing a p-value for their test statistics using the conditional density (4.11), conditioned on feature A. Similarly, features F, G and H are tested conditioned on the outcome of the test for feature C. The combined correlation between features governs the new p-value drawn from the conditional density. Note that the decision for features A, the feature with the largest test statistic, is kept unchanged. The procedure attempts to reconsider the decisions for some of the test statistics which are closer to the cutoff value for statistical significance.

The networks for features declared not significant are tested in the same manner but in reverse order. The larger test statistics are attached to smaller test statistics and tested conditioned on the smaller test statistics. The goal again is to revise the decision (rejection/non-rejection) for features which have borderline test statistics values.

6.4 Initial Results

The initial goal was to produce a method that stabilizes the variance of the number of discoveries under dependencies. In order to illustrate the method proposed in this chapter, simulation 2 from page 34 is repeated (one instance of the simulation, not 100 repetitions) to evaluate the proposed method.

Simulation 5

Data for each dependence structure given in Table 4.1 are simulated using multivariate normal distribution 200 times under true global null hypothesis (i.e., there is no mean difference for any feature across the two groups). Data for 500 features are generated and tested using both regular t-tests and the proposed network testing method at a significance level of 0.05. Any discovery made is a type I error since the global null hypothesis is true. The numbers of discoveries from each simulation were used for calculating the variance of the number of discoveries. Figure 6.3 below illustrates the reduction in variance obtained by the proposed method. If all features were independent, one would expect the variance of the number of discoveries to be equal to 500*0.05*0.95 = 23.75. The expected number of discoveries, i.e., Type I errors, is 25.

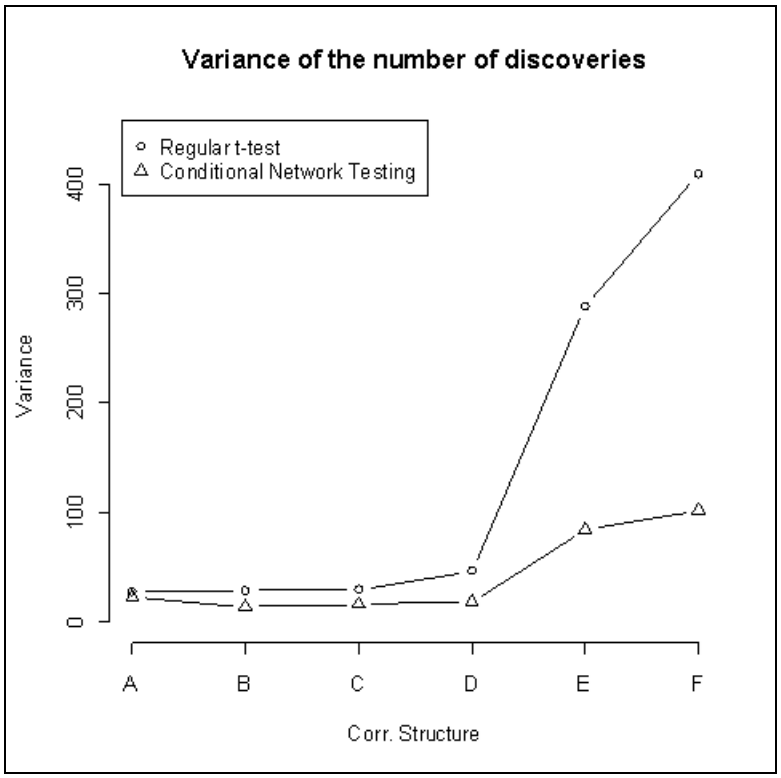

Figure 6.3: Variance of the number of discoveries.
The line with a circle symbol indicates the variance of the number of discoveries when a series of t-tests is used. The line with the triangle symbol indicates the variance of the discoveries made by the Conditional Network Testing. A – F are dependence structures listed in Table **4.1**. The graph illustrates the reduction in variance after using conditional network testing

The figure above suggests that the network testing method is capable of reducing the variance of the number of discoveries (Type I errors under this scenario) under dependencies. While it is not reduced to the level under independence, the variance is drastically reduced at the highest level of dependence. The method was applied to the two real datasets considered in this dissertation. The initial t-test declared 3170 probes as significant at 0.05 level. The first stage (network testing with features declared statistically significant by the initial test) of the CNT reduced this to 1675 probes and the second stage (network testing within features declared non-significant by the initial test) declared 9 probes as statistically significant making the total number of discoveries 1685. Similarly, the first stage in testing multiple myeloma data reduced the number of discoveries from 514 to 298 and the second level declared 11 probes that were initially declared as non-significant as significant making a total of 309 discoveries.

This chapter put forward a method to control the variance of the number of discoveries made in large scale simultaneous mean testing of two groups. A simple simulation was used to illustrate the proposed methods ability to control this variability. However, the dependence structures used in this simulation do no accurately depict the dependence structures in real world data. A method for simulating data with dependencies closer to that of real world data is proposed in the next chapter. This method is used to evaluate CNT in Chpater 8 and compare error control methods used on the p-values obtained from CNT with the error control methods used on t-tests.

The small sample results shown in section 4.3 illustrates that the conditional density derived through the t-test holds for small samples. However, the computation of combined correlations for small samples do not result in a smooth estimate of the correlation density between test statistics. Since CNT relies on the empirical density of correlations for constructing networks, the method will not be used on small samples.

Chapter 7 - Plasmode Data with preserved Combined Correlation Structure

7.1 Overview of Plasmode Data Methods

The properties and performance characteristics of statistical methods for high-dimensional data are commonly evaluated with simulations. Data that are simulated with unrealistic or overly simplistic structures may not adequately test the characteristics of a statistical method. Chapters 4 and 5 demonstrated that the two group comparisons in high dimensional data must be done with care due to the instability in the number of discoveries made under dependencies. The realistic structure of datasets must be reflected in simulated data that are used for evaluating statistical methods. While the assumption of independence is not valid in general, simulating data with systematic dependence structures such as block diagonal covariance matrices may also be inadequate to represent realistic dependencies. This section suggests a method to simulate plasmode data. Plasmode is a term coined several years ago to describe data sets that are derived from real data but for which some truth is known. Mehta et al. (2004) more concisely refer to a plasmode as "a real data set whose true structure is known." The plasmodes can accommodate unknown correlation structures among genes, unknown distributions of effects among differentially expressed genes, an unknown null distribution of gene expression data, and other aspects that are difficult to model using theoretical distributions. In addition, preservation of the dependence structure of the data is expected up to a certain extent.

The existing plasmode methods do not allow different dependence structures in the two groups being compared (cf., Gadbury et al., 2008). In addition, these methods assume equal sample sizes in the two groups. Since the density of the correlations depend on the sample size according to (5.8), the assumption of equal sample sizes in the two groups may not be appropriate to represent real datasets. This is illustrated for lung cancer data and multiple myeloma datasets in Figure 7.1. The plot shows differences in the empirical correlation distributions between the two treatment groups and their differences with the distribution of the combined correlation statistic.

A comparison of Figure 5.4 and Figure 7.1 can be used to determine the departure of these dependence structures from one of complete independence among features. The differences of correlation densities between groups suggest that assumption of equal correlation structures in the two groups may not be reasonable. In addition, using only one of the groups as the base group for which to construct plasmode data (i.e., as in Gadbury et al, 2008) may result in datasets that are not representative of the original data. A new method is suggested here that can be used for simulating more realistic data that captures the characteristics of the original data to a higher extent. One might say that he proposed method generalizes what has been done earlier, and it allows more of the original structure in a dataset to be preserved in simulated data.

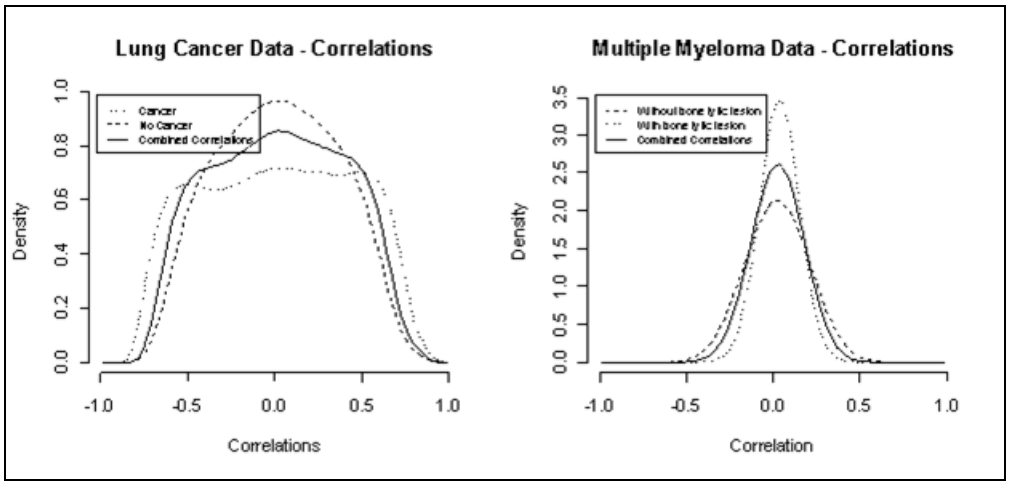

Figure 7.1: Comparison of correlation distributions.
The first plot shows different correlations for the lung cancer data. The dashed curve is the correlation within the non-cancer groups and the dotted curve is for the cancer group. The solid line indicates the distribution of the combined correlation given in (4.4**)**. The dashed and dotted lines in the second plot represent correlations in the groups without bone lytic lesions and with bone lytic lesions, respectively. The solid line represents the combined correlation of the bone lytic data.

7.2 New Plasmode Method

As discussed above, the method proposed in this section attempts to simulate datasets with some known truth about the data. One objective is that the combined correlation of the data being simulated represents the correlation structure of the original dataset which is being used as a template for the simulated data. The algorithm to generate plasmode data is described below. Assume that there are n_C samples in the control group and n_T samples in the treatment group

(Treatment vs. Control terminology is used for the convenience of reference while the test may be between any types of groups), and a total of K features are being tested.

Plasmode Data Algorithm

1. Compute the t-test statistics for all features.
   Denote these t-test statistic values by T_i ($i = 1, \ldots, K$) and store these values.
2. Compute the mean m_{Ci} and m_{Ti} for all features within control and treatment groups separately.
3. Compute and store the standard deviation for each feature within groups.
   This step results in s_{Ci} and s_{Ti} for the ith feature.
4. Center data using the means in step 2 within each group.
   Substract the mean of each feature from the observed values. Perform this step separately for the two groups being compared. Note that the null hypothesis for no mean difference between treatment groups is now exactly true for every feature.
5. Obtain bootstrap samples of the treatment and control groups separately.
   Select n_C sized samples from the control group with replacement and n_T sized samples from the treatment group with replacement to construct the new plasmode treatment and control groups. This dataset is referred to as the plasemode null data set and
   sampling with replacement introduces some sampling variability. Since the data are centered in step 4, there are no statistically different features in the data except for what occurs from sampling variability.
6. Compute the probabilities $|T_i| \Big/ \sum_{j=1}^{K} |T_j|$ where T_i and T_j are defined in step 1.
7. Select a predetermined proportion, p, of features from the list of features, with corresponding probabilities computed in step 6.
   These features will be simulated with a difference between the two groups. The difference is made according to the differences seen in the original sample and the process is explained in the next step. The intention is to assess the ability of statistical methods to correctly identify these features.
8. Restore the mean structure for the p proportion of features (selected in step 7) in the treatment group of the plasmode dataset.

If the i[th] feature is selected in step 7, then add $T_i\sqrt{\frac{s_{Ci}^2}{n_C}+\frac{s_{Ti}^2}{n_T}}$ to the sample values for the i[th] feature in the plasmode treatment group. This reinstates the mean difference that existed in the i[th] feature between the treatment and control groups. However, since the subjects of the plasmode data are selected randomly in step 5, some randomness is added to the overall differences seen in each data set.

The global null hypothesis is exactly true for the resulting dataset in step 4. The plasmode samples obtained in step 5 can be thought of as samples drawn from a population for which the global null hypothesis is true. The employment of the bootstrap introduces the random variability to the samples. Since the resampling in step 5 is done within groups, it can be assumed that the dependence structure of the data is preserved. The knowledge of the mean structure allows the evaluation and comparison of error control methods and other statistical methods. The simulation below illustrates some properties of the proposed plasmode method.

Simulation 6

This simulation illustrates the proposed plasmode method's ability to generate the null distribution of p-values under complete independence among features. Data were randomly generated for 500 independent features in two groups (40 subjects in each group) using the multivariate normal distribution separately for groups where are marginal distributions are standard normal and all features are independent. The p-values for testing for a mean difference for all 500 features were computed using a t-test and recorded. This process is repeated 200 times to obtain 200 sets of 500 p-values. This process is referred to below as 'repeated samples,' indicating that the high-dimensional data sets were generated separately from a hypothetical statistical model (in this case, a standard normal distribution). Histograms with 100 bins are drawn to these p-values and their density heights were averaged to obtain an average density of p-values. Similarly, the standard deviation is calculated for the bin counts to obtain the standard deviation of the distribution of p-values from 500 tests. The first plot of Figure 7.2 illustrates the behavior of this mean and standard deviation. The solid line indicates the mean p-values and the dashed lines indicate the one standard deviation limits around the means. One of these datasets was used as a template to generate plasmode data with a true null distribution. Steps 1 thru 5 of

the plasmode procedure were repeatedly used (i.e., under the true global null hypothesis) to generate 200 plasmode datasets from one simulated dataset, and the mean and the standard deviations of the p-value distribution were obtained the same way as above. The solid line in the second plot shows the mean of the p-values obtained by the plasmode method and the dashed lines indicate one standard deviation limits. Since the global null hypothesis is true in all simulated cases, the distribution of p-values from 500 tests is expected to be uniform on the interval 0 to 1. The figure shows that the p-value distribution of plasmode data on average represents the p-value distribution of repeated samples. The standard deviation at smaller p-values is slightly higher in plasmode data compared to repeated samples. The distribution from plasmode data showing increased variability near zero suggests that a higher variance in the number of discoveries from plasmode data can be expected. The reason for this effect will be discussed in more detail later in this section.

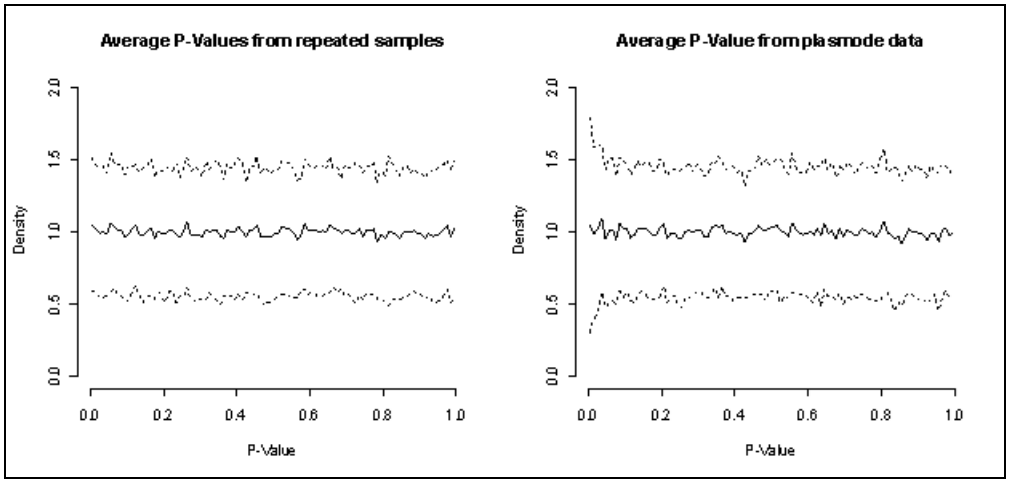

Figure 7.2: Comparison of null p-value distributions.
Null p-value distributions from repeated samples and from the plasmode algorithm. On average, the p-value distribution from plasmode data resembles the p-value distribution of the repeated samples. The dashed lines indicate the 1 standard deviation limits of p-values from repeated samples in the first plot and repeated plasmodes in the second plot. Variance in the plasmode data is slightly higher at smaller p-values than in the repeated samples.

In order to illustrate the behavior of the number of discoveries and its variance, Figure 6.3 is recreated for plasmode data. A sample from each correlation structure in Table 4.1 is used as a template for plasmode data. Only steps 1 – 5 were used so that global null hypothesis is true. Figure 7.3 below shows the variance of the number of discoveries by using the t-test on both repeated samples and plasmode data. Since the global null hypothesis is true, all discoveries are

type I errors. This shows that there is a higher variability in the number of discoveries using plasmode data. This increment can also be seen in Figure 7.2 where there is an increased variability near smaller p-values. However, this variability is reduced by a large amount when CNT is used. CNT obtains a variability that is closer to the application of CNT in repeated samples.

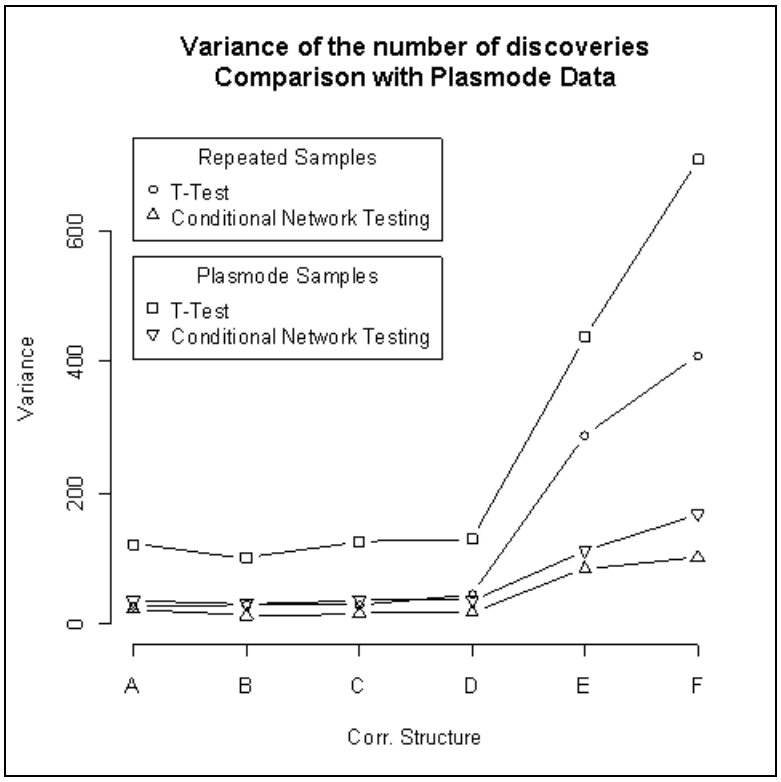

Figure 7.3: Comparison of the variance of the number of discoveries
Comparison of the variance of the number of discoveries under the global null hypothesis using repeated samples and plasmode data. The plot shows that the variance is higher for plasmode data than from using the t-test. However, it is also drastically reduced by CNT.

The plasmode method discussed above is applied to lung cancer and multiple myeloma datasets. The information about the simulation is given below.

Simulation 7

The algorithm described above was applied to the lung cancer data and the multiple myeloma datasets to obtain 1000 plasmode datasets. Only steps 1 thru 5 were used, thus the global null hypothesis is true for all plasmode datasets. The mean and the standard deviation of the p-value distribution were generated similar to the earlier case using histograms of 100 bins. Plots in Figure 7.4 show the mean and standard deviation of empirical p-value distributions.

They show that while the mean of the p-values is similar to the independent case, the standard deviations behave differently. It can be suspected that this may be due to the correlations present in the data.

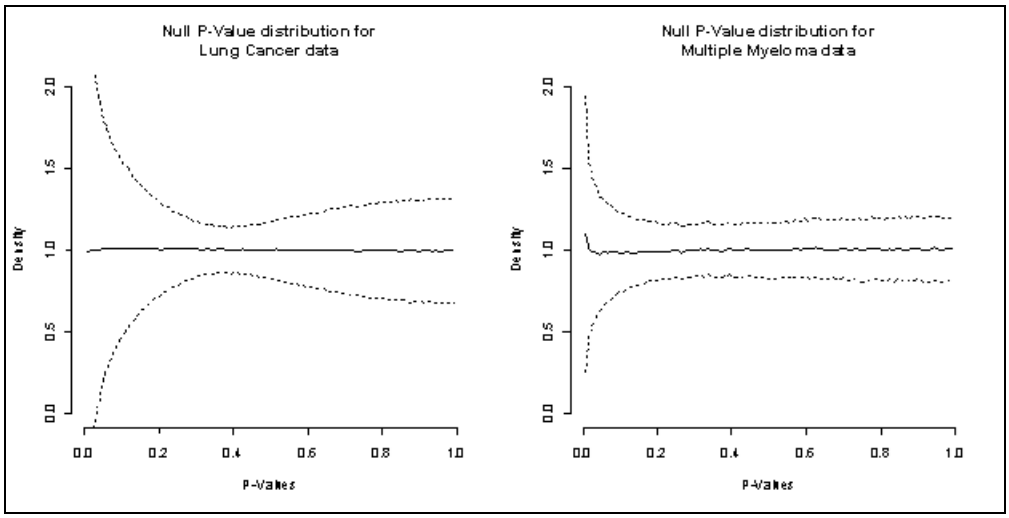

Figure 7.4: Null p-value distributions for plasmode data.
Average p-value distributions from plasmode data for lung cancer and multiple myeloma data shown in black solid lines. The dotted lines indicate the one standard deviation limits. The global null hypothesis is true for the data.

While the mean p-value distribution is being captured by the plasmode method, the process of resampling within groups described above does not always capture the correlation density within groups. This is illustrated using the lung cancer and multiple myeloma datasets and the simulation information is given below.

Simulation 8

Plasmode data for both lung cancer dataset and multiple myeloma dataset are generated using the method described above. 100 plasmode datasets were generated for each of these datasets. The pairwise empirical correlation densities and the density of the combined correlation are computed for these plasmode data. These are illustrated in Figure 7.5 below. The black solid lines indicate the empirical correlation densities for individual groups and the combined correlations for the original samples. The gray lines indicate the empirical correlation densities for the 100 plasmode data sets.

The plots below show that the correlation of the plasmode data generated for the lung cancer data portrays the correlation densities of the original sample. This is true not only for the

combined correlations but also for the correlations within the individual groups. However, this is not the case for the multiple myeloma data. Both individual correlation and combined correlation densities are wider for the plasmode data compared to correlations of the original data. It can be seen Figure 5.4 that the combined correlation densities for the multiple myeloma data (within groups and combined correlations) are much closer to the correlation densities under independence. If repeated samples are drawn from a population of independent features, the sampling distribution of the correlation for these data would match the sampling distribution of the correlations given in (2.18). However, when all features are close to independence, plasmode datasets seem to exhibit higher correlations in their empirical densities than were present in the original data. This increment in correlations causes the combined correlation to increase. This can be illustrated using a simple simulation.

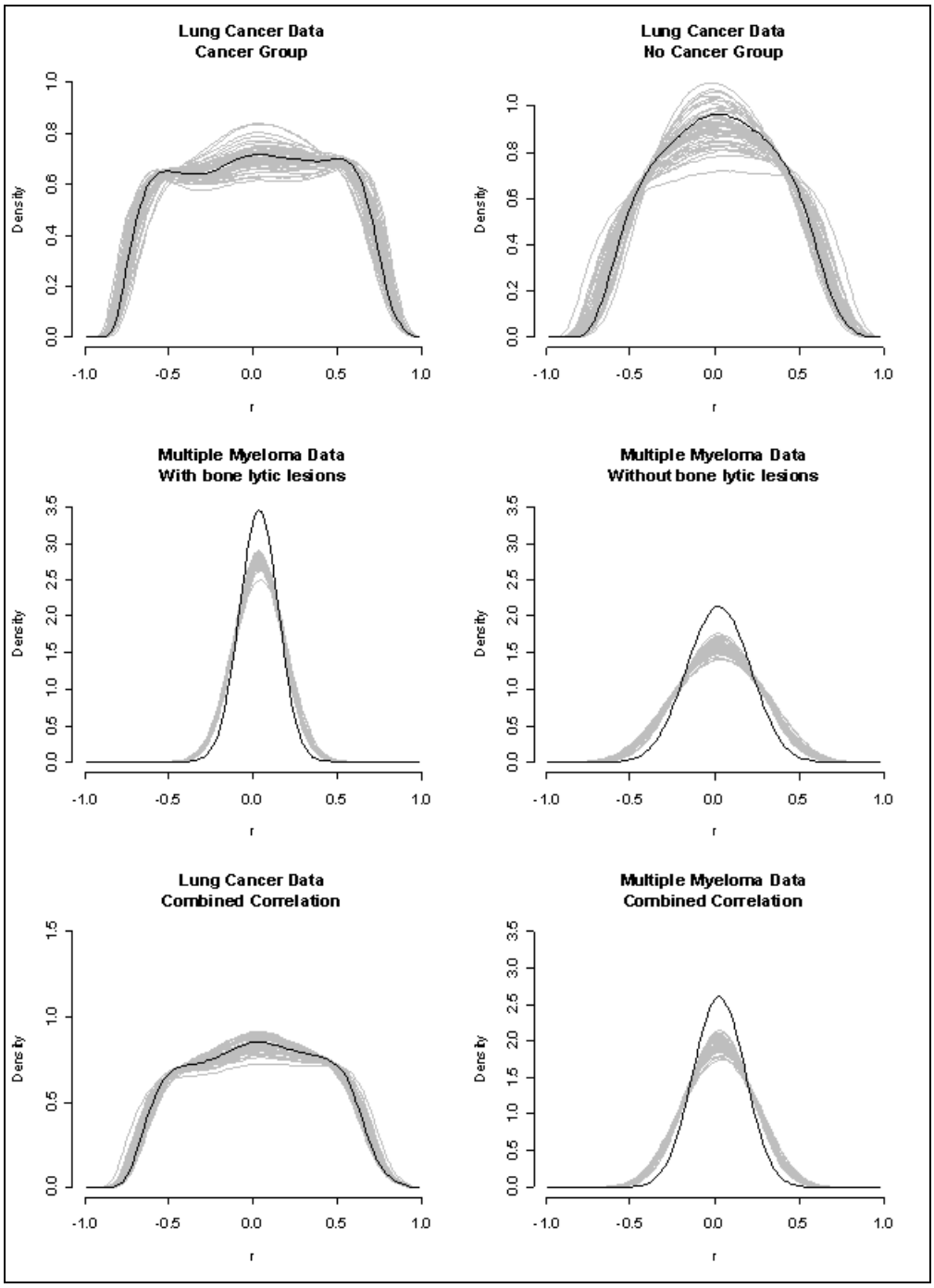

Figure 7.5: Comparison of plasmode correlation distributions.
Empirical correlation densities in individual groups and combined correlation densities obtained by the new plasmode data method. Solid black lines indicate the empirical correlations of the original sample. The gray lines indicate empirical correlations obtained from 100 plasmode datasets. This shows that the plasmode method captures the correlation densities in the lung cancer data, but does not quite capture correlations in the multiple myeloma data.

Simulation 9

100 datasets were simulated using a standard normal density for two groups with 500 independent features and 40 samples in each group. The combined correlations for these data are computed. The empirical densities of combined correlations are illustrated in the first plot of Figure 7.6 below. One of these independent datasets was used as the template for generating 100 plasmode datasets and their combined correlations were also computed. Their empirical densities are illustrated in the second plot. The comparison between the two groups of densities show the higher correlations in the plasmode data, compared to repeated samples.

There are (at least) two possible reasons for this. One is that the resamples in the bootstrap procedure could produce repeats of values that result in higher correlation estimates among features in a HD dataset.

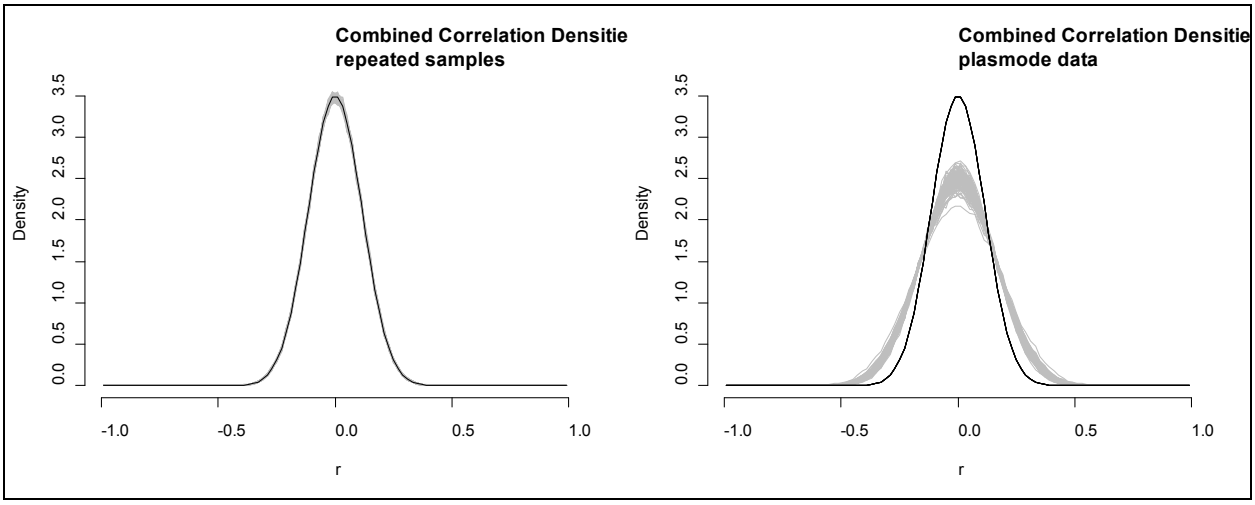

Figure 7.6: Correlation density for plasmode data under independence.
The gray lines in the first plot show the combined correlation densities for repeated independent samples. The solid black line in both plots indicated the distribution of the correlation density under independence. The gray lines in the second plot show the combined correlation densities for plasmode data. This figure illustrates that when the data are independent, the correlation densities of the plasmode data are inflated (have slightly heavier tails).

Another reason is that the resamples from a bootstrap procedure tend to center themselves on the sample estimate from the original data. A simple simulation can be used to explain this.

Simulation 10

20 values were generated for two variables (X and Y, from a standard normal distribution) independently from each other. Although X and Y are independent, the sample correlation between X and Y is not zero due to sampling variability. The samples generated for this simulation resulted a sample correlation coefficient of $r = 0.349$. While this is a much larger estimate of the true correlation of zero, it is not unreasonable to observe such large values for a sample size of 20. These data were used to generate 10000 bivariate bootstrap samples. The mean of the resulting 10000 correlations was 0.340. The true value of that is zero, and is located at the 3rd percentile of this bootstrap correlation distribution. This indicates the entire distribution of correlations has higher correlation values than expected under independence. This effect is not seen when repeatedly generating bivariate independent normal data. Since the bootstrap uses the empirical distribution of the data, its ability to capture the sampling variability of a statistic depends on the empirical distribution. An unusual sample will yield an unusual bootstrap distribution.

Even if a high-dimensional dataset was realized from a large multivariate distribution with many zeros in the correlation matrix, the magnitudes of many estimated correlations will be far from zero as expected. However, bootstrap samples will reflect these estimates and the empirical distribution of correlations will likely have wider tails (on both positive and negative sides) than a distribution of correlations under complete independence. In a dataset that appears to be generated from a distribution with a strong dependence structure, this effect will be less apparent. The magnitudes of estimated correlations will both overestimate and underestimate the corresponding parameter (correlation) in HD data. Thus both a widening and narrowing effect will be present in the tails of the simulated bootstrap distribution of empirical correlations. It is unknown to what extent this effect has been seen or has created issues in the analysis or simulation of high-dimensional data. It needs further investigation. For the purposes here, it is noted that with data that are close to a structure of independence, the plasmode procedure will interject some added dependence structure in simulated data sets. With data that are highly dependent, the plasmode procedure will produce data sets with a similar correlation structure (i.e., similar to that seen in the empirical distribution of original correlations). The CNT procedure deals with dependence in the same way, regardless of whether the dependence is real

or created in the plasmode procedure, and it reduces added variability in the number of discoveries due to dependence as shown in Figure 7.3.

It can be seen in Figure 7.5 that correlation densities for some of the plasmode data are close to the correlation density of the original data while some of them are quite different from the original data. For example, the first plot in Figure 7.5 shows two gray lines peaking above the rest of the densities. One of the ways that can be used to ensure that the correlation densities of the plasmode data represent the correlation density of the original data is to filter out these samples that have a correlation density deviating far from the correlation density for the original data. An area between curves (ABC) measure explained in section 2.5.5 can be calculated for each dataset and simulated data that resulted in a high ABC measure can be filtered out. One way that this can be done is to set a threshold for an ABC measure and discard samples if a particular sample has an ABC value above the threshold. This method can be very inefficient since samples are discarded after all pairwise correlations are computed, which is a time consuming process for large scale datasets. For example, assume that the goal is to obtain 100 plasmode datasets. If the above criteria is used for filtering, it is possible that maybe 150 or more datasets need to be evaluated before obtaining 100 plasmode samples if the ABC threshold is very small. Another possible way to do this is to generate some extra amount of plasmode data than required (say 5% extra datasets) and remove the datasets that result in large ABC measures (5% of the plasmode data that result in the largest ABC measure). If the goal was to obtain 100 plasmode datasets, only 105 need to be evaluated. However, since a threshold for ABC is not set under this scheme, the correlation distributions for some plasmode data may not be as close to that of the original data. The variance of the number of discoveries of the lung cancer data and multiple myeloma data under the global null hypothesis can be investigated using the plasmode method introduced above. The simulation information is given below.

Simulation 11

100 plasmode datasets were generated for each of the two real datasets using steps 1 – 5 (true global null hypothesis). The groups in these datasets (cancer and no cancer groups in lung cancer data and with and without bone lytic lesions in multiple myeloma data) were tested for

mean differences using both t-tests and CNT for comparisons. The mean and standard deviation of the number of discoveries is illustrated in the figure below.

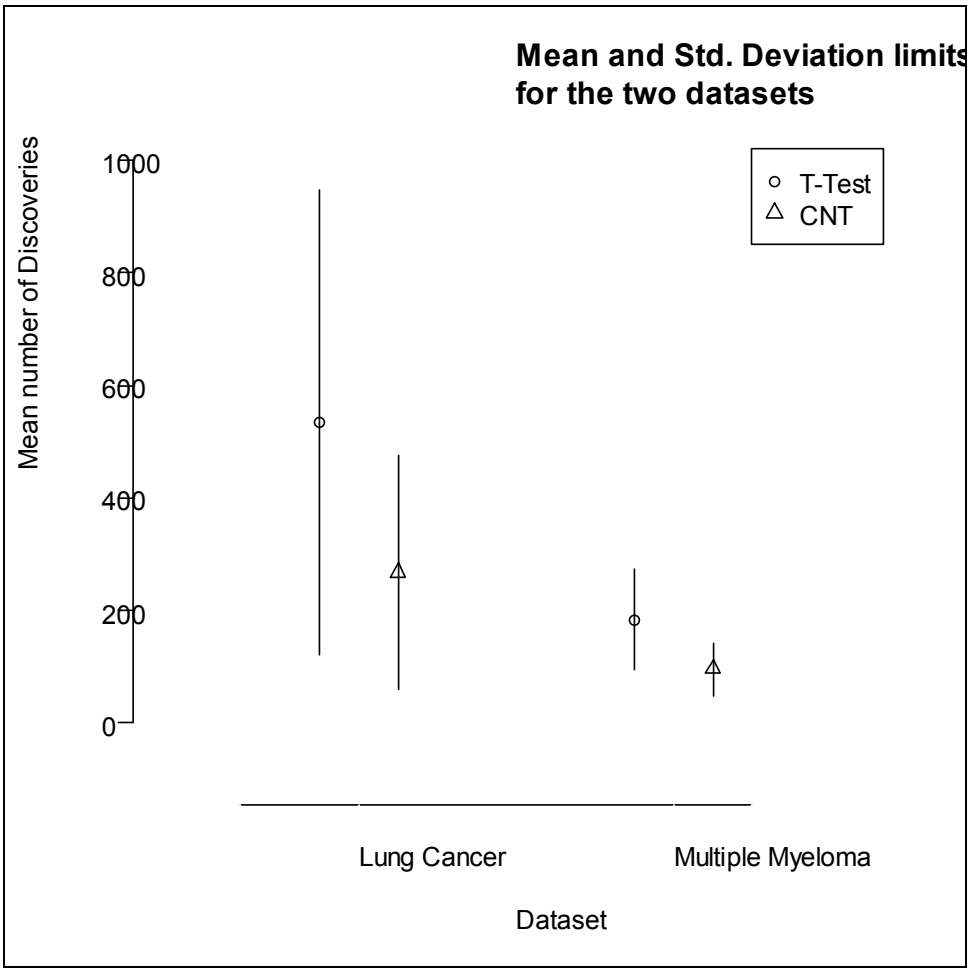

Figure 7.7: The mean and the standard deviation of the number of discoveries for plasmode data.
Results for plasmode data generated using lung cancer data and multiple myeloma data as templates (Steps 1 -5 of plasmode algorithm was used to obtain datasets with a true global null hypothesis). Symbols indicate the mean of the number of discoveries and the vertical lines indicate the one standard deviation limits. Both the mean and the standard deviation of the number of discoveries is reduced by CNT compared to t-tests (all discoveries are type I errors since the global null hypothesis is true). At $\alpha = 0.05$, under the true global null hypothesis, the lung cancer data are expected to produce 630 discoveries and the multiple myeloma data are expected to produce 198 findings from the t-tests.

The figure above illustrates that the standard deviation of the number of discoveries is reduced by the use of CNT. The mean of the number of discoveries is also reduced. The large standard deviation in lung cancer data can be explained by the strong dependence structures illustrated in Figure 7.1. The standard deviation of the number of discoveries in the multiple

myeloma dataset is quite small and this can be expected since the dependence structure is much closer to independence. The standard deviation shown in Figure 7.7 for the t-tests may be higher than the true (population) standard deviation, since the plasmode data generated from datasets with independent (or close to independent) features have an inflated variance of the number of discoveries when the t-test is used. The comparison of results between the lung cancer and multiple myeloma datasets must be done with caution since they have different dimensions and dependence structures.

This chapter suggested a method to simulate data, attempting to preserve the correlation structures within the groups being compared. It was shown that the method correctly captures the dependence structures when the correlations are very high and deviate far from independence. However, when the correlations are closer to independence, the method results in plasmode datasets with slightly wider correlation densities compared to the original data. This method can be employed to evaluate statistical methods that compare the mean of two groups under dependencies, but caution must be taken when making inferences about the variance of the measures, since some dependence structures may be exaggerated when the original data are closer to independence.

The following chapters explore the use of CNT and t-tests with other error control methods. Simulations are used to illustrate the results and plasmode datasets are used to make inferences about the results obtained for the lung cancer data and multiple myeloma data.

Chapter 8 - Effects of Conditional Network Testing

Chapter 6 introduced the concept of testing dependent features conditionally within networks determined by the combined correlation between test statistics. This chapter further investigates the characteristics of conditional network testing and its effects on some accuracy measures.

8.1 Effect on the False Discovery Proportion, Type I and II Errors

The performance of testing methods can be evaluated using their type I error, type II error, and false discovery proportions. Simulations are employed to evaluate the performance of CNT compared to t-tests in the following sections. Simulation information are provided below.

Simulation 12

Data were simulated for dependence structures described in Table 4.1 using a multivariate normal distribution for two groups, 40 subjects in each groups and 500 features for each subjects. Data for each dependence structure were simulated 200 times to compute the mean and standard deviation of the accuracy measures of interest. On average 10% of the features were simulated differently between the two groups. Feature locations were randomly selected and 0.8 was added to the mean vector of those features in one group while all the other elements of the mean vector remained at zero. This results in a mean difference of 0.8 for the selected 10% of the features. This mean difference results in a difference of 3.5 on the t-statistic scale. Testing methods were assessed by their ability to correctly identify these features. The following table can be used to define accuracy measures of interest

Table 8.1: Notations for accuracy measures

		True Situation		Total
		True H_0	True H_a	
Test Result	Reject H_0	V	S	R
	Do not reject H_0	U	T	P
	Total	L	M	K

Using Table 8.1, the following definitions for key proportions can be given:

False discovery proportion = V/R

Type I error = V/L

Type II error = T/M

Since the data are simulated, all values in Table 8.1 are known. However, for a real dataset only K, R and P are known. Both t-tests and CNT are used to analyze simulated data and the following sections discuss the effect of CNT on type I error, type II error and false discovery proportions.

8.1.1 Effect of CNT on False Discovery Proportion

Results from simulation 12 are used to compare false discovery proportions between t-tests and CNT. Since the features that were generated differently are known, the quantity V/R in Table 8.1 can be calculated for each simulated dataset. Tests were conducted at $\alpha = 0.05$ level. This procedure generates 200 false discovery proportions for each dependence structure, which can be used to study the behavior of the false discovery proportion under different dependence structures. The first plot of Figure 8.1 below illustrates the behavior of the mean false discovery proportion. The standard deviation of the false discovery proportion is shown in vertical dotted lines. CNT results in slightly smaller standard devations compared to the t-tests.

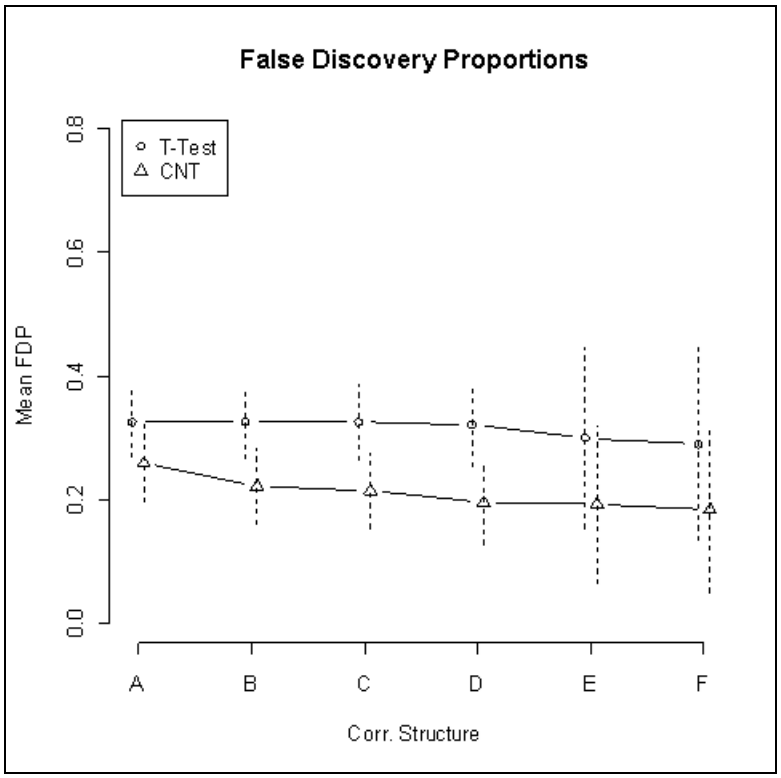

Figure 8.1: Average false discovery proportions (FDP) comparing t-test and CNT.
Tests are done at $\alpha = 0.05$ level. The line with the circle symbol indicates the FDP using the t-test and the line with the triangle symbol indicates the FDP using CNT. A-F are correlation structures defined in Table **4.1**. The vertical dotted lines indicate a 1-standard deviation limit at each dependence structure.

This figure illustrates that the proportion of false discoveries made by CNT on average is smaller than the proportion of false discoveries made by the t-tests. This illustrates that CNT not only reduces the variance of the number of discoveries but also reduced the average proportion of false discoveries made. However, at strong dependence structures (indicated by E and F in Table 4.1 and Figure 8.1) the average FDPs falls within 1-standard deviation limits of both methods. Figure 8.2 below shows the FDPs made in each simulation under different dependence structures.

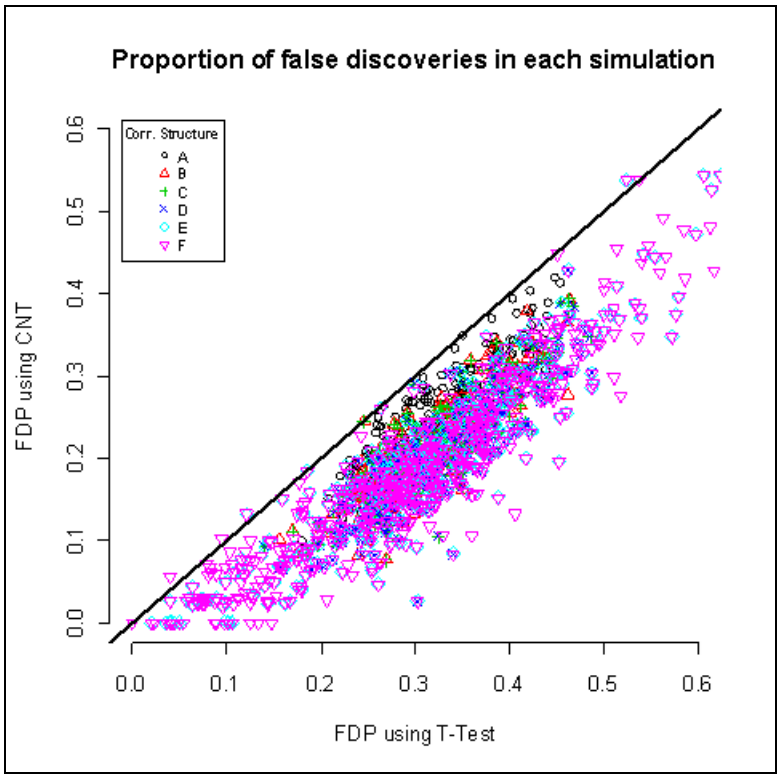

Figure 8.2: Proportion of false discoveries for each simulated dataset.
Proportion of false discoveries made in the analysis of each simulated dataset by the t-test and CNT. Figure illustrates that CNT results in a less proportion of false discoveries compared to using the t-test.

Figure 8.2 lists the proportion of false discoveries made in all 1200 simulations (6 dependence structures × 200 simulations). This shows that not only the average false discovery proportion is reduced, but the false discovery proportion in the analysis of each dataset is also reduced. The plot also shows that in rare occasions, the false discovery propotion using CNT is slightly higher than the false discovery proportion using the t-test. The coded symbols show that this only occurs in the strongest dependence structure defined in Table 4.1.

8.1.2 Effect of CNT on Type I Error Proportion

Information from simulation 12 can be used to determine the behavior of the type I error proportion of the tests. Since V and L are known in Table 8.1, type I error proportions, their means and standard deviations can be easily computed for simulated data. Figure 8.3 below illustrates the behavior of the type I error proportion under the different dependence structures considered.

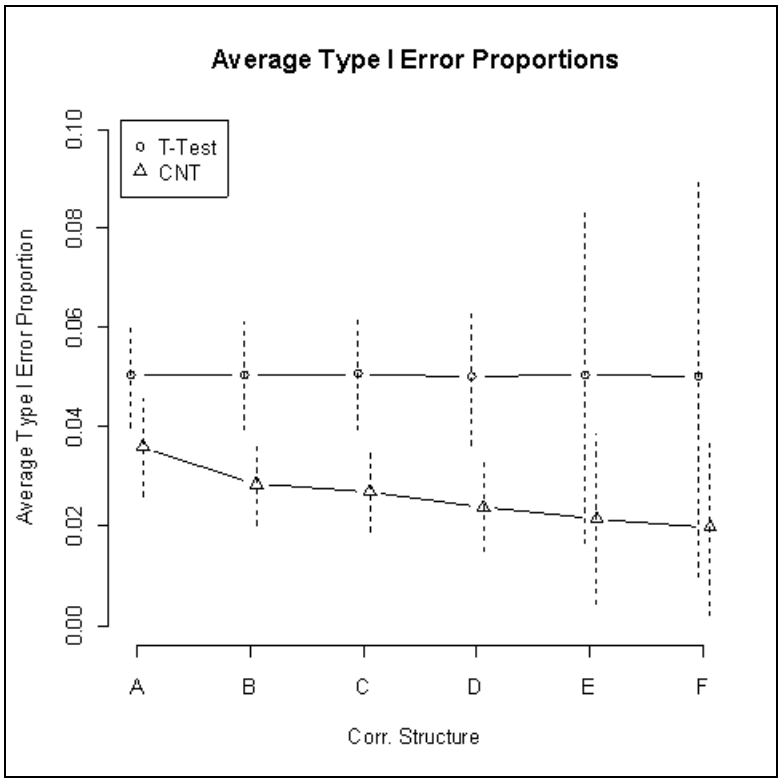

Figure 8.3: Average type I error proportions.
Average type I error proportions using the t-test and CNT. The Figure shows that both average type I error proportions and its variance is reduced by CNT compared to the t-test. (Tests were done at $\alpha = 0.05$ level)

Figure 8.3 above shows that the average type I error proportions made by the t-test is fairly consistent across the different dependence structures while the variance increases with the increasing strength of the dependence structure. CNT has reduced both the average type I error proportion and its standard deviation (shown in vertical lines) compared to the t-test.

8.1.3 Effect of CNT on Type II Error Proportion

Similar to the type I error proportion, the type II error proportion can also be computed by the available information in simulation 12. Figure 8.4 below illustrates the behavior of type II error proportions using the t-test and using CNT.

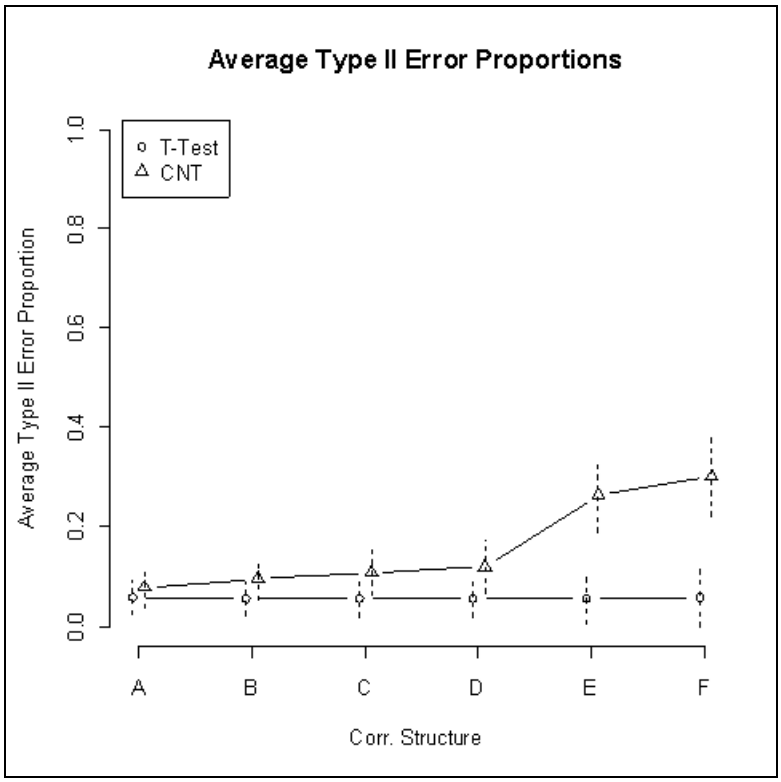

Figure 8.4: Type II error proportions.
Type II error proportions for the t-test and CNT (at $\alpha = 0.05$). The Figure illustrates that there is an increment in the type II error proportion when using CNT versus individual t-tests. The increment is largest at the strongest dependence structure.

Figure 8.4 above shows that on average CNT makes more type II errors compared to the t-test. This is because the null hypothesis for the features with small differences between the two groups fail to reject, once conditioned on another features. The standard deviation of the type II error proportion is also slightly increased when CNT is used.

The discussion above shows that the proposed method stabilizes the variance of the number of discoveries by suppressing some of the false discoveries made by the t-test. However, this process causes the type II error proportion to increase, since the computed p-values,

conditioned on another feature, are larger. In comparison, the average positive predictive value (i.e., the proportion of truly different features among the set of features declared statistically significant, *S*/*R* in Table 8.1. This is equivalent to the the quantity one minus the false discovery proportion) is higher for CNT than it is for t-tests. It can be assumed that the power of the test is impacted by the suggested method. This is discussed further in the next section.

8.2 Effect on Power at Local Alternatives

The power is defined as the probability of a test rejecting the null hypothesis when the null hypothesis is false (i.e., not committing a type II error). The power of the test depends on the magnitude of the actual difference, and it is equivalent to the *size* of the test under the null hypothesis. Since the test using the conditional density described in Chapter 6 depends on the correlation between test statistics, only a pair of tests is considered for evaluating the power. The conditional density given in (4.11) and (4.14) assumes that the mean differences in the i^{th} and j^{th} features are zero across the two treatment groups. Densities in (4.11) and (4.14) can be derived keeping the assumption of zero mean difference for the i^{th} feature but allowing a certain mean difference across treatment groups for the j^{th} feature. Let the standardized difference between the two groups for the j^{th} feature be λ_j, then

$$\frac{\mu_i - \nu_i}{\sqrt{\sigma_i^2 + \tau_i^2}} = 0 \text{ but } \frac{\mu_j - \nu_j}{\sqrt{\sigma_j^2 + \tau_j^2}} = \lambda_j$$

in (4.1) and (4.2), and the joint density of test statistics given in (4.5) can be written as

$$\begin{pmatrix} z_i = \dfrac{\bar{x}_i - \bar{y}_i}{\sqrt{\sigma_i^2 + \tau_i^2}} \\ \\ z_j = \dfrac{\bar{x}_j - \bar{y}_j}{\sqrt{\sigma_j^2 + \tau_j^2}} \end{pmatrix} \sim N\left(\delta'' = \begin{bmatrix} \dfrac{\mu_i - \nu_i}{\sqrt{\sigma_i^2 + \tau_i^2}} = 0 \\ \\ \dfrac{\mu_j - \nu_j}{\sqrt{\sigma_j^2 + \tau_j^2}} = \lambda_j \end{bmatrix}, \Sigma'' = \begin{bmatrix} 1 & \rho_z \\ \rho_z & 1 \end{bmatrix} \right)$$

Note that under the same notation, ρ_z is unchanged by the change in mean structure. By following a similar derivation, the density in (4.11) can be written as

$$f_{z_j|z_i<-z^* \text{ or } z_i>z^*}(z_j \mid z_i < -z^* \text{ or } z_i > z^*; \lambda_j) =$$

$$\frac{\phi(z_j-\lambda_j)\left[1+\Phi\left(\frac{-z^*-[\rho_z(z_j-\lambda_j)]}{\sqrt{1-\rho_z^2}}\right)-\Phi\left(\frac{z^*-[\rho_z(z_j-\lambda_j)]}{\sqrt{1-\rho_z^2}}\right)\right]}{\Phi(-z^*)+1-\Phi(z^*)} \tag{8.1}$$

And the density in (4.14) can be rewritten as

$$f_{z_j|-z^*<z_i<z^*}(z_j \mid -z^* < z_i < z^*; \lambda_j) =$$

$$\frac{\phi(z_j)\left[\Phi\left(\frac{z^*-[\rho_z(z_j-\lambda_j)]}{\sqrt{1-\rho_z^2}}\right)-\Phi\left(\frac{-z^*-[\rho_z(z_j-\lambda_j)]}{\sqrt{1-\rho_z^2}}\right)\right]}{\Phi(z^*)-\Phi(-z^*)} \tag{8.2}$$

(8.1) and (8.2) can be used to investigate the power of tests at different alternatives (λ_j values). The conditional probability of rejecting a feature with a standardized difference of λ_j (power for detecting a λ_j difference) in the jth feature, conditioned on the status of the ith feature, can be obtained by integrating above densities. For a level α test, if the ith feature was declared significant, then the power of detecting a λ_j standardized difference in the jth test is given by

$$1-\beta = \int_{-\infty}^{-z^*} f_{z_j|z_i<-z^* \text{ or } z_i>z^*}(z_j \mid z_i < -z^* \text{ or } z_i > z^*; \lambda_j)dz_j + \int_{z^*}^{\infty} f_{z_j|z_i<-z^* \text{ or } z_i>z^*}(z_j \mid z_i < -z^* \text{ or } z_i > z^*; \lambda_j)dz_j \tag{8.3}$$

where - z^* and z^* are lower and upper cut off limits for level α. Similarly, if the ith feature was not declared significant, then the power of detecting a λ_j standardized difference in jth feature is given by

$$1-\beta = \int_{-\infty}^{-z^*} f_{z_j|-z^*<z_i<z^*}(z_j \mid -z^* < z_i < z^*; \lambda_j)dz_j + \int_{z^*}^{\infty} f_{z_j|-z^*<z_i<z^*}(z_j \mid -z^* < z_i < z^*; \lambda_j)dz_j \tag{8.4}$$

(β in (8.3) and (8.4) is defined as the probability of making a type II error). The integrations in (8.3) and (8.4) can be evaluated at different levels of λ_j to study the behavior of power at

different effect sizes. Since closed form solutions to these integrals do not exist, their values are obtained through numerical methods. Figure 8.5 below illustrates the power curves for three scenarios for tests performed at level $\alpha = 0.05$. The combined correlation between i^{th} and j^{th} features was set to 0.6, and the figure illustrates these curves for different values of λ_j.

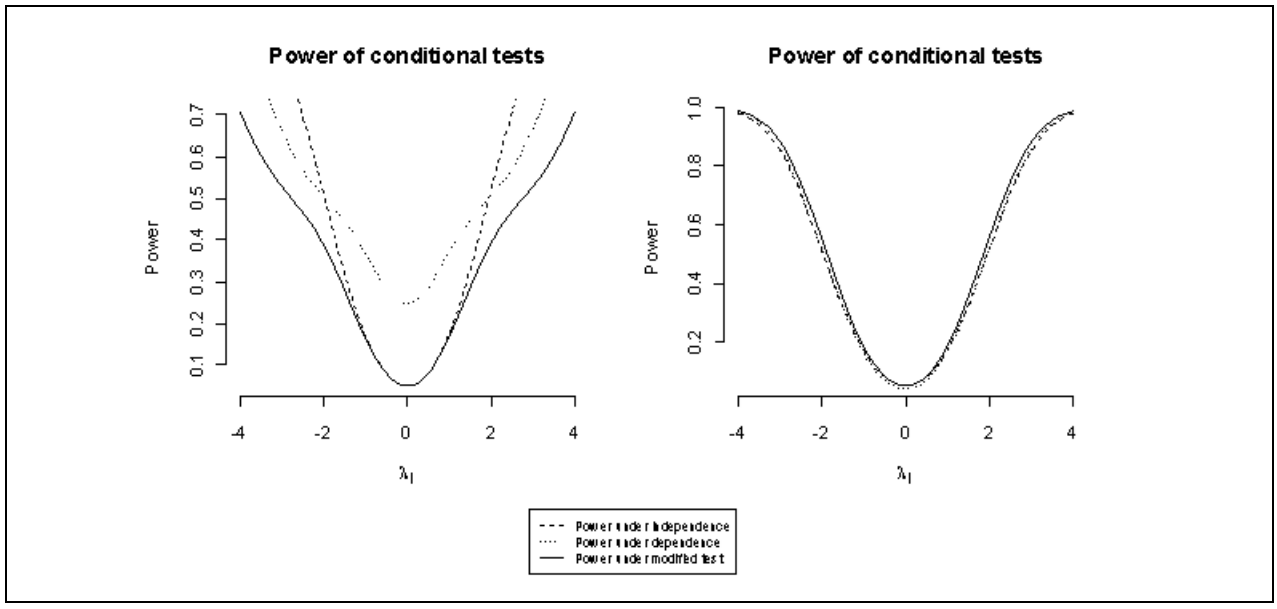

Figure 8.5: Power curves for CNT.
Power for testing λ_j difference at level 0.05. The first plot is obtained by evaluating (8.3**)** and the second plot is obtained by evaluating (8.4**)** over the shown range. The dashed lines indicate the power of the j^{th} test if it is independent of an i^{th} test. It is equivalent to the power of the regular marginal t-test. The dotted lines indicate the power of the j^{th} test conditioned on the i^{th} test when the combined correlation between the i^{th} and j^{th} test is equal to 0.6. The first plot shows that when the null hypothesis for the i^{th} test is rejected, the power of the correlated j^{th} test increases at all levels of λ_j, including $\lambda_j = 0$ in which case the type I error is elevated. The second plot shows that the power is not very different from the independent case when the null hypothesis for the i^{th} test fails to reject. The solid lines show the power under CNT. The first plot shows that the power is reduced and the level of the test is 0.05, but some power to detect a large difference is lost. The second plot shows that there is little difference between power curves.

The plots above show the behavior of power under different conditions. The dashed lines in both plots indicate the power of a test under a condition of independence with another test. This shows that when the features are independent, the power curve is that of a regular t-test. The dotted curve in the first plot indicates the power of the j^{th} test when the null hypothesis for the correlated i^{th} test is rejected. This shows that the power for the j^{th} test increases at all levels of λ_j including zero, in which case the type I error rate is elevated. This illustrates the increment of type I errors under dependencies. The power is not affected when the i^{th} test fails to reject the

null hypothesis as shown in the second plot. The solid curves in the plots show the power under CNT. The first plot shows the decrement in power, and the power for testing a zero mean difference is equal to α (i.e., the Type I error is restored to nominal values for the conditional test). However, the test loses power to detect large differences in effects compared to the independent case. This loss in power explains the increased type II error rate in Figure 8.4. The second plot shows the power is not affected when the i^{th} test fails to reject the null hypothesis but slight gains in power can be seen under CNT. Note that the integrals in (8.3) and (8.4) are evaluated numerically to obtain curves in Figure 8.5. These curves are subject to some variability due to these numerical computations.

The material above investigated the behavior of false discovery proportion, type I error proportion, and type II error proportion using data simulated with systematic dependence structures. It was discussed above that while these dependence structures allow basic comparisons between methods, they are not representative of dependence structures in real data. Therefore, the plasmode data method introduced in Chapter 7 is used to simulate data using lung cancer data and multiple myeloma data as templates.

Simulation 13

Plasmode data were generated using lung cancer and multiple myeloma datasets as templates to illustrate the effects of CNT on datasets with more realistic dependence structures. The steps described in the plasmode algorithm were used and 10% of the features were generated with a mean difference between the groups (p = 0.1 in step 8). Since the location of these features are known, error measures such as the false discovery proportion, type I errors, type II errors can be calculated as above. The figure below summarizes the results from 100 simulated plasmode datasets.

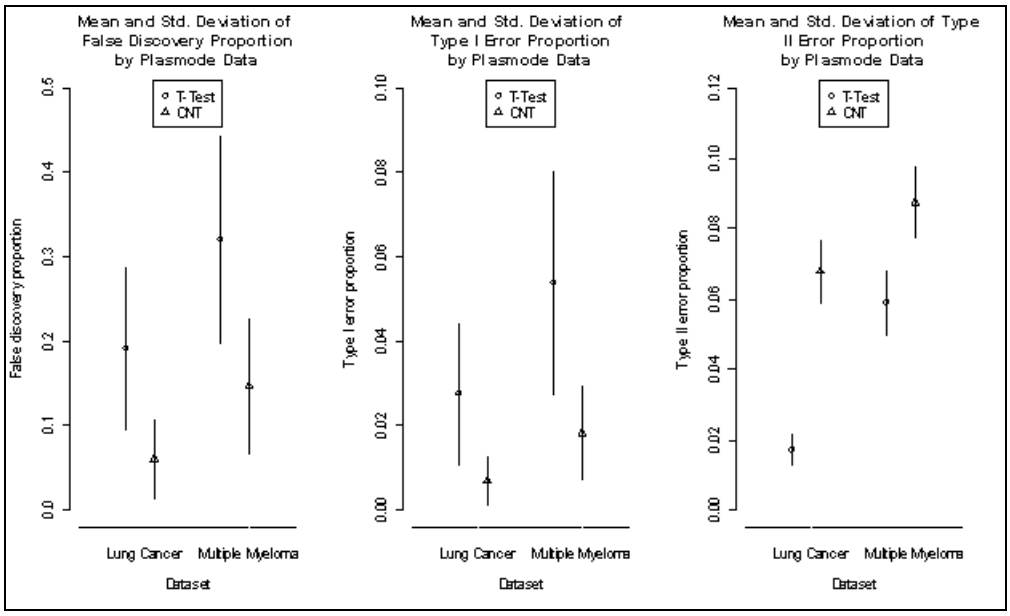

Figure 8.6: Comparison of false discovery proportions, type I and type II error proportions.

False discovery proportion, type I error proportion and type II error proportions by plasmode data. Results are based on 100 plasmode datasets. (Tests were conducted at $\alpha = 0.05$). There is a reduction in false discovery proportion and type I error proportion but increment in type II error. The standard deviations (vertical lines) are reduced for false discovery proportion and type I error proportion for CNT compared to t-test. The standard deviation of type II error proportion for lung cancer data is increased for CNT while that for multiple myeloma data is about the same for both t-tests and CNT. The percentage changes are listed in Table **8.2** below.

The figure above illustrates that plasmode data behave the same way as simulated data in earlier simulations. There is a reduction in false discovery proportion and type I error proportion and their standard deviations when CNT is used. Type II error proportion and its variance on the other hand are increased. The percentage changes in results using CNT versus using t-tests are given in the table below.

Table 8.2: Percentage changes in accuracy measures

Data	FDP	Type I Error	Type II Error
Lung Cancer	62% (33%)	70% (59%)	200%* (93%*)
Multiple Myeloma	54% (35%)	66% (58%)	48%* (8%*)

All numbers are percent reductions except type II error values (indicated with a *).

This chapter explored the behavior of some error rates that can be used to assess the performance of the proposed method. They were done using both simple simulations and

plasmode data. It was illustrated that there is a reduction in average false discovery proportions and type I error proportions. Their standard deviations are smaller for CNT compared to t-tests. However, there is an elevated average type II error proportion when CNT is used. The standard deviation of the type II error proportion is also increased for CNT compared to t-tests. The increments are comparatively smaller for the datasets that used multiple myeloma data as the template but are higher for datasets that used the lung cancer data as the template. The increased type II error proportion can be expected since the type I error proportion and false discovery proportions are improved (smaller). The next chapter further investigates these measures under different FDR control methods that are often used in applications.

Chapter 9 - FDR Control with Conditional Network Testing

Chapter 6 introduced the concepts of conditional network testing and illustrated some of the initial results of using this proposed method. Chapter 8 evaluated the new method alongside the regular t-test, comparing false discovery proportion, type I and type II errors. This chapter compares results when FWER and FDR control methods are applied to CNT and t-tests. Namely Bonferroni, Benjamini & Hochberg (BH), Benjamini & Yekutieli (BY), and Storey's Q-Value method are considered for comparisons. Bonferroni and Holms methods attempts to control the FWER, thus they are quite conservative when compared to BH, BY and Q-Value methods which attempt to control the FDR.

The estimates of FDR are also evaluated under the t-test and CNT. Efron's estimate of FDR under dependence (Efron, 2007), and Storey's estimate of FDR (Storey, 2002) are computed. Storey's estimate of FDR is discussed in Chapter 2 - and shown in (2.6). Efron's estimate takes the dependencies among features into account and is also discussed in Chapter 2 - . This estimate is given in (2.11**)**. Efron's estimate is in fact a correction to an existing FDR estimate. FDR_0 in (2.11**)** is an initial estimate of FDR. In this work Storey's estimate of FDR is used as FDR_0, the initial estimate in Efron's technique.

9.1 Mean and Variance of Number of Discoveries

In order to study the mean and variance of the number of discoveries made by each method under consideration, simulation 2 in page 34 is repeated.

Simulation 14

The features are generated with equal mean vectors similar to simulation 2. FWER and FDR control methods were applied to the results obtained from t-tests and CNT. Figure 9.1 below shows the mean and variance of the number of discoveries made by each control method. Since the data were generated with no difference, all 'discoveries' at level 0.05 are type I errors. The control methods make a very few discoveries in all dependence structures considered. Among them, the two methods that control FWER, Bonferroni and Holm's methods do not declare any features as statistically significant. On average, the number of discoveries (i.e., type I errors) made by CNT is smaller than the number made by the t-test (i.e., less type I errors). The

variance of the number of discoveries for the t-test and CNT are as shown in Figure 6.3, where the CNT substantially reduced the variance of the number of type I errors. The error control methods under consideration have a very small variance compared to the t-test or CNT. This is due to the fact that these methods make a very small number of discoveries when applied to a dataset with a true global null hypothesis. The FWER control methods (Bonferroni and Holm methods) show a very high conservative nature when applied to high dimensional data, thus will not be considered for further evaluation.

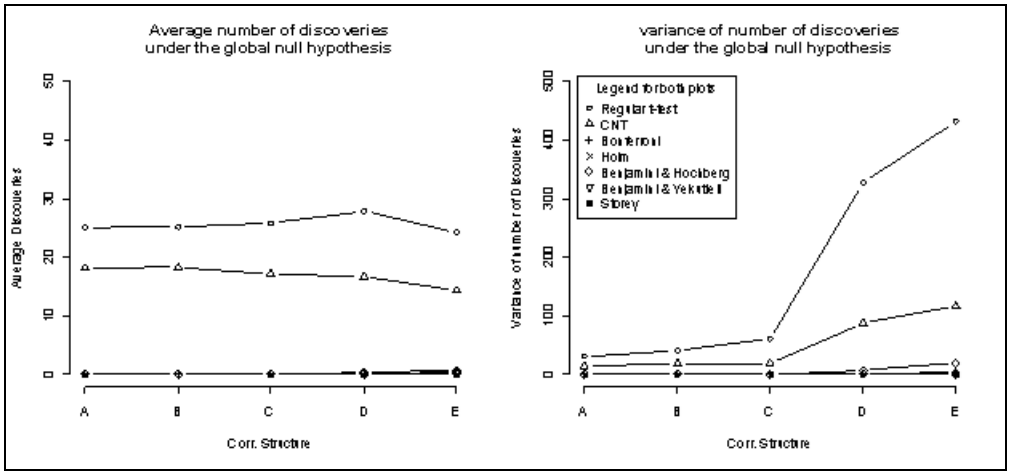

Figure 9.1: Mean and variance of the number of discoveries under different error control methods.
Simulations were done with a true global null hypothesis (i.e., no features are different between the groups). Bonferroni, Holm, BH, BY and Storey's methods (used on the set of p-values obtained from t-tests) are conservative and only make a very small number of type I errors. T-test on the other hand makes the expected number of type I errors at all correlation structures (25 type I errors among 500 features tested at $\alpha = 0.05$). The second plot shows the increment of the variance of the number of discoveries with increasing strength of dependencies.

The FDR control methods, BH, BY and Storey's Q values method operates on the set of p-values obtained from a series of t-tests (or any other type of test) and they adjust the significance level for an individual test. The plot in Figure 9.1 can be a little misleading in that the lines for the regular t-test and CNT are at significance levels that are unadjusted for multiple tests. The other lines are adjusted for multiple testing and, thus, use much lower significance levels for each test, the level depending on the method. CNT yields an adjusted set of p-values for the features being tested. The adjustment is based on correlation structure rather than simultaneous testing. So the above FDR control methods can also be used on the set of p-values resulting from CNT as a way of controlling FDR in the results and accommodating correlation

structure. The following sections compare the results of applying FDR estimation methods and FDR control methods applied to the p-values of t-tests and p-values of CNT.

9.2 Estimates of False Discovery Rate

True values and estimates of FDR can be compared with each other for studying the behavior of FDR and its estimates under different dependencies. A simulation is used to empirically evaluate the estimates of FDR. The results in this chapter and Figures 9.2 – 9.7 are obtained from the following simulation.

Simulation 15

Simulation 12 (page 77) is repeated for evaluating these methods with 10% of the features simulated with a mean difference between the groups and tested at $\alpha = 0.05$ level. This simulation is used to evaluate the FDR estimates in this section and the evaluation of accuracy measures in the following sections.

Since the data are simulated with known differences between the groups, they can be used to obtain the true false discovery proportion. The two estimates considered are Efron's estimate and Storey's estimate that can be evaluated using these known mean structures. Efron's estimate operates on the values of test statistics. Since the test statistic values for CNT are the same as those for a regular t-test, it will only be computed once. Storey's method operates on the p-values of the tests, therefore estimates of FDP can be computed separately for the t-test and CNT. Figure 9.2 below compares these estimates with the true proportions of false discoveries.

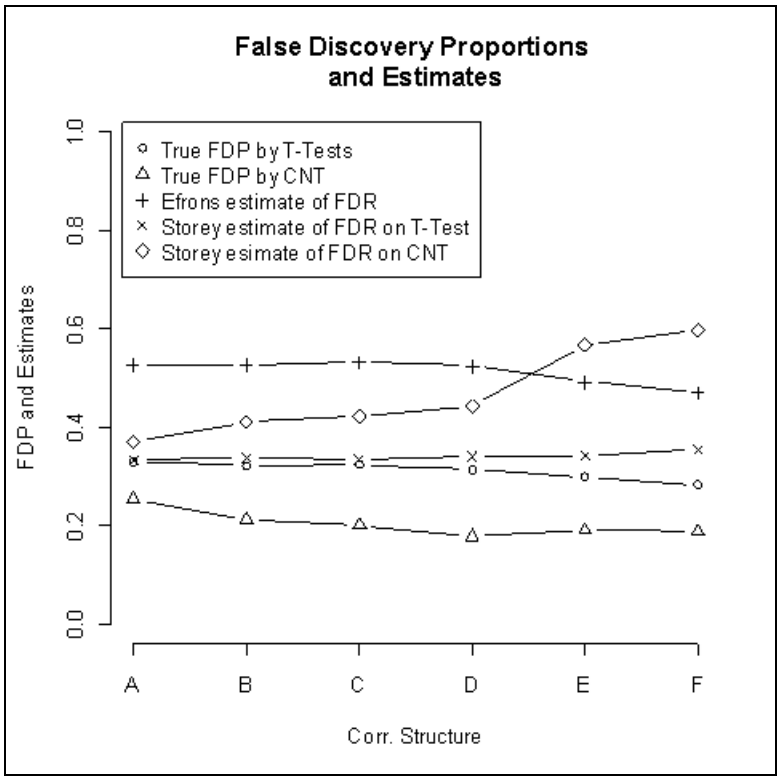

Figure 9.2: Comparison of true and estimated false discovery proportions.
Average false discovery proportions from 200 datasets for each dependence structure using the t-tests and CNT are the same as Figure **8.1** (Circle and triangle symbols in the plot). The Storey's estimate closely estimates the false discovery proportion by the t-tests but over estimates when used on the p-values from CNT.

The figure above illustrates that the FDP estimate put forward by Storey closely follows the average false discovery proportion made by the t-tests. It is almost exactly the same amount when under independence (correlation structure A). The estimate stays constant for increasing dependence structures while the average proportion of false discoveries decreased by a very small amount. However, these are empirically computed and subject to a certain degree of sampling variability. This method over estimates the proportion of false discoveries when used on the set of p-values obtained from CNT. CNT adjusts the p-values obtained from a series of t-tests and some of the p-values that were smaller are changed to be larger p-values (but only a small proportion of larger p-values are made smaller). This weakens the *signal* analyzed by Storey's method for determining the proportion of true null hypotheses. Since the number of smaller p-values is reduced, Storey's method decides that a larger portion of the features declared significant are false findings, resulting in larger estimates of the false discovery rate. In

addition, Storey's method assumes a fixed rejection region where CNT uses different rejection regions based on conditional distributions.

Efron's estimate, which is an adjustment to Storey' estimate, only operates on the values of the test statistics. They are unaltered by CNT and are only calculated once. The estimate for these simulated data overestimates the false discovery proportion. This estimate is subject to two parameters x and x_0 given in (2.11) and (2.12**)**. They may be adjusted for each dataset but used the same values (x = 1.96 and x_0 = 1) for all simulated datasets. The following sections compare false discovery proportions, type I and type II error proportions for FDR control methods applied to the t-tests and CNT

9.3 False Discovery Proportions

The above section investigated "estimates" of false discovery rates applied to t-tests and CNT. CNT results in a different set of p-values drawn from different distributions and allows the application of any FDR control method that operates on p-values. This section investigates the false discovery proportions obtained by applying FDR "control methods" to the set of p-values attained by the regular t-tests and the p-values obtained by CNT. Since the simulations are conducted so that the truly different features are known, the true false discovery proportions can be calculated for each dataset.

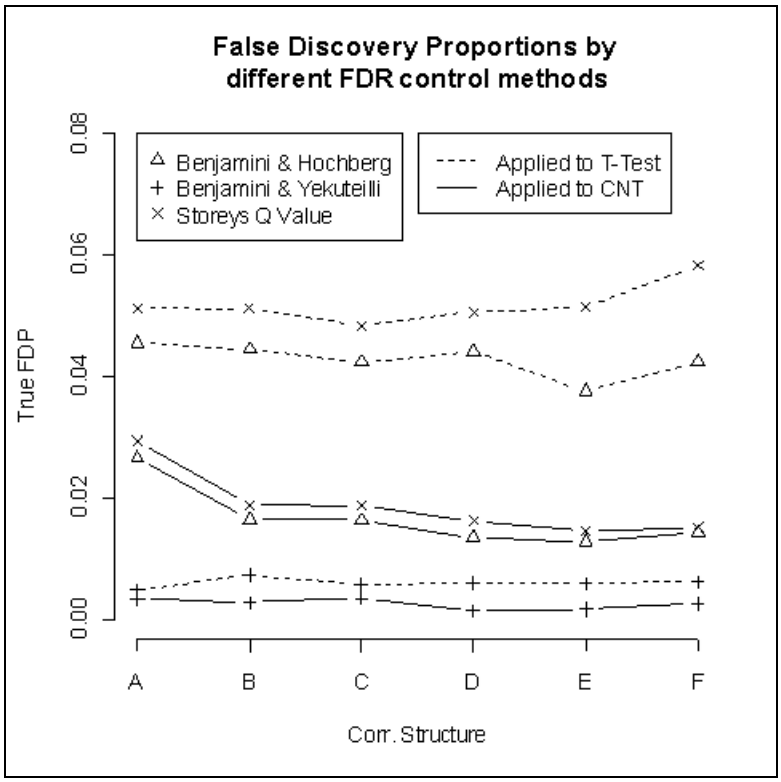

Figure 9.3: Comparison of false discovery proportions.
Comparison of false discovery proportions by BH, BY and Q-Value methods applied to t-tests and CNT. FDR control level is set to 0.05. Y axis is the true FDP averaged over 200 datasets for each dependence structure.

Figure 9.3 shows that CNT results in a smaller proportion of false discoveries under any FDR control method compared to the t-test. In addition, this figure shows that the BH and Storey's Q value methods behave similarly when applied to the p-values obtained by CNT. The Q-Value and BH methods produce largely different results between t-tests and CNT. This difference is very small when BY method is applied. BY method adjusts for the dependence structure and it may be the reason for the similarity seen in false discovery proportions. The application of CNT not only improves the false discovery proportion, but also stabilizes the results as shown in the next figure. It is important to note that Figure 9.2 and 9.3 show related, but still distinctly different concepts. Figure 9.2 regards FDP (false discovery proportion) estimation where a significance level is set (0.05 in cases here) and the true FDP for both t-tests and CNT can be determined. Estimates of FDP using different estimation methods at that significance level can also be computed. Figure 9.3 regards FDR control where the significance level is adjusted to a smaller value based on a desired level of false discovery rate. The adjustment may differ depending on the method of FDR control. Significance levels are much

smaller, but the true FDP in simulations can still be determined based on discoveries found by the different methods.

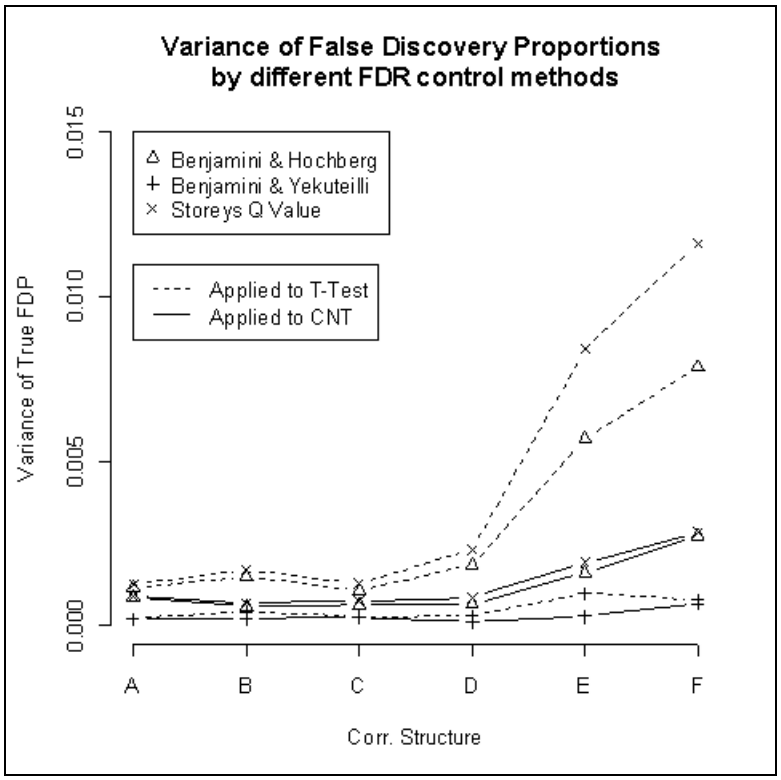

Figure 9.4: Comparison of variance of number of discoveries.
Variance of the false discovery proportions (over 200 simulated datasets for each dependence structure) by applying FDR control methods to t-tests and CNT. The variance for CNT is lower than it is for t-tests for all methods being compared.

This figure illustrates that the proportions of false discoveries is more stable when FDR control methods are applied to the p-values obtained from CNT than by t-tests. BH and Q-Value methods again show similar results. The variance of the false discovery proportion shows little difference between the t-tests and CNT when BY method is applied. The following section investigates the behavior of type I errors under these FDR control methods.

9.4 Type I Error Proportion

The type I error proportion is defined in Table 8.1 and can be calculated for simulated data since the features simulated differently are known. The plot below compares the behavior of average type I error proportions by applying the three FDR control methods to t-tests and CNT.

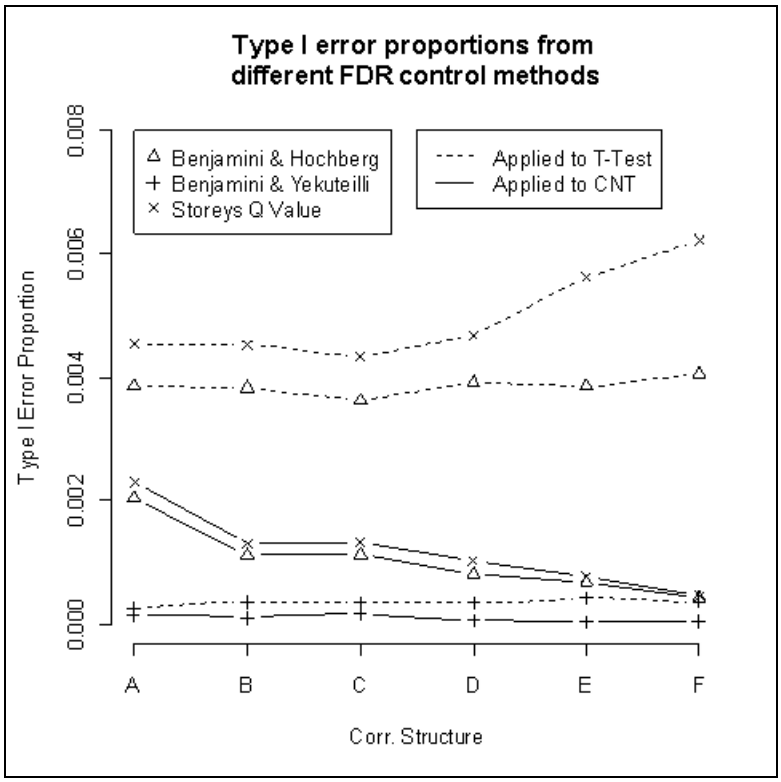

Figure 9.5: Comparison of type I error proportions.
Average type I error proportions for FDR control methods applied to t-tests and CNT. Type I error proportions are computed according to the definition in Table **8.1**. BH, BY and Q-Value methods are controlling FDR at 0.05 level. Data for each dependence structure was simulated 200 times to obtain the averages.

Above figure shows that the type I error proportion is reduced by a large amount when the Q-Value method and BH method are applied to CNT instead of t-tests. There is not much difference when BY is applied to CNT and t-tests. This again may be due to the fact that BY adjusts for dependence structures. BH and Q-Value methods behave similarly when applied to the p-values obtained by CNT. The variance of type I errors behave similarly to the variance of the false discovery proportions. However, these variances vary in the range of 10^{-4} and will not be illustrated here.

9.5 Type II Error Proportion

Similar to above cases, the type II errors can be computed for simulated data. Since there was a reduction in false discovery proportion and type I errors by using CNT over the t-test, it can be expected that CNT makes more type II errors compared to the t-tests. This is illustrated in the figure below.

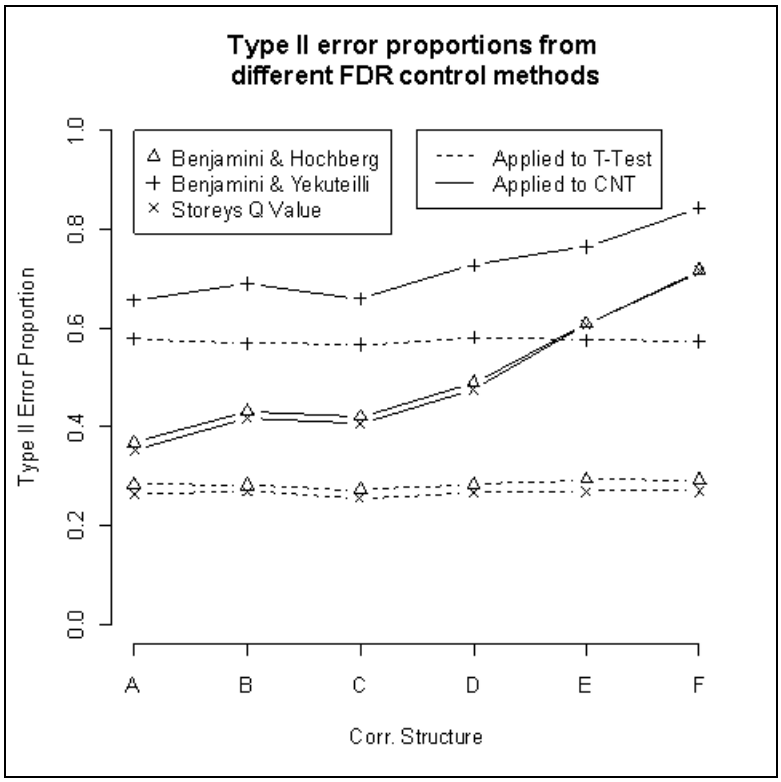

Figure 9.6: Comparison of type II error proportions.
Average type II proportions comparison for FDR control methods between CNT and t-tests. The figure shows application of FDR control methods (FDR controlled at 0.05) to t-tests produces a lower type II error proportion compared to CNT.

The plot above illustrates the behavior of type II errors under different dependencies. Although the type II error proportions are better (lower) for t-tests than CNT, the variance of type II error proportion is generally smaller when FDR methods are applied to CNT, compared to the t-test. The figure below illustrates the variance of the type II errors under these control methods. The variance is quite similar for the three FDR control methods for both t-tests and CNT under weak dependencies. However, the variance is much lower for CNT compared to t-tests at stronger dependencies. The BY method shows the highest variance at all dependence structures considered.

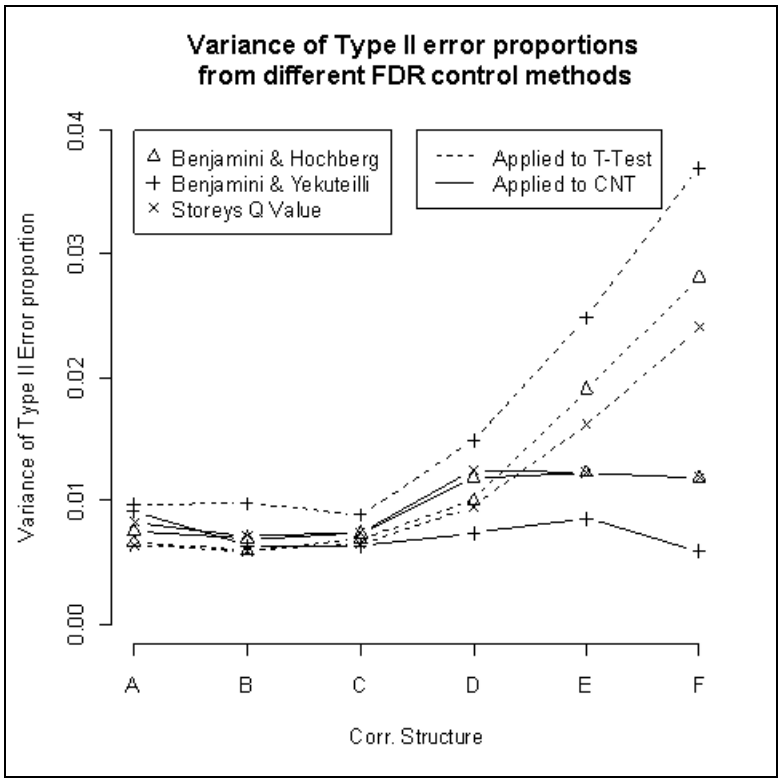

Figure 9.7: Comparison of variance of type II error proportions.
Variance of type II error proportions under different FDR control methods (Controlling at 0.05 level). While there is not much difference in the variance at weaker dependencies, CNT has a smaller variance compared to the t-test at strong dependencies.

CNT provides a different set of p-values that can be used in FDR control methods. This chapter compared the application of FDR control methods for the p-values obtained by t-tests and adjusted p-values obtained by CNT. The true false discovery proportions and type I error proportions are lower when FDR control methods are applied to p-values obtained by CNT compared to the p-values for t-tests. There is an increased type II error proportion when these methods are applied to CNT, which can be expected because false discovery proportions and type I error proportions are reduced. BH and Q-value methods produce similar results when applied to CNT p-values for data simulated here. On average, the false discovery proportion and type I error proportions for BY method applied to t-tests and CNT behave relatively similarly. It can be suspected that this is due to the reason that BY provides some correction for the dependence structure. It was illustrated that these FDR control methods produce stable results (i.e. less variance) when using p-values from CNT compared to using p-values from regular t-tests.

9.6 Plasmode Simulation Results

Above simulations were performed using systematic dependence structures. The simulation described below uses plasmode data (using lung cancer and multiple myeloma data as templates) to evaluate false discovery, type I and type II error proportions. The dependence structures for these data are more closer to the dependence structures of the real life data.

Simulation 16

Plasmode data were generated using lung cancer data and multiple myeloma data with known mean structures (10% of the features different between the groups). Both CNT and t-tests were performed on these data and FDR control methods BH, BY and Storey's Q-value methods were used on the p-values obtained from these two tests. Since these methods are more conservative than the t-test, initial test and FDR control was done at 0.05 level. False discovery proportions from BH, BY and Q-Value methods are shown in the Figure 9.8 below. This figure illustrates that the false discovery proportions and their standard deviations are reduced when FDR control methods are applied to the p-values obtained by CNT compared applying FDR control methods to the p-values obtained by t-tests. Among the considered methods, Q value method is the least conservative. It estimates the proportion of null hypotheses for a given dataset. BH method is comparatively more conservative since it operates under the assumption of true global null hypothesis. The BY procedure is the most conservative since it adjusts for the dependence structure.

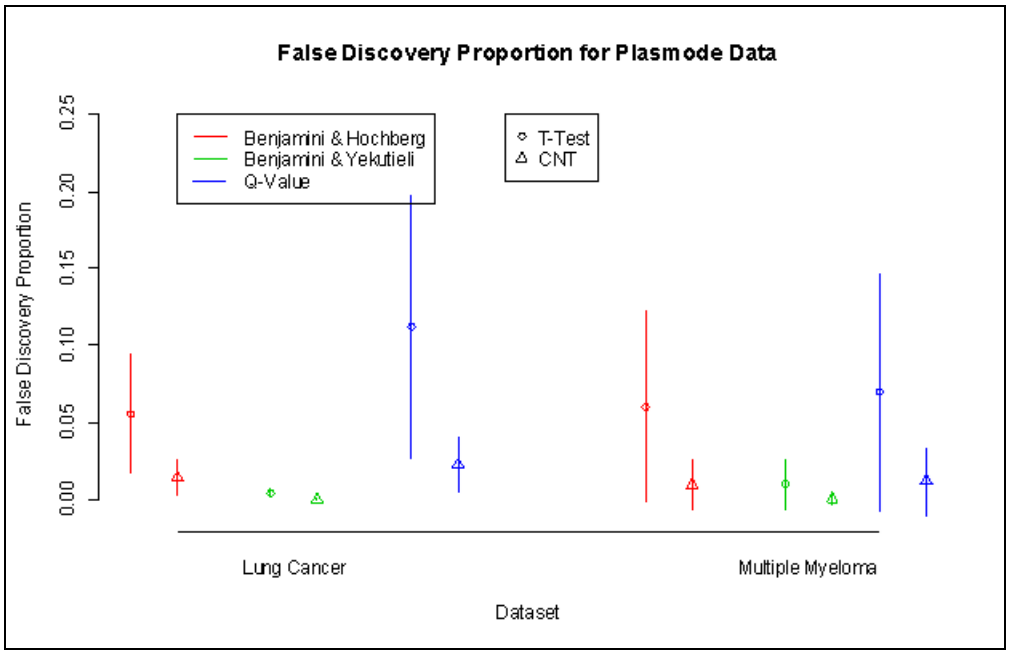

Figure 9.8: Comparison of false discovery proportions.
Comparison of true false discovery proportions between t-tests and CNT when FDR control methods are applied. The plot shows that the mean and standard deviation of false discovery proportions are smaller when applied to the p-values obtained by CNT. Percentage reductions are listed in Table **9.1** below. FDR control level is set to 0.05.

The percentage reductions in average false discovery proportions and their standard deviations are listed in Table 9.1 below.

Table 9.1: Percentage Reduction in False Discovery Proportions from using CNT vs. t-tests

	Lung Cancer Data		Multiple Myeloma	
	Mean	Std. Dev.	Mean	Std. Dev.
BH	73%	69%	83%	73%
BY	81%	73%	88%	76%
QV	79%	78%	81%	71%

The similarity between BY and Q value methods is also seen in the percentage reduction in average false discovery proportions and its standard deviation.

Figure 9.9 below illustrates the behavior of type I error proportions that resulted by using FDR control methods. The figure is almost identical to the figure above, for the proportion of false discoveries.

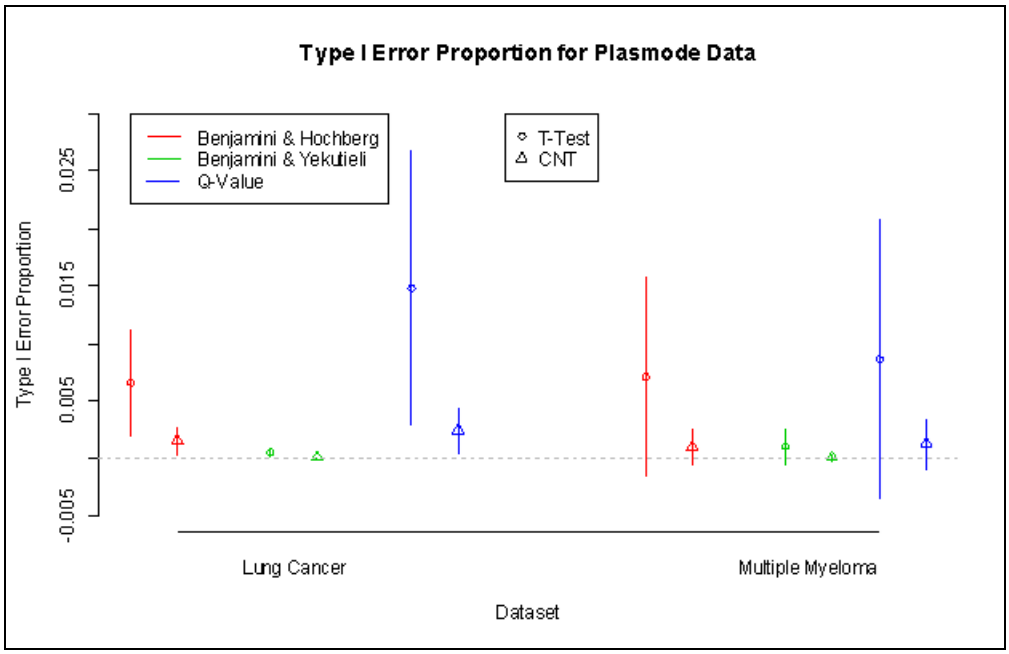

Figure 9.9: Comparison of type I error proportions.
In general, the type I error proportions and its standard deviations are smaller for CNT compared to t-tests. Percentage reductions are given in the table below. FDR control level was set to 0.05.

The table below lists the percentage reduction in type I errors.

Table 9.2: Percentage Reduction in type I error proportions from using CNT versus t-tests

	Lung Cancer Data		Multiple Myeloma	
	Mean	Std. Dev.	Mean	Std. Dev.
BH	76%	73%	86%	81%
BY	82%	75%	89%	79%
QV	83%	83%	85%	81%

Above plots and tables illustrates that the false discovery and type I error proportions are reduced when FDR method applied to CNT results, compared to t-test results. The figure below illustrates the behavior of type II error proportions.

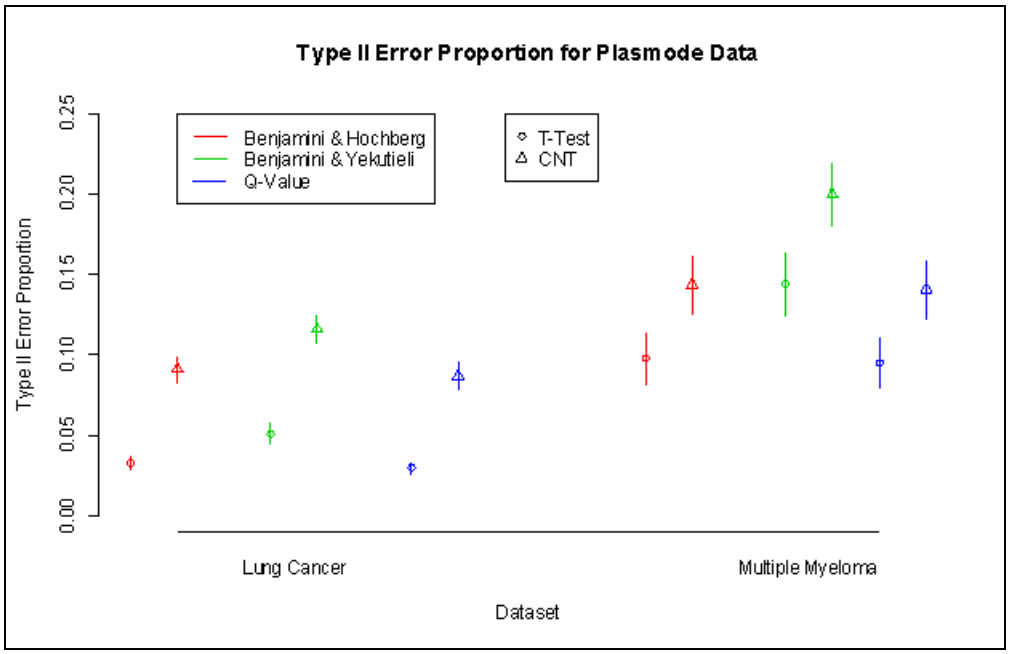

Figure 9.10: Comparison of Type II error proportions.
Proportions of type II errors that resulted by the application of FDR control methods to the P-values obtained by CNT and t-tests. Figure illustrates the increment of type II error proportion and the table below quantifies the percentage increment of type II error.

The figure above illustrates the increment of type II error proportions by all FDR control methods. This can be expected since both type I error and false discovery proportions were decreased. The percentage increment of type II error proportion in CNT compared to t-tests are listed in the following table.

Table 9.3: Percentage increment in type II error proportions

	Lung Cancer Data		Multiple Myeloma	
	Mean	Std. Dev.	Mean	Std. Dev.
BH	175%	73%	47%	14%
BY	126%	22%	38%	2%
QV	189%	119%	48%	16%

In addition to the average type II error proportion, the standard deviation of the type II errors proportions is also increased.

This chapter compared the results of applying FDR control methods to the set of p-values obtained by t-tests and the set of p values obtained by CNT. Results showed that the type I error and false discovery proportions are less when FDR methods are applied to CNT. However, this gain in false discovery proportion is at the expense of increased type II errors. It might be noted in the plasmode results that the type I errors were quite small after adjusting for FDR = 0.05. Type II errors are relatively large. Controlling FDR at level, say 0.1, would make these two error levels less extreme. Above illustrations imply that CNT can be effectively used in experiments where the emphasis is on the set of features declared as significant. FDR control methods can be employed to further reduce the proportion of false discoveries made by CNT. The following chapter discusses additional ideas that can be used to further improve the CNT method

.

Chapter 10 - Concluding Remarks and Future of Conditional Network Testing

The problem of large scale multiple testing was introduced in Chapter 1 and many challenges when analyzing such data were discussed. One challenge addressed by many authors concerns the behavior of the test statistics or the distribution of the p-values in large scale multiple testing under dependencies. It was shown that these dependencies cause the tests to be unstable and the variance of the number of discoveries (Hu et al, 2010) and the variance of the false discovery rates (Efron, 2009) behave differently than under independence. This work explored the effect of dependencies on the variance of the number of discoveries and proposed a method, *Conditional Network Testing* to control that variance. In addition, the behavior of false discovery rate control methods applied to conditional network testing was compared with their application to the regular t-tests. Two real world datasets, lung cancer and multiple myeloma datasets, were used to illustrate the results. Chapter 2 discussed some of the work that is already done addressing these issues in multiple testing. While some of these methods make strong assumptions about the data, the others relax some of the assumptions at the expense of simplicity in algorithms, theory and implementations. Chapter 2 also introduced some of the theory and concepts that were the intuition behind the development of conditional network testing.

Chapter 3 illustrated the behavior of the variance of the number of discoveries under dependencies. This showed that the variance increases with the increasing dependence structure and a closed form expression for the variance of the number of discoveries was derived which is an extension of the variance given in Hu et al (2010). Chapter 4 derived closed forms of the conditional densities of test statistics and the correlation coefficient between test statistics. This chapter also explored the behavior of this conditional density under different conditions and different dependence structures. Chapter 5 explored the behavior of correlations between test statistics and derived its sampling distribution. Chapter 6 introduced the concepts behind the proposed method and discussed conditional testing within networks in detail. The intuition behind conditional network testing is to adjust the p-values obtained by t-tests by considering the correlations among the features being tested. In addition this chapter showed that the use of conditional network testing resulted in a smaller variance of the number of discoveries compared

to the t-tests. A procedure to visualize the networks of features constructed from data was also shown.

Chapter 7 introduced a new method to simulate data with an unknown dependence structure but a known mean structure. This is an extension to the method suggested by Gadbury et al (2008). This chapter also discussed some of the issues that arise when using this method on different datasets. Chapter 8 further explored the effects of conditional network testing in terms of type I errors, type II errors, and false discovery proportions and compared them with the t-tests. This showed that conditional network testing reduces the type I error rate and false discovery proportion while reducing the variance of the number of discoveries, but type II error rates are increased. Chapter 9 discussed the application of FDR control methods to the p-values obtained from conditional network testing, and then compared them with the t-test results. Similar to chapter 8 results, all FDR control methods resulted in lower type I error rates and false discovery proportions, but higher type II error rates when applied to conditional network testing. CNT resulted in a smaller variance for the false discovery and type I error proportions, but a slightly higher variance for the type II error proportion.

Simulations were used to illustrate the characteristics of Conditional Network Testing and to compare the results with those obtained using regular t-tests. They showed that the variance of the number of discoveries is reduced by Conditional Network Testing at all dependence structures. While statistical methods can be evaluated using simulation by generating datasets repeatedly, their applicability in real life applications must also be assessed. In real applications only one dataset is collected. The high variance in the results using the t-test means that the number of discoveries made by the test or the actual discoveries themselves may not be reliable. The only way to corroborate the results is to repeat the experiment, which is not an option in most of the real world studies (i.e. due to financial and time limitations). Since the variance of the number of discoveries from Conditional Network Testing is less compared to the t-tests, results obtained using CNT are more reliable (repeated tests would tend to yield similar results).

In addition, it was shown that the false discovery proportions resulting from Conditional Network Testing is lower than the false discovery proportions resulting from using t-tests. FDR

control methods are often used to control the number of false findings in large scale testing. Many of these can be applied to either test statistics values or p-values. It was shown that application of these methods to the p-values obtained by Conditional Network Testing further reduces the false discovery proportion. Not only the false discovery proportions are less, their variances are also reduced compared to the application of these methods to p-values obtained from usual t-tests. Therefore, Conditional Network Testing provides a more reliable way to compare two groups in large scale simultaneous testing.

While there are reductions in false discovery proportions and type I error rates, there is an increment in type II error rates. Conditional network testing misses some of the true positives, especially when there are very strong dependencies. Simulations showed that the type I error rate and type II error rates stay constant at certain levels under the t-test. Any correction that reduces the type I error rate will likely increase the type II error rate. However, the variance of the type II error rates is also increased for Conditional Network Testing at higher dependence structures. Another drawback of Conditional Network Testing is its high demand of computational power. For a large dataset, Conditional Network Testing requires a large amount of memory to keep all pairwise correlations stored. In addition, the procedure of cycling through all features takes a long computational time. However, the conventional desktop or laptop computers today have enough power to run these analyses. The simulations and analyses presented in this dissertation were run in a UNIX cluster which has 1,200 cores and over 640 gigabytes of memory.

Conditional Network Testing procedure can be further improved to stabilize and speed up the process. Since networking is done separately for features declared statistically significant and declared not significant, they can be run in parallel to reduce the execution time. Since the testing of networks is done independently from each other, this process can also be run in parallel computers/cores which will further improve the analysis time. At its current stage, the conditional testing is done only conditioned on the most closely related feature. The testing process may be further improved by conditioning on all of the features in the branch (e.g. Condition J on F, C and A in figure Figure 6.2.).

The results shown in this work illustrated that Conditional Network Testing controls the variance of the number of discoveries and produces stable false discovery and type I error

proportions. Conditional Network Testing also allows FDR control methods to be used on the set of resulting p values which further reduces the proportion of false discoveries. Conditional Network Testing has the potential to be a competitive option in the realm of high dimensional data analysis.

References

Allison, D. B., Cui, X., Page, G. P., & Sabripour, M. (2006). Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews*, 55-65.

Allison, D. B., Gadbury, G. L., Heo, M., Fernández, J. R., Lee, C.-K., Prolla, T. A., et al. (2002). A Mixture Model Approach for the Analysis of Microarray Gene Expression Data. *Computational Statistics & Data Analysis*.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, M. J., et al. (2000). Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25-29.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the Flase Discovery Rate: a Practical and Powerful Approach to Multiple Testing. *J. R. Statist. Soc. B*, 289-300.

Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 1165-1188.

Berger, J. O., & Selke, T. (1987). Testing a point null hypothesis: The irreconcilability of P values and evidence. *Journal of American Statistical Association*, 112-122.

Böhning, D., Hempfling, A., Schelp, F. P., & Schlattmann, P. (1992). The area between curves (ABC) measure in nutritional anthropometry. *Statistics in Medicine*, 1289–1304.

Bolstad, B. M., Irizarry, R. A., Astrand, M., & Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 185-193.

Bradley, E. L. (1985). Overlapping coefficient. *Encyclopedia of Statistical Sciences*, 546–547.

Breiman, L. (2001). *Machine Learning.* Springer.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classificaiton and Regression Trees.* Wadsworth, Belmont: Chapman & Hall/CRC.

Cope, L. M., Irizarry, R. A., Wu, Z., & Speed, T. P. (2004). A benchmark for Affymetrix GeneChip expression measures. *Bioinformatics*, 323-331.

Donoho, D. L. (2000). High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. *AMS Math Challenges Lecture*.

Efron, B. (2004). Large-scale simultaneous hypothesis testing: The choice of a null hypothesis. *JASA*, 96-104.

Efron, B. (2007). Correlation and large-scale simultaneous significance testing. *JASA*, 93-103.

Efron, B. (2010). Correlated z-values and the accuracy of large-scale statistical estimates (with discussion and Rejoinder). *JASA*, 1042-1069.

Gadbury, G. L., Page, G. P., Heo, M., Mountz, J. D., & Allison, D. B. (2003). Randomization tests for small samples: an application for genetic expression data. *Appl. Statist.*, 365-376.

Gadbury, G. L., Xiang, Q., Yang, L., Barnes, S., Page, G. P., & Allison, D. B. (2008). Evaluating Statistical Methods Using Plasmode Data Sets in the Age of Massive Public Databases: An Illustration Using False Discovery Rates. *PLos Genetics*.

Garrett, K. A., Dendy, S. P., Frank, E. E., Rouse, M. N., & Travers, S. E. (2006). Climate change effects on plant disease: Genomes to ecosystems. *Annual Review of Phytopathology*, 489-509.

Hochberg, Y., & Tamhane, A. C. (1987). *Multiple Comparison Procedures.* New York: John Wiley & Sons.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 65-70.

Hu, X., Gadbury, G. L., Xiang, Q., & Allison, D. B. (2010). Illustrations on using the distribution of a p-value in high dimensional data anlaysis. *Advances and Applications in Statistical Sciences*, 191-213.

Ihmels, J., Bergmann, S., Berman, J., & Barkai, N. (2005). Comparative gene expression analysis by a differential clustering approach: Application to the Candida albicans transcription program. *PLoS Genetics*.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., et al. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 249-64.

Irizarry, R. A., Ladd-Acosta, C., Carvalho, B., Wu, H., Brandenburg, S. A., Jeddeloh, J. A., et al. (2008). Comprehensive high-throughput arrays for relative methylation (CHARM). *Genome Research*, 780-790.

Johnstone, I. M., & Titterington, D. M. (2009). Statistical challenges of high-dimensional data. *Phil. Trans. R. Soc. A*, 4237-4253.

Kanehisa, M., & Susumu, G. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucl. Acid Res.*, 27-30.

Katajamaa, M., Miettinen, J., & Orešič, M. (2006). MZmine: toolbox for processing and visualization of mass spectrometry based molecular profile data. *Bioinformatics*, 634-636.

Kim, K. I., & Wiel, M. A. (2008). Effects of dependence in high-dimensional multiple testing problems. *BMC Bioinformatics*, 114.

Klebanov, L., & Yakovlev, A. (2007). How high is the level of technical noise in microarray data? *Biology Direct*.

Knudsen, S. (2002). *A Biologists guide to Analysis of DNA Microarray Data.* New York, NY: John Wiley & Sons, Inc.

Kyung, I. K., & Wiel, M. (2008). Effects of dependence in high-dimensional multiple testing problems. *Bioinformatics*.

McLachlan, G. J., Do, K.-A., & Ambroise, C. (2004). *Analyzing Microarray gene expression data.* Hoboken, NJ: John Wiley & Sons, Inc.

Mehta, T., Tanik, M., & Allison, D. (2004). Towards sound epistemological foundations of statistical methods for high-dimensional biology. *Nature Genetics*, 943 - 947.

Owen, A. B. (2005). Variance of the number of false discoveries. *J. R. Statist. Soc. B*, 411-426.

Pepe, M. S., Longton, G., Anderson, G. L., & Schummer, M. (2003). Selecting Differentially Expressed Gene from Microarray Experiments. *Biometrics*, 133-142.

Ritchie, M. E., Silver, J., Oshlack, A., Holmes, M., Diyagama, D., Holloway, A., et al. (2007). A comparison of background correction methods for two-color microarrays. *Bioinformatics*, 2700-2707.

Ritchie, M. E., Silver, J., Oshlack, A., Holmes, M., Diyagama, D., Holloway, A., et al. (2007). A comparison of background correction methods for two-colour microarrays. *Bioinformatics*, 2700-2707.

Sauve, A. C., & Speed, T. P. (2004). Normalization, Baseline Correction and Alignment of High-Throughput mass spectrometry data. *In press*.

Schilling, C. H. (1999). *Metabolic Pathway Analysis: Basic Concepts and Scientific Applications in the Post-genomic Era.* Biotechnology Press.

Schott, J. R. (2007). A test for the equality of covariance matrices when the dimension is large relative to the sample sizes. *Computational Statistics and Data Analysis*, 6535-6542.

Schweder, T., & Spjøtvoll , E. (1982). Plots of P-values to evaluate many tests simultaneously. *Biometrica*, 493-502.

Silverstein, R. M., & Webster, F. X. (2002). *Spectrometric Identification of Organic Compounds.* New Delhi: John Wiley & Sons.

Smyth, G. K. (2004). Linear Models and Empirical Bayes Methods for Assessing Differencial expression in Microarray experiments. *Statistical Applications in Genetics and Molecular Biology*.

Southworth, L. K., Owen, A. B., & Stuart, K. K. (2009). Aging Mice Show a Decreasing Correlation of Gene Expression within Genetic Modules. *PLoS Genetics*.

Spira, A., Beane, J. E., Shah, V., Steiling, K., Liu, G., Schembri, F., et al. (2007). Airway epithelial gene expression in the diagnostic evaluation of smokers with suspect lung cancer. *Nature Medicine*, 361-366.

Srivastava, M. S., & Yanagihara, H. (2010). Testing the equality of several covariance matrices with fewer observations than the dimension. *Elsevier*, 1319-1329.

Stine, R. A., & Heyse, J. F. (2001). Non-parametric estimates of overlap. *Stat Med*, 215-36.

Storey, J. D. (2002). A direct approach to false discovery rates. *J. R. Statist. Soc. B*, 479-498.

Storey, J. D., Tibshirani, R., & Efron, B. (2001). Microarrays empirical Bayes methods, and false discovery rates. *Dept. of Statistics Technical Report*, 217.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *PANS*, 15545-15550.

Tanaka, T. S., Jaradat, S. A., Lim, M. K., Kargul, G. J., Wang, X., Grahovac, M. J., et al. (2000). Genome-wide expression profiling of mid-gestation placenta and embryo using a 15,000 mouse developmental cDNA microarray. *PNAS*, 9127–9132.

Team, R. D. (2011). R: A Language and Environment for Statistical Computing.

Thimmulappa, R. k., Mai, K. H., Srisuma , S., Kensler , T. W., Yamamoto , M., & Biswal, S. (2002). Identification of Nrf2-regulated genes induced by the chemopreventive agent sulforaphane by oligonucleotide microarray. *Cancer research* , 5196-5203.

Tian, E., Zhan, F., Walker, R., Rasmussen, E., Ma, Y., Barlogie, B., et al. (2003). The role of the Wnt-signaling antagonist DKK1 in the development of osteolytic lesions in multiple myeloma. *The New Englan Journal of Medicine*, 2483-2494.

Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance Analysis of microarrays applied to the ionizing radiation response. *PNAS*, 5116-5121.

Westfall, P., & Young, S. (1993). *Resampling-based Multiple Testing: Examples and methods for p-value adjustment.* New York: John Wiley & Sons.

Appendix A: R Programs

```
#########################################################################
##
##   Function for performing row-wise t-tests for a given matrix
##   Grouping variable must be provided separately and is used to group
##   columns in the data matrix
##   Input:
##        Data matrix – subjects in columns and variables in rows
##        Grouping – grouping variable used to group subjects
##   Output:
##        Matrix with columns t-statistic values, p-values and degrees of
##        freedom
##
#########################################################################

rowttest<-function(data1, groups) {

 dottest<-function(d, g) {
   x<-d[g==unique(g)[1]]
   y<-d[g==unique(g)[2]]
   t<-t.test(x, y, alternative="two.sided")
   c(t.stat=t$statistic, p.val=t$p.value, df=t$parameter)
 }

 t(apply(X=data1, MARGIN=1, FUN=dottest, g=groups))
}
```

```
#########################################################################
##
##   Function for performing row-wise t-tests and transform the p-values
##   to obtain normal z values using 2.10 for a given matrix
##   Grouping variable must be provided separately and is used to group
##   columns in the data matrix
##   Input:
##        Data matrix – subjects in columns and variables in rows
##        Grouping – grouping variable used to group subjects
##   Output:
##        Matrix with columns transformed test statistic values,
##        p-value and degrees of freedom (for the t-statistic)
##
#########################################################################

rowttest2<-function(data1, groups) {

 dottest<-function(d, g) {
   x<-d[g==unique(g)[1]]
   y<-d[g==unique(g)[2]]
   t<-t.test(x, y, alternative="two.sided")
   c(t.stat=t$statistic, p.val=t$p.value, df=t$parameter)
 }

 tt<-t(apply(X=data1, MARGIN=1, FUN=dottest, g=groups))
 tt[,1]<-sign(tt[,1])*qnorm(tt[,2]/2, lower.tail=F)
 tt
}
```

```
#########################################################################
##
##   Function to generate block diagonal correlation matrices with
##   given correlations and block sizes
##
##   Input:
##     size of the matrix, size of blocks, correlation value in blocks
##
##   Output:
##     size x size block diagonal matrix with correlation values in
##     diagonal blocks
##
#########################################################################

makecormat<-function(tsize, bsize, rho) {
 if (tsize%%bsize==0) {
   out.mat<-matrix(0, nrow=tsize, ncol=tsize)

   b.mat<-matrix(rho, nrow=bsize, ncol=bsize)
   diag(b.mat)<-rep(1, bsize)

   for ( i in 1:(tsize/bsize) ) {
     out.mat[((i-1)*bsize+1):((i-1)*bsize+bsize),
             ((i-1)*bsize+1):((i-1)*bsize+bsize)] <-b.mat
   }
 }
 out.mat
}
```

```
#########################################################################
##
##   my.image.plot.r
##
##   Alternative plot for image.plot(fields)
##   Produces multiple image plots on the same plot with consistent
##   range across all plots
##   Used to produce figure 3.1
##   inputs: z: matrix of color values for the plot
##           Other plot parameters
##   Output: Produces the plot
##
#########################################################################


my.image.plot<-function(z, plot.main="", x.lab="", y.lab="", plot.sub="") {

   plot(NULL, xlim=c(0, ncol(z)), ylim=c(0, nrow(z)), axes=F,
        main=plot.main, xlab=x.lab, ylab=y.lab, sub=plot.sub)
     for(i in 0:(ncol(z)-1)) {
     for(j in 0:(nrow(z)-1)) {
        rect(i, j, i+1, j+1, col=z[i+1,j+1], border=0)
     }
   }

   rect(0, 0, nrow(z), ncol(z))

   text(1:10-0.5, par("usr")[3], srt=90, adj=1, labels=colnames(z), xpd=T, cex=0.8)
   text(par("usr")[1], 1:10-0.5, srt=0, adj=1, labels=colnames(z), xpd=T, cex=0.8)
}
```

```
#########################################################################
##
##   blockplots.r (produces figure 3.1)
##
##   Reads simulated data and plots the variance of the
##   number of discoveries in an image.plot style figure.
##   The variances are converted to color values to pass
##   to the my.image.plot function. Multiple plots are
##   drawn separately on the same canvas to the same scale.
##   This program uses previously saved simulation data.
##
#########################################################################

b1<-unique(d[,3])
b2<-unique(d[,4])
r1<-unique(d[,1])
r2<-unique(d[,2])

varout<-NULL

for(ba in b1) {
  for(bb in b2) {
    dx<-d[d[,3]==ba & d[,4]==bb,]
    x<-sort(unique(d[,1]))
    y<-sort(unique(d[,2]))
    z<-matrix(0, nrow=length(x), ncol=length(y))
    colnames(z)<-x
    rownames(z)<-y
    dy<-cbind(dx[,1:4], vs=apply(X=dx[,105:204], MARGIN=1, FUN=var))
    varout<-rbind(varout, dy)
  }
}

varout<-cbind(index=1:nrow(varout), varout)

varout<-varout[order(varout[,6]),]
varout<-cbind(varout, cols=rep(tim.colors(160), rep(10, 160)))
varout<-varout[order(varout[,1]),]


par(mfrow=c(length(b1), length(b2)))

for(i in 0:15) {
   vs<-varout[(i*100+1):((i+1)*100),]
   matcol<-matrix(vs[,7], ncol=10)
   colnames(matcol)<-t(unique(vs[2]))
   rownames(matcol)<-t(unique(vs[3]))
   my.image.plot(matcol, y.lab=paste("b size",vs[1,4]),
                 x.lab=paste("b size",vs[1,5]),
                 plot.main=paste(vs[1,4], "x", vs[1,5]))
}

x11(150, 30)
vs<-tim.colors(160)
plot(NULL, xlim=c(0,length(unique(vs))), ylim=c(0, 1), axes=F,
     xlab="", ylab="")
for(i in 0:(length(unique(vs))-1)) {
   rect(i, 0, i+1, 1, col=vs[i+1], border=F)
}

vs2<-varout[order(varout[,6]),6]
x.lab<-NULL
for(i in 1:10) {
```

```
  x.lab<-c(x.lab, vs2[i*160])
}

indx<-1:10*16
text(indx-10, par("usr")[3]-0.5, srt=90, adj=1, labels=round(x.lab, 2), xpd=T,
cex=0.8)
```

```
#####################################################################
##
##  Probability Density, cumulative density and quantile functions
##  of the conditional density given in 4.11
##  the null hypothesis
##  Inputs:
##            Value of x
##            t: t-test value
##            rho: combined correlation between test statistics
##            p: probability (for quantiles)
##
##  d.cond.dens: pdf of conditional density conditioned on rejecting H0
##  p.cond.dens: cumulative conditional density conditioned on  rejecting
##  q.cond.dens: quantile function of conditional density conditioned on
##        not rejecting H0 for the jth test
##
#####################################################################


d.cond.dens<-function(x, t, rho) {
    dnorm(x)*(1+pnorm((-t-rho*x)/sqrt(1-rho^2))-pnorm((t-rho*x)/sqrt(1-
rho^2)))/(2*pnorm(-t))
}

p.cond.dens<-function(x, r, t, lower.tail=T) {
  if ( length(x) > 1 ) {
    if ( length(r) == 1 ) r<-rep(r, length(x))
    if ( length(r) > 1 & length(r)!= length(x) ) stop("r and x must of same length")

    if ( length(t) == 1 ) t<-rep(t, length(x))
    if ( length(t) > 1 & length(t)!= length(x) ) stop("t and x must of same length")
  }

  d<-cbind(x, r, t)

  d.cond.dens<-function(x, t, rho) {
    dnorm(x)*(1+pnorm((-t-rho*x)/sqrt(1-rho^2))-pnorm((t-rho*x)/sqrt(1-
rho^2)))/(2*pnorm(-t))
  }

  af<-function(v) {
    integrate(f=d.cond.dens, lower=-Inf, upper=v[1], rho=v[2], t=v[3])$value
  }

  val<-apply(X=d, MARGIN=1, FUN=af)
  if( !lower.tail ) val<-1-val
  val
}


q.cond.dens<-function(p, r, lower.tail=T, interv=c(-4, 4)) {

  q.cond.dens.2<-function(p, t, r, lower.tail=T, interv=c(-4, 4)) {
     af<-function(x, p, t, r, lt=lower.tail){
```

```
        abs(p-p.cond.dens(x, r, t, lower.tail=lt))
      }
      optimize(f=af, interval=interv, r=r, t=t, p=p)$minimum
  }
  tx<-qnorm(p, lower.tail=lower.tail)
  q.cond.dens.2(p, tx, r, lower.tail, interv=interv)
}


#####################################################################
##
##  Probability Density, cumulative density and quantile functions
##  of the conditional density given in 4.14
##  Inputs:
##            Value of x
##            t: t-test value
##            rho: combined correlation between test statistics
##            p: probability
##
##  d.cond.dens2: pdf of conditional density conditioned on not rejecting H0
##  p.cond.dens2: cumulative conditional density conditioned on not rejecting
##  q.cond.dens2: quantile function of conditional density conditioned on
##        not rejecting H0 for the jth test
##
#####################################################################



d.cond.dens2<-function(x, t, rho) {
   dnorm(x)*(pnorm((t-rho*x)/sqrt(1-rho^2))-pnorm((-t-rho*x)/sqrt(1-
rho^2)))/(pnorm(t)-pnorm(-t))
}

p.cond.dens2<-function(x, r, t, lower.tail=T) {
  if ( length(x) > 1 ) {
    if ( length(r) == 1 ) r<-rep(r, length(x))
    if ( length(r) > 1 & length(r)!= length(x) ) stop("r and x must of same length")

    if ( length(t) == 1 ) t<-rep(t, length(x))
    if ( length(t) > 1 & length(t)!= length(x) ) stop("t and x must of same length")
  }

  d<-cbind(x, r, t)

   d.cond.dens2<-function(x, t, rho) {
    dnorm(x)*(pnorm((t-rho*x)/sqrt(1-rho^2))-pnorm((-t-rho*x)/sqrt(1-
rho^2)))/(pnorm(t)-pnorm(-t))
  }

  af<-function(v) {
    integrate(f=d.cond.dens2, lower=-Inf, upper=v[1], rho=v[2], t=v[3])$value
  }

  val<-apply(X=d, MARGIN=1, FUN=af)
  if( !lower.tail ) val<-1-val
  val
}

q.cond.dens2<-function(p, r, lower.tail=T, interv=c(-4, 4)) {
   q.cond.dens2.2<-function(p, t, r, lower.tail=T, interv=c(-4, 4)) {
       af<-function(x, p, t, r, lt=lower.tail){
         abs(p-p.cond.dens2(x, r, t, lower.tail=lt))
       }
```

```
    optimize(f=af, interval=interv, r=r, t=t, p=p)$minimum
   }
   tx<-qnorm(p, lower.tail=lower.tail)
   q.cond.dens2.2(p, tx, r, lower.tail, interv=interv)
}
```

```
##############################################################################
##
##   Function for computing density of sample correlation coefficient
##   under independence (Stuart and Ord, 2009)given in 2.18
##   input: x (value of correlation)
##          n sample size
##
##############################################################################

dens.f<-function(x, n) {
 1/beta(0.5, 0.5*(n-2))*(1-x^2)^(0.5*(n-4))
}
```

```
##############################################################################
##
##   Program that creates the 3D scatterplot for the intermediate joint density
##   of correlations given in 5.1.
##
##############################################################################

par(mfrow=c(1, 2))
n1<-40
n2<-40

sd11<-3
sd12<-1
sd21<-2
sd22<-1

feats<-500*499/2

delta<-0.01

a<-(sd11*sd12)/n1/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
b<-(sd21*sd22)/n2/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))

#a<-(1/n1)/(1/n1+1/n2)
#b<-(1/n2)/(1/n1+1/n2)

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

fn<-function(x, y, n1, n2) {
   c1<-(sd11*sd12)/n1/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
   d1<-(sd21*sd22)/n2/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
   1 / (2*c1*d1*beta(0.5, 0.5*(n1-2))*beta(0.5, 0.5*(n2-2))) * (1 -
((x+y)/(2*c1))^2)^(0.5*(n1-4)) * (1 - ((x-y)/(2*d1))^2)^(0.5*(n2-4))
}

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

x1<- -(a+b)
```

```
y1<- b-a

x2<- b-a
y2<- -(a+b)

x3<- a+b
y3<- a-b

x4<- a-b
y4<- a+b

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

x<-seq(-(a+b)+delta/2, (a+b)-delta/2, by=delta)

out<-NULL

for(i in x) {
  ll<- y1 - (y1-y2)/(x1-x2)*x1 + (y1-y2)/(x1-x2)*i
  if ( abs(ll) > 1 ) {
    ll<- y2 - (y2-y3)/(x2-x3)*x2 + (y2-y3)/(x2-x3)*i
  }

  ul<- y1 - (y1-y4)/(x1-x4)*x1 + (y1-y4)/(x1-x4)*i
  if ( abs(ul) > 1 ) {
    ul<- y4 - (y4-y3)/(x4-x3)*x4 + (y4-y3)/(x4-x3)*i
  }

  y<-seq(ll, ul, by=delta)

  for(j in y) {
    out<-rbind(out, c(i, j, fn(i, j, n1, n2)))
  }
}

colnames(out)<-c("X", "Y", "Z")
scatterplot3d(out, pch=".", xlab="x", ylab="y", zlab=expression(f[X*Y]*"(x, y)"),
box=F, cex.axis=0.7,
main=expression("Joint density "*f[X*Y]*"(x, y; a, b)"))
title(sub=expression(n[i]*"=40  "*n[j]*"=40  "*sigma[i]^'*'*"=1   "*sigma[j]^'*'*"=1
"*tau[i]^'*'*"=1   "*tau[j]^'*'*"=1"*""))

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


n1<-80
n2<-40

sd11<-3
sd12<-2
sd21<-1
sd22<-1

feats<-500*499/2

delta<-0.01

a<-(sd11*sd12)/n1/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
b<-(sd21*sd22)/n2/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))

#a<-(1/n1)/(1/n1+1/n2)
#b<-(1/n2)/(1/n1+1/n2)
```

```
#####################################################

fn<-function(x, y, n1, n2) {
   c1<-(sd11*sd12)/n1/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
   d1<-(sd21*sd22)/n2/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
   1 / (2*c1*d1*beta(0.5, 0.5*(n1-2))*beta(0.5, 0.5*(n2-2))) * (1 -
((x+y)/(2*c1))^2)^(0.5*(n1-4)) * (1 - ((x-y)/(2*d1))^2)^(0.5*(n2-4))
}

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

x1<- -(a+b)
y1<- b-a

x2<- b-a
y2<- -(a+b)

x3<- a+b
y3<- a-b

x4<- a-b
y4<- a+b

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

x<-seq(-(a+b)+delta/2, (a+b)-delta/2, by=delta)

out<-NULL

for(i in x) {
  ll<- y1 - (y1-y2)/(x1-x2)*x1 + (y1-y2)/(x1-x2)*i
  if ( abs(ll) > 1 ) {
    ll<- y2 - (y2-y3)/(x2-x3)*x2 + (y2-y3)/(x2-x3)*i
  }

  ul<- y1 - (y1-y4)/(x1-x4)*x1 + (y1-y4)/(x1-x4)*i
  if ( abs(ul) > 1 ) {
    ul<- y4 - (y4-y3)/(x4-x3)*x4 + (y4-y3)/(x4-x3)*i
  }

  y<-seq(ll, ul, by=delta)

  for(j in y) {
    out<-rbind(out, c(i, j, fn(i, j, n1, n2)))
  }
}

colnames(out)<-c("X", "Y", "Z")
scatterplot3d(out, pch=".", xlab="x", ylab="y", zlab=expression(f[X*Y]*"(x, y)"),
box=F, cex.axis=0.7,
main=expression("Joint density "*f[X*Y]*"(x, y; a, b)"))
title(sub=expression(n[i]*"=80  "*n[j]*"=40  "*sigma[i]^'*'*"=3   "*sigma[j]^'*'*"=2
"*tau[i]^'*'*"=1   "*tau[j]^'*'*"=1"*""))
```

```
##############################################################################
##
##  Program that produces figure 5.2
##
##  Integrates the joint density function of correlations to
##  to obtain the density function of combined correlation
##
##############################################################################

library(scatterplot3d)

par(mfrow=c(1, 2))

n1<-80
n2<-40

sd11<-3
sd12<-2
sd21<-1
sd22<-1

feats<-500*499/2

delta<-0.01

a<-(sd11*sd12)/n1/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
b<-(sd21*sd22)/n2/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))

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

fn<-function(x, y, n1, n2) {
   c1<-(sd11*sd12)/n1/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
   d1<-(sd21*sd22)/n2/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
   1 / (2*c1*d1*beta(0.5, 0.5*(n1-2))*beta(0.5, 0.5*(n2-2))) * (1 -
((x+y)/(2*c1))^2)^(0.5*(n1-4)) * (1 - ((x-y)/(2*d1))^2)^(0.5*(n2-4))
}

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

apc<-NULL

for(i in 1:feats) {
   d11<-rnorm(n1, sd=sd11)
   d12<-rnorm(n1, sd=sd12)
   d21<-rnorm(n2, sd=sd21)
   d22<-rnorm(n2, sd=sd22)

   dat<-rbind(c(d11, d21), c(d12, d22))
   colnames(dat)<-c(rep("X", n1), rep("Y", n2))
   rownames(dat)<-paste("G", 1:2, sep="")
   apc<-rbind(apc, c(n1, n2, comb.cor(dat, colnames(dat))))
}

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

x1<- -(a+b)
y1<- b-a

x2<- b-a
y2<- -(a+b)

x3<- a+b
```

```
y3<- a-b

x4<- a-b
y4<- a+b

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

x<-seq(-(a+b)+delta/2, (a+b)-delta/2, by=delta)

out<-NULL

for(i in x) {
  ll<- y1 - (y1-y2)/(x1-x2)*x1 + (y1-y2)/(x1-x2)*i
  if ( abs(ll) > 1 ) {
    ll<- y2 - (y2-y3)/(x2-x3)*x2 + (y2-y3)/(x2-x3)*i
  }

  ul<- y1 - (y1-y4)/(x1-x4)*x1 + (y1-y4)/(x1-x4)*i
  if ( abs(ul) > 1 ) {
    ul<- y4 - (y4-y3)/(x4-x3)*x4 + (y4-y3)/(x4-x3)*i
  }

  y<-seq(ll, ul, by=delta)

  vol<-0

  for(j in y) {
    vol <- vol + fn(i, j, n1, n2)*delta
  }

  out<-rbind(out, c(i, vol))

}


x<-seq(-1, 1, length.out=100)
hist(apc[,3], breaks=x, freq=F, main=expression("Density of "*r[z]),
xlab="r", ylab="Density f(x)")
lines(out)
title(sub=expression(n[1]*"=80  "*n[2]*"=40  "*sigma[1]^'*'*"=3   "*sigma[2]^'*'*"=2
"*tau[1]^'*'*"=1   "*tau[2]^'*'*"=1"*""))


n1<-40
n2<-40

sd11<-1
sd12<-1
sd21<-1
sd22<-1

apc2<-NULL

for(i in 1:feats) {
   d11<-rnorm(n1, sd=sd11)
   d12<-rnorm(n1, sd=sd12)
   d21<-rnorm(n2, sd=sd21)
   d22<-rnorm(n2, sd=sd22)

   dat<-rbind(c(d11, d21), c(d12, d22))
   colnames(dat)<-c(rep("X", n1), rep("Y", n2))
   rownames(dat)<-paste("G", 1:2, sep="")
   apc2<-rbind(apc2, c(n1, n2, comb.cor(dat, colnames(dat))))
```

```
}

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

a<-(sd11*sd12)/n1/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
b<-(sd21*sd22)/n2/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))

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

x1<- -(a+b)
y1<- b-a

x2<- b-a
y2<- -(a+b)

x3<- a+b
y3<- a-b

x4<- a-b
y4<- a+b

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

x<-seq(-(a+b)+delta/2, (a+b)-delta/2, by=delta)

out<-NULL

for(i in x) {
  ll<- y1 - (y1-y2)/(x1-x2)*x1 + (y1-y2)/(x1-x2)*i
  if ( abs(ll) > 1 ) {
    ll<- y2 - (y2-y3)/(x2-x3)*x2 + (y2-y3)/(x2-x3)*i
  }

  ul<- y1 - (y1-y4)/(x1-x4)*x1 + (y1-y4)/(x1-x4)*i
  if ( abs(ul) > 1 ) {
    ul<- y4 - (y4-y3)/(x4-x3)*x4 + (y4-y3)/(x4-x3)*i
  }

  y<-seq(ll, ul, by=delta)

  vol<-0

  for(j in y) {
    vol <- vol + fn(i, j, n1, n2)*delta
  }

  out<-rbind(out, c(i, vol))

}

x<-seq(-1, 1, length.out=100)
hist(apc2[,3], breaks=x, freq=F, main=expression("Density of "*r[z]),
xlab="r", ylab="Density f(x)")
lines(out)
title(sub=expression(n[1]*"=40  "*n[2]*"=40  "*sigma[1]^'*'*"=1   "*sigma[2]^'*'*"=1
"*tau[1]^'*'*"=1   "*tau[2]^'*'*"=1"*""))
```

```
#####################################################################
##
##  Quantile function for combined correlations. Produces theoretical
##  and empirical quantiles plot and returns both quantiles
##  Input: data, quantile number, symbol to plot, whether to plot,
##
##  output: produces quantile-quantile plot
##          returns both theoretical and empirical quantiles
##
#####################################################################

corquantile<-function(dat, x.by=0.05, x.pch=1, oline=T, x.plot=T,
                      plot.xlab="Theoretical", plot.ylab="Data", plot.main="Quantile
plot") {

   n<-ncol(dat)

   cordensq<-function(qn, n) {
      t.fun<-function(x, y, n){
        ival<-integrate(dens.f, lower=-1, upper=x, n=n)$value
        abs(y-ival)
      }
      optimize(t.fun, interval=c(-1, 1), y=qn, n=n)$minimum
   }

   out<-NULL

   for(i in seq(0, 1, by=x.by)) {
     out<-rbind(out, c(i, cordensq(i, n)))
   }

   if(is.null(rownames(dat)))  rownames(dat)<-paste("G", 1:nrow(dat), sep="")

   apcx<-getallcors(dat)

   y<-quantile(apcx[,3], prob=out[,1])

   out<-cbind(Theoretical=out[,2], Data=y)

   out<-out[-c(1, nrow(out)),]
   if (x.plot) plot(out, pch=x.pch, xlab=plot.xlab, ylab=plot.ylab, main=plot.main)
   if (oline) abline(a=0, b=1)
   invisible(out)
}
```

```
#####################################################################
##
##   Function that computes the combined correlation (Eqn. 4.4) of
##   a given pair of features
##
##     inputs: data, grouping variable
##     output: combined correlation of the pair of variables
##
#####################################################################

   comb.cor<-function(dat, groups) {
     g1<-groups==unique(groups)[1]
```

```
     g2<-groups==unique(groups)[2]

     vx1<-var(dat[1,g1])/sum(g1)
     vx2<-var(dat[2,g1])/sum(g1)
     vy1<-var(dat[1,g2])/sum(g2)
     vy2<-var(dat[2,g2])/sum(g2)

     px<-cor(dat[1,g1], dat[2,g1])
     py<-cor(dat[1,g2], dat[2,g2])

     pc <- (px*sqrt(vx1*vx2)+py*sqrt(vy1*vy2))/sqrt((vx1+vy1)*(vx2+vy2))

    if (abs(pc)>=1) {
      if (pc>0) {
        pc <- 0.999
      }
      if (pc<0) {
        pc <- -0.999
      }
    }

    pc
}
```

```
#####################################################################
##
##   Program that computes all combined correlation of a given
##   dataset
##     inputs: data, grouping variable
##     output: all pairwise combined correlations in a list
##   Uses comb.cor function above
##
#####################################################################

allcomb.cor<-function(dat, groups) {

  N<-nrow(dat)*(nrow(dat)-1)/2
  pcor<-numeric(N)
  g1<-character(N)
  g2<-character(N)

  k<-0

  for( i in 1:nrow(dat) ) {
    for( j in i:nrow(dat) ) {
      if( i != j ) {
        k<-k+1
        d<-rbind(dat[i,], dat[j,])
        pcor[k]<-comb.cor(d, groups)
        g1[k]<-rownames(dat)[i]
        g2[k]<-rownames(dat)[j]
      }
    }
   }
  data.frame(g1=I(g1), g2=I(g2), pcor)
}
```

```
#####################################################################
##
##  The function that generates the distribution function
##  of combined correlation under independence (Eqn. 5.5).
##     input: sample sizes, delta (smaller for higher accuracy)
##            standard deviations
##
##     output: function that can be used to generate combined
##             correlation density function at r (0≤r≤1) values
##
#####################################################################

combcordens<-function(n1, n2, delta=0.0005, sd11=1, sd12=1, sd21=1, sd22=1) {

  fn<-function(x, y, n1, n2) {
     c1<-(sd11*sd12)/n1/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
     d1<-(sd21*sd22)/n2/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
     1 / (2*c1*d1*beta(0.5, 0.5*(n1-2))*beta(0.5, 0.5*(n2-2))) * (1 -
((x+y)/(2*c1))^2)^(0.5*(n1-4)) * (1 - ((x-y)/(2*d1))^2)^(0.5*(n2-4))
  }

  a<-(sd11*sd12)/n1/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))
  b<-(sd21*sd22)/n2/sqrt((sd11^2/n1+sd21^2/n2)*(sd12^2/n1+sd22^2/n2))

  x<-seq(-(a+b)+delta/2, (a+b)-delta/2, by=delta)

  out<-NULL

  x1<- -(a+b)
  y1<- b-a
  x2<- b-a
  y2<- -(a+b)
  x3<- a+b
  y3<- a-b
  x4<- a-b
  y4<- a+b

  for(i in x) {
     ll<- y1 - (y1-y2)/(x1-x2)*x1 + (y1-y2)/(x1-x2)*i
     if ( abs(ll) > 1 ) {
       ll<- y2 - (y2-y3)/(x2-x3)*x2 + (y2-y3)/(x2-x3)*i
     }

     ul<- y1 - (y1-y4)/(x1-x4)*x1 + (y1-y4)/(x1-x4)*i
     if ( abs(ul) > 1 ) {
       ul<- y4 - (y4-y3)/(x4-x3)*x4 + (y4-y3)/(x4-x3)*i
     }

     y<-seq(ll, ul, by=delta)
     vol<-0

     for(j in y) {
       vol <- vol + fn(i, j, n1, n2)*delta
     }

     out<-rbind(out, c(i, vol))
  }

  approxfun(out[,1], out[,2])
}
```

```
#####################################################################
##
##  Conditional Network Testing function
##   This function analyzes data in networks. Networks are constructed
##   by comparing the empirical and theoretical (under independence)
##   distribution of correlations. Features in the network are
##   conditionally tested using the conditional density of test
##   statistics. Analysis is done separately for features declared
##   significant and not significant by the initial test. Network
##   information can be accessed by passing g1 (for features declared
##   initially significant) and g2 (for features initially declared not
##   significant).
##
##  Inputs:  Data, group identifier, level of significance
##           ctrlcat: control factor for comparing correlation distributions
##           g1 and g2: for storing network information
##
##  Output: T-Test results (T-Statistic, p-values, d.f.) and CNT results
##          If g1 and g2 are specified networking information are assigned
##          to them which can be used for visualization of networks
##
##   Uses following functions
##      allcomb.cor
##      rowttest
##      combcordens
##      d.cond.dens
##      d.cond.dens2
##      p.cond.dens
##      p.cond.dens2
##
#####################################################################

nettest1<-function(dat, groups, alpha=0.05, ctrlfac=0.1, g1, g2) {

 ##
 ## The following part prepares the data, computes all combined correlations
 ## and generates initial tables for networking. These tables are used to keep
 ## grouping information and test information of each features being tested.
 ##

   gnames<-rownames(dat)

   d1<-dat[, groups==unique(groups)[1]]
   d2<-dat[, groups==unique(groups)[2]]
   s1<-apply(X=d1, MARGIN=1, FUN=sd)
   s2<-apply(X=d1, MARGIN=1, FUN=sd)
   d1<-d1/s1
   d2<-d2/s2
   dat2<-cbind(d1, d2)
   colnames(dat2)<-colnames(dat)
   rownames(dat2)<-rownames(dat)
   apc2<-allcomb.cor(dat2, groups)

   h2<-hist(apc2[,3], breaks=100, plot=F)
   smth<-smooth.spline(h2$mids, h2$density)
   edf<-approxfun(smth$x, smth$y, rule=1)

   tt<-rowttest(dat, colnames(dat))
   tt[,1]<-sign(tt[,1])*qnorm(tt[,2]/2, lower.tail=F)
   tt<-tt[order(tt[,2], decreasing=T),]
   apc2<-apc2[order(abs(apc2[,3]), decreasing=T),]
```

```
   tts<-matrix(tt[tt[,2]<alpha,], ncol=3)
   colnames(tts)<-colnames(tt)
   rownames(tts)<-rownames(tt)[tt[,2]<alpha]
   tta<-matrix(tt[tt[,2]>=alpha,], ncol=3)
   colnames(tta)<-colnames(tt)
   rownames(tta)<-rownames(tt)[tt[,2]>=alpha]

   i1<-match(apc2[,1], rownames(tts))
   i2<-match(apc2[,2], rownames(tts))

   i3<-i1+i2
   scor<-apc2[!is.na(i3),]
   i1<-match(apc2[,1], rownames(tta))
   i2<-match(apc2[,2], rownames(tta))
   i3<-i1+i2
   acor<-apc2[!is.na(i3),]
   rm(i1, i2, i3)

   cfun<-combcordens(ncol(dat1), ncol(dat2))
   id2<-cfun(h2$mids)
   id4<-edf(h2$mids)
   id4[is.na(id4)]<-min(id4[!is.na(id4)])
   id1<-id2*sum(h2$count)/sum(id2)
   mcounts<-id4*sum(h2$counts)/sum(id4)
   ctble<-data.frame(mids=h2$mids, lb=h2$breaks[-length(h2$breaks)], ub=h2$breaks[-1],
               count=round(mcounts, 0), idc=round(id1, 0))
   ctble<-data.frame(ctble, dif=ctble$count-ctble$idc, mods=numeric(nrow(ctble)))
   ctble[is.na(ctble)]<-0
   ttw<-data.frame(tts, chng=logical(nrow(tts)), pv2=numeric(nrow(tts)),
              grp=numeric(nrow(tts)))

   ngroups<-0
   groupings<-NULL
   mgns<-NULL
   glist<-NULL

   ctble$dif<-ctble$dif-max(ctble$dif)*ctrlfac
   tt<-cbind(tt, tt[,2])


##
## The section below creates networks within features that are declared
## as statistically significant by the initial test. Table named 'groupings'
## retains all grouping information
##

   if ( nrow(tts) > 1 ) {
      for(i in 1:nrow(tts)) {
         cgn<-rownames(tts)[i]
         ing<-as.character(groupings[!is.na(match(groupings$att, cgn)), 1])
         wcor<-scor[scor[,1]==cgn | scor[,2]==cgn,]
         wcor<-wcor[is.na(match(wcor[,1], ing)) &  is.na(match(wcor[,2], ing)),]
         if(nrow(wcor)<=0) next()

         for(j in 1:nrow(wcor)) {
             assigned<-F
             w2<-wcor[j,]
             if (abs(tt[cgn,1]) > abs(tt[w2[1,1:2][w2[1,1:2]!=cgn],1])) next()
             ctb<-ctble[ctble[,3]>as.numeric(w2[3]),][1,]
             if( ctb$dif > 0 & ctb$mods < ctb$dif ) {
                bgn<-w2[1:2][w2[1:2]!=cgn]
                cgs<-glist[glist[,1]==cgn | glist[,1]==bgn,2]
```

```
                if (length(cgs)==0) {
                   ngroups<-ngroups+1
                   gnumber<-ngroups
                } else {
                   glist[!is.na(match(glist[,2], cgs)),2]<-min(cgs)
                   groupings[!is.na(match(groupings$groupno, cgs)), 2]<-min(cgs)
                   gnumber <- min(cgs)
                }
                cc<-apc2[apc2[,1]==cgn & apc2[,2]==bgn | apc2[,1]==bgn &
apc2[,2]==cgn,3]
                groupings<-rbind(groupings, data.frame(gname=cgn, groupno=gnumber,
att=bgn, tvg=tts[cgn,1], tva=tts[bgn,1], cc=cc))
                ttw[c(cgn, bgn), "chng"]<-T
                ttw[c(cgn, bgn), "grp"]<-gnumber
                glist<-rbind(glist, data.frame(gname=cgn, grp=gnumber),
data.frame(gname=bgn, grp=gnumber))
                ctble[ctble$mids==ctb$mids,"mods"]<-
ctble[ctble$mids==ctb$mids,"mods"]+1
                assigned<-T
             }
             if (assigned) break()
          }
       }

##
##  The following section cycles through each group defined in 'groinpings' table
##  and draws a new p-value using the conditional densities. New p-values are stored
##  in the fourth column of object 'tt'
##

   for( grp in unique(groupings$groupno) ) {
       tgp<-groupings[groupings$groupno==grp,]
       i<-nrow(tgp)
       gns<-c(as.character(tgp[i,1]), as.character(tgp[i,3]))
       g1<-gns[max(abs(tgp[i,4:5]))==abs(tgp[i,4:5])]
       g2<-gns[gns!=g1]

       newq<-qnorm(alpha/2, lower.tail=F)

       adjlist<-data.frame(gname=g1, tv=tts[g1, 1], pv=tts[g1, 2], ncv=qnorm(alpha/2,
lower.tail=F), np=tts[g1,2])

       newp<-ifelse(tt[g1,2] < alpha,
              2*p.cond.dens(x=abs(tts[g2,1]), r=tgp[i,6], t=qnorm(alpha/2,
lower.tail=F), lower.tail=F),
              2*p.cond.dens2(x=abs(tts[g2,1]), r=tgp[i,6], t=qnorm(alpha/2,
lower.tail=F), lower.tail=F))
       newq<-ifelse(tt[g1,2] < alpha,
              q.cond.dens(alpha/2, r=tgp[i,6], lower.tail=F, interv=c(-4, 4)),
              q.cond.dens2(alpha/2, r=tgp[i,6], lower.tail=F, interv=c(-4, 4)))


       adjlist<-rbind(adjlist,
             data.frame(gname=g2, tv=tts[g2, 1], pv=tts[g2, 2], ncv=newq, np=newp))


       tgp<-tgp[-i,]

       repeat{

             i<-i-1

             if(i <= 0) {
```

```
                if (nrow(tgp)==0)
                    break()
                else
                    i<-nrow(tgp)
            }

            gns<-c(as.character(tgp[i,1]), as.character(tgp[i,3]))

            for(j in 1:nrow(adjlist) ){
                 agns<-c(as.character(adjlist[j,1]))
                 idx<-match(agns, gns)
                 if (!is.na(idx)) {
                      ga<-gns[match(agns, gns)[!is.na(match(agns, gns))]]
                      gc<-gns[is.na(match(gns, ga))]

                      newp<-ifelse(adjlist[j,5]<alpha,
                            2*p.cond.dens(abs(tts[gc,1]), r=tgp[i,6], t=adjlist[j,4],
lower.tail=F),
                            2*p.cond.dens2(abs(tts[gc,1]), r=tgp[i,6],
t=adjlist[j,3], lower.tail=F))
                      newq<-ifelse(adjlist[j,5]<alpha,
                            q.cond.dens.2(alpha/2, r=tgp[i,6], t=adjlist[j,4],
lower.tail=F),
                            q.cond.dens2.2(alpha/2, r=tgp[i,6], t=adjlist[j,4],
lower.tail=F))

                      tt[gc,4]<-newp
                      adjlist<-rbind(adjlist,
                      data.frame(gname=gc, tv=tts[gc, 1], pv=tts[gc, 2], ncv=newq,
np=newp))
                      tgp<-tgp[-i,]
                      break()
                 }
            }

       }
   }
   }




##
##  The section below constructs networks within features that were declared
##  as not significant in the initial test. Table named 'groupings2' retains
##  all grouping information.
##

   ttwa<-data.frame(tta, chng=logical(nrow(tta)), pv2=numeric(nrow(tta)),
grp=numeric(nrow(tta)))

   ngroups<-0
   groupings2<-NULL
   mgns<-NULL
   glist<-NULL
   i<-1
   j<-1

   for(i in 1:nrow(tta)) {
      cgn<-rownames(tta)[i]
      ing<-as.character(groupings2[!is.na(match(groupings2$att, cgn)), 1])
      wcor<-acor[acor[,1]==cgn | acor[,2]==cgn,]
```

```
      wcor<-wcor[is.na(match(wcor[,1], ing)) &  is.na(match(wcor[,2], ing)),]
      if(nrow(wcor)<=0) next()

      for(j in 1:nrow(wcor)) {
         assigned<-F
         w2<-wcor[j,]
         if (abs(tt[cgn,1]) > abs(tt[w2[1,1:2][w2[1,1:2]!=cgn],1])) next()
         ctb<-ctble[ctble[,3]>as.numeric(w2[3]),][1,]
         if( ctb$dif > 0 & ctb$mods < ctb$dif ) {
             bgn<-w2[1:2][w2[1:2]!=cgn]
             cgs<-glist[glist[,1]==cgn | glist[,1]==bgn,2]
             if (length(cgs)==0) {
                 ngroups<-ngroups+1
                 gnumber<-ngroups
             } else {
                 glist[!is.na(match(glist[,2], cgs)),2]<-min(cgs)
                 groupings2[!is.na(match(groupings2$groupno, cgs)), 2]<-min(cgs)
                 gnumber <- min(cgs)
             }
             cc<-apc2[apc2[,1]==cgn & apc2[,2]==bgn | apc2[,1]==bgn & apc2[,2]==cgn,3]
             groupings2<-rbind(groupings2, data.frame(gname=cgn, groupno=gnumber,
att=bgn, tvg=tta[cgn,1], tva=tta[bgn,1], cc=cc))
             ttwa[c(cgn, bgn), "chng"]<-T
             ttwa[c(cgn, bgn), "grp"]<-gnumber
             glist<-rbind(glist, data.frame(gname=cgn, grp=gnumber),
data.frame(gname=bgn, grp=gnumber))
             ctble[ctble$mids==ctb$mids,"mods"]<-ctble[ctble$mids==ctb$mids,"mods"]+1
             assigned<-T
         }
         if (assigned) break()
      }
   }

##
##  The following section tests the networks within features that were declared
##  as not significant by cycling through the networks defined in 'groupings2'.
##  New p-values are drawn from the conditional densities and stored in the fourth
##  column of the object 'tt'
##

   for( grp in unique(groupings2$groupno) ) {

         tgp<-groupings2[groupings2$groupno==grp,]
         i<-nrow(tgp)
         gns<-c(as.character(tgp[i,1]), as.character(tgp[i,3]))
         g1<-gns[min(abs(tgp[i,4:5]))==abs(tgp[i,4:5])]
         g2<-gns[gns!=g1]

         newq<-qnorm(alpha/2, lower.tail=F)

         adjlist<-data.frame(gname=g1, tv=tta[g1, 1], pv=tta[g1, 2], ncv=qnorm(alpha/2,
lower.tail=F), np=tta[g1,2])

         newp<-ifelse(tt[g1,2] < alpha,
               2*p.cond.dens(x=abs(tta[g2,1]), r=tgp[i,6], t=qnorm(alpha/2,
lower.tail=F), lower.tail=F),
               2*p.cond.dens2(x=abs(tta[g2,1]), r=tgp[i,6], t=qnorm(alpha/2,
lower.tail=F), lower.tail=F))
         newq<-ifelse(tt[g1,2] < alpha,
               q.cond.dens(alpha/2, r=tgp[i,6], lower.tail=F, interv=c(-4, 4)),
               q.cond.dens2(alpha/2, r=tgp[i,6], lower.tail=F, interv=c(-4, 4)))
```

```
        adjlist<-rbind(adjlist,
              data.frame(gname=g2, tv=tta[g2, 1], pv=tta[g2, 2], ncv=newq, np=newp))


        tgp<-tgp[-i,]

        repeat{

             i<-i-1

             if(i <= 0) {
                if (nrow(tgp)==0)
                    break()
                else
                    i<-nrow(tgp)
             }

             gns<-c(as.character(tgp[i,1]), as.character(tgp[i,3]))

             for(j in 1:nrow(adjlist) ){
                agns<-c(as.character(adjlist[j,1]))
                idx<-match(agns, gns)
                if (!is.na(idx)) {
                     ga<-gns[match(agns, gns)[!is.na(match(agns, gns))]]
                     gc<-gns[is.na(match(gns, ga))]

                     newp<-ifelse(adjlist[j,5]<alpha,
                         2*p.cond.dens(abs(tta[gc,1]), r=tgp[i,6], t=adjlist[j,4],
lower.tail=F),
                         2*p.cond.dens2(abs(tta[gc,1]), r=tgp[i,6], t=adjlist[j,3],
lower.tail=F))
                     newq<-ifelse(adjlist[j,5]<alpha,
                         q.cond.dens.2(alpha/2, r=tgp[i,6], t=adjlist[j,4],
lower.tail=F),
                         q.cond.dens2.2(alpha/2, r=tgp[i,6], t=adjlist[j,4],
lower.tail=F))

                     tt[gc,4]<-newp
                     adjlist<-rbind(adjlist,
                     data.frame(gname=gc, tv=tta[gc, 1], pv=tta[gc, 2], ncv=newq,
np=newp))
                     tgp<-tgp[-i,]
                     break()
                }
             }

        }
   }
   if (!missing(g1)) g1<<-groupings
   if (!missing(g2)) g2<<-groupings2
   tt[gnames,]   # Outputs the t-test results along with CNT results
}
```

```
##############################################################################
##
##   Intermediate function that analyzes group information from nettest1 function
##   and creates the .net file that can be opened for visualization in Pajek.
##   This function includes all networks in the same .net file.
##
##   Inputs:
##     groupings: grouping information from CNT function
##     filename: filename to output .net file
##   Output:
##     Saves a .net file in the given file name that can be opened with Pajek
##
##############################################################################

drawnetworks<-function(groupings, filename=NULL) {

   if (is.null(filename)) stop("An output file is required")

   gtble<-groupings[order(groupings[,2]),c(3,1,2)]

   x1<-groupings[,4]
   names(x1)<-groupings[,1]

   x2<-groupings[,5]
   names(x2)<-groupings[,3]

   sizes<-c(x1, x2)
   sizes<-abs(sizes[unique(names(sizes))])
   sizes<-sizes/(max(sizes)+0.1)

   outtext<-NULL
   postext1<-NULL
   postext2<-NULL
   siztext<-NULL
   outlist<-NULL

   k<-1
   for(g in 1:length(unique(gtble[,3]))) {
      xcor<-1
      tmpg<-gtble[gtble[,3]==unique(gtble[,3])[g],]
      tmpg<-tmpg[order(tmpg[,1]),]

      gns<-unique(c(as.character(tmpg[,1]), as.character(tmpg[,2])))
      for(i in 1:length(gns)) {
         outtext<-c(outtext, paste(k, " \"", gns[i],"\"", sep=""))
         siztext<-c(siztext, sizes[gns[i]])
         postext1<-c(postext1, xcor+i)
         xcor<-xcor+1
         postext2<-c(postext2, g+i)
         outlist<-c(outlist, gns[i])
         k<-k+1
      }
   }

   postext1<-postext1/max(postext1)
   postext2<-postext2/max(postext2)

   outtext<-paste(outtext, postext1, postext2, siztext)
   outtext<-c(paste("*Vertices", k-1), outtext, "*Arcs")

   seqn<-1:length(outlist)
   names(seqn)<-outlist
```

```
   for(i in 1:nrow(gtble)) {
      outtext<-c(outtext, paste(seqn[as.character(gtble[i,1])],
seqn[as.character(gtble[i,2])]))
   }

   fileconn<-file(filename)
   writeLines(outtext, fileconn)
   close(fileconn)
}
```

```
##############################################################################
##
##   Intermediate function that analyzes group information from nettest1 function
##   and creates the .net file that can be opened for visualization in Pajek.
##   This function creates a separate file for each network in grouping table.
##   The first portion of the .net file names must be provided
##
##   Inputs:
##     groupings: grouping information from CNT function
##     filename: first portion of the file name
##   Output:
##     Saves a .net files for each network in with specified file names
##
##############################################################################

drawnetworks2<-function(groupings, filehead=NULL) {
   vistext<-NULL

   for(g in 1:length(unique(groupings[,2]))) {

      tgroupings<-groupings[groupings[,2]==unique(groupings[,2])[g],]

      gtble<-tgroupings[,c(3,1,2)]

      x1<-tgroupings[,4]
      names(x1)<-tgroupings[,1]

      x2<-tgroupings[,5]
      names(x2)<-tgroupings[,3]

      sizes<-c(x1, x2)
      sizes<-abs(sizes[unique(names(sizes))])
      sizes<-sizes/(max(sizes)+0.1)

      outtext<-NULL
      postext1<-NULL
      postext2<-NULL
      siztext<-NULL
      outlist<-NULL

      k<-1
      xcor<-1

      gns<-unique(c(as.character(gtble[,1]), as.character(gtble[,2])))
      for(i in 1:length(gns)) {
         outtext<-c(outtext, paste(k, " \"", gns[i],"\"", sep=""))
         siztext<-c(siztext, sizes[gns[i]])
         postext1<-c(postext1, xcor+i)
         xcor<-xcor+1
         postext2<-c(postext2, i)
```

```
         outlist<-c(outlist, gns[i])
         k<-k+1
      }

      postext1<-postext1/max(postext1)
      postext2<-postext2/max(postext2)

      outtext<-paste(outtext, postext1, postext2, siztext)
      outtext<-c(paste("*Vertices", k-1), outtext, "*Arcs")

      seqn<-1:length(outlist)
      names(seqn)<-outlist

      vistext<-c(vistext, paste("Group", g, ":", k-1, "Vertices"))

      for(i in 1:nrow(gtble)) {
         outtext<-c(outtext, paste(seqn[as.character(gtble[i,1])],
seqn[as.character(gtble[i,2])]))
      }

      fileconn<-file(paste(filehead, g, "net", sep="."))
      writeLines(outtext, fileconn)
      close(fileconn)

   }
      vistext<-c(paste(g, "Groups: Summary"), vistext)
      fileconn<-file(paste(filehead, "summary.txt", sep=""))
      writeLines(vistext, fileconn)
      close(fileconn)

}
```

```
##############################################################################
##
##  Function to generate plasmode data given a dataset
##
##  Input:
##     dat: template data
##     groups: groupings information (vector)
##     differences: weather to add differences (if not all null)
##     perc: proportion to make different (if differences is true)
##
##   Output:
##     Plasmode dataset
##
##############################################################################

plasmode<-function(dat, groups, differences=F, prec=0) {
   source("rowttest.r")
   dat1<-dat[, colnames(dat)==unique(groups)[1]]
   dat2<-dat[, colnames(dat)==unique(groups)[2]]

   mn1<-apply(dat1, 1, mean)
   mn2<-apply(dat2, 1, mean)

   dat1<-dat1-mn1
   dat2<-dat2-mn2

   tt<-rowttest(dat, groups)
   probs<-abs(tt[,1])/sum(abs(tt[m,1]))
```

```
   vr1<-apply(dat1, 1, var)
   vr2<-apply(dat2, 1, var)
   n1<-ncol(dat1)
   n2<-ncol(dat2)
   vfac<-sqrt(vr1/n1+vr2/n2)

   dat1x<-dat1[, sample(1:ncol(dat1), ncol(dat1), replace=T)]
   dat2x<-dat2[, sample(1:ncol(dat2), ncol(dat2), replace=T)]

   if (differences) {
       ndif<-round(nrow(dat)*perc, 0)
       indx<-sample(1:nrow(dat), ndif, prob=probs)
       dat1x[indx,]<-dat1x[indx,]+tt[indx]*vfac[indx]
   }

   cbind(dat1x, dat2x)
}
```

```
##########################################################################
##
##
##   Conditional density functions with specific standardized mean
##   differences. These functions are used to evaluate power of
##   conditional network testing
##
##   functions:
##       alt.cond.dens1: pdf for the density with non-zero means when H0 is rejected
##       alt.cond.dens2: pdf for the density with non-zero means when H0 is not
##         rejected
##       p.alt.cond.dens1: cdf for the density with non-zero means when H0 is rejected
##       p.alt.cond.dens2: cdf for the density with non-zero means when H0 is not
##         rejected
##
##   Inputs:
##      z2, z2.val: z values for pdfs and cdfs
##      r: combined correlation between test statistics
##      mu1: mean for the ith test (mu1 is set to zero in this dissertation
##      mu2: mean for the jth tests
##      z.star: cutoff value for a given level of significance under independence
##
##   Outputs:
##      pdf and cdf values for specified paramers
##
##########################################################################

alt.cond.dens1<-function(z2, r, mu1, mu2, z.star) {
  dnorm(z2-mu2)*(1+pnorm((-z.star-(mu1+r*(z2-mu2)))/sqrt(1-r^2))-pnorm((z.star-
(mu1+r*(z2-mu2)))/sqrt(1-r^2)))/(pnorm(-z.star-mu1)+1-pnorm(z.star-mu1))
}

alt.cond.dens2<-function(z2, r, mu1, mu2, z.star) {
  dnorm(z2-mu2)*(pnorm((z.star-(mu1+r*(z2-mu2)))/sqrt(1-r^2))-pnorm((-z.star-
(mu1+r*(z2-mu2)))/sqrt(1-r^2)))/(pnorm(z.star-mu1)-pnorm(-z.star-mu1))
}

p.alt.cond.dens1<-function(z2.val, r, mu1, mu2, z.star) {
  integrate(alt.cond.dens1, lower=-Inf, upper=z2.val, r=r, mu1=mu1, mu2=mu2,
z.star=z.star)$value
}

p.alt.cond.dens2<-function(z2.val, r, mu1, mu2, z.star) {
  integrate(alt.cond.dens2, lower=-Inf, upper=z2.val, r=r, mu1=mu1, mu2=mu2,
z.star=z.star)$value
}
```

```
##########################################################################
##
##   Program generating power curves for the conditional network tests.
##   Uses functioins
##       alt.cond.dens1
##
##########################################################################

alpha<-0.05

r.vals<-c(-0.6, 0.6)
mu1.vals<-c(-2, 0, 1)
mu2.vals<-seq(-4, 4, length.out=50)
```

```
x<-seq(-4, 4, length.out=50)
mu1<-0
r<-0.6

out<-NULL
o2<-NULL
o3<-NULL

qm<-abs(q.cond.dens.2(alpha/2, r=r, t=1.96))

for(mu2 in mu2.vals) {

   a1<-integrate(alt.cond.dens1, lower=-Inf, upper=-qm, mu1=mu1, r=r, mu2=mu2,
z.star=1.96)$value+
       integrate(alt.cond.dens1, lower=qm, upper=Inf, mu1=mu1, r=r, mu2=mu2,
z.star=1.96)$value

   a2<-pnorm(-1.96, mean=mu2)+pnorm(1.96, mean=mu2, lower.tail=F)

   a3<-integrate(alt.cond.dens1, lower=-Inf, upper=-1.96, mu1=mu1, r=r, mu2=mu2,
z.star=1.96)$value+
       integrate(alt.cond.dens1, lower=1.96, upper=Inf, mu1=mu1, r=r, mu2=mu2,
z.star=1.96)$value

   out<-c(out, a1)
   o2<-c(o2, a2)
   o3<-c(o3, a3)
}

par(mfrow=c(1, 2))

plot(mu2.vals, out, type="l", xlab=expression(lambda[j]), ylab="Power", main="Power of
conditional tests", axes=F)
axis(1)
axis(2)
lines(mu2.vals, o2, lty=2)
lines(mu2.vals, o3, lty=3)

out<-NULL
o2<-NULL
o3<-NULL

qm<-abs(q.cond.dens2.2(alpha/2, r=r, t=1.96))

for(mu2 in mu2.vals) {

   a1<-integrate(alt.cond.dens2, lower=-Inf, upper=-qm, mu1=mu1, r=r, mu2=mu2,
z.star=1.96)$value+
       integrate(alt.cond.dens2, lower=qm, upper=Inf, mu1=mu1, r=r, mu2=mu2,
z.star=1.96)$value

   a2<-pnorm(-1.96, mean=mu2)+pnorm(1.96, mean=mu2, lower.tail=F)

   a3<-integrate(alt.cond.dens2, lower=-Inf, upper=-1.96, mu1=mu1, r=r, mu2=mu2,
z.star=1.96)$value+
       integrate(alt.cond.dens2, lower=1.96, upper=Inf, mu1=mu1, r=r, mu2=mu2,
z.star=1.96)$value

   out<-c(out, a1)
   o2<-c(o2, a2)
   o3<-c(o3, a3)
}
```

```
plot(mu2.vals, out, type="l", xlab=expression(lambda[j]), ylab="Power", main="Power of
conditional tests", axes=F)
axis(1)
axis(2)
lines(mu2.vals, o2, lty=2)
lines(mu2.vals, o3, lty=3)

legend(legend=c("Power under independence", "Power under dependence", "Power under
modified test"), lty=c(2, 3, 1), cex=0.7)
x11(width=9, height=1)
par(mar=c(0, 0, 0, 0))
plot(0, 0, axes=F, xlab="", ylab="", col="white")
legend("center",legend=c("Power under independence", "Power under dependence", "Power
under modified test"), lty=c(2, 3, 1), cex=0.7)
```

```
#########################################################################
##
##   Program that produces plots for comparing false discovery proportion,
##   type I error, type II errors (Figure 9.2)
##
##   Uses previously saved simulation data
##
#########################################################################


ems<-function(dat, n) {

 a<-dat[,1]       # Number of Sigs
 b<-dat[,2]       # total rejections
 c<-dat[,3]       # true rejections

 fdr<-(b-c)/b
 t1e<-(b-c)/(n-a)
 t2e<-(a-c)/a

 mn.fdr<-mean(fdr[!is.nan(fdr)])
 vr.fdr<-var(fdr[!is.nan(fdr)])

 mn.t1e<-mean(t1e[!is.nan(t1e)])
 vr.t1e<-var(t1e[!is.nan(t1e)])

 mn.t2e<-mean(t2e[!is.nan(t2e)])
 vr.t2e<-var(t2e[!is.nan(t2e)])

 c(mn.fdr, vr.fdr, mn.t1e, vr.t1e, mn.t2e, vr.t2e)
}

tt.out<-NULL
cn.out<-NULL
mn.ef.out<-NULL
vr.ef.out<-NULL

mn.qvtt.out<-NULL
vr.qvtt.out<-NULL

mn.qvcn.out<-NULL
vr.qvcn.out<-NULL

bh.tt.out<-NULL
bh.cn.out<-NULL
```

```
by.tt.out<-NULL
by.cn.out<-NULL

qv.tt.out<-NULL
qv.cn.out<-NULL

for(i in 1:6) {
  load(paste("simd11",i,"rData", sep="."))
  print(dim(outmat))

  tt.out<-rbind(tt.out, ems(outmat[,c(2, 3, 4)], n=500))
  cn.out<-rbind(cn.out, ems(outmat[,c(2, 5, 6)], n=500))
  mn.ef.out<-c(mn.ef.out, mean(outmat[,27]))
  vr.ef.out<-c(vr.ef.out, var(outmat[,27]))

  mn.qvtt.out<-c(mn.qvtt.out, mean(outmat[,28]))
  vr.qvtt.out<-c(vr.qvtt.out, var(outmat[,28]))

  mn.qvcn.out<-c(mn.qvcn.out, mean(outmat[,29]))
  vr.qvcn.out<-c(vr.qvcn.out, var(outmat[,29]))

  bh.tt.out<-rbind(bh.tt.out, ems(outmat[,c(2, 15, 16)], n=500))
  bh.cn.out<-rbind(bh.cn.out, ems(outmat[,c(2, 17, 18)], n=500))

  by.tt.out<-rbind(by.tt.out, ems(outmat[,c(2, 19, 20)], n=500))
  by.cn.out<-rbind(by.cn.out, ems(outmat[,c(2, 21, 22)], n=500))

  qv.tt.out<-rbind(qv.tt.out, ems(outmat[,c(2, 23, 24)], n=500))
  qv.cn.out<-rbind(qv.cn.out, ems(outmat[,c(2, 25, 26)], n=500))

}

plot(tt.out[,1], type="b", ylim=c(0, 1), axes=F, xlab="Corr. Structure", ylab="FDP and
Estimates", main="False Discovery Proportions
and Estimates")
axis(2)
axis(1, labels=LETTERS[1:6], at=1:6)
lines(cn.out[,1], type="b", pch=2)
lines(mn.ef.out, type="b", pch=3)
lines(mn.qvtt.out, type="b", pch=4)
lines(mn.qvcn.out, type="b", pch=5)

legend("topleft", inset=0.02, legend=c("True FDP by T-Tests", "True FDP by CNT",
"Efrons estimate of FDR", "Storey estimate of FDR on T-Test", "Storey esimate of FDR
on CNT"), pch=1:5)

plot(tt.out[,2], type="b", ylim=c(0, 0.04), axes=F, xlab="Corr. Structure",
ylab="Variance of FDP and Estimates", main="Variance of False Discovery Proportions
and Estimates")
axis(2)
axis(1, labels=LETTERS[1:6], at=1:6)
lines(cn.out[,2], type="b", pch=2)
lines(vr.ef.out, type="b", pch=3)
lines(vr.qvtt.out, type="b", pch=4)
lines(vr.qvcn.out, type="b", pch=5)

legend("topleft", inset=0.02, legend=c("True FDP by T-Tests", "True FDP by CNT",
"Efrons estimate of FDR", "Storey estimate of FDR on T-Test", "Storey esimate of FDR
on CNT"), pch=1:5)

plot(tt.out[,1], type="b", ylim=c(0, .08), axes=F, xlab="Corr. Structure", ylab="True
FDP", main="False Discovery Proportions by
different FDR control methods")
```

```
axis(2)
axis(1, labels=LETTERS[1:6], at=1:6)
lines(cn.out[,1], type="b", pch=2)
lines(bh.tt.out[,1], type="b", pch=2, lty=2)
lines(bh.cn.out[,1], type="b", pch=2)
lines(by.tt.out[,1], type="b", pch=3, lty=2)
lines(by.cn.out[,1], type="b", pch=3)
lines(qv.tt.out[,1], type="b", pch=4, lty=2)
lines(qv.cn.out[,1], type="b", pch=4)

legend(x=1, y=0.08, legend=c("Benjamini & Hochberg", "Benjamini & Yekuteilli",
"Storeys Q Value"), pch=2:4)
legend(x=3.5, y=0.08, legend=c("Applied to T-Test", "Applied to CNT"), lty=2:1)

plot(tt.out[,2], type="b", ylim=c(0, .015), axes=F, xlab="Corr. Structure",
ylab="Variance of True FDP", main="Variance of False Discovery Proportions
by different FDR control methods", col='white')
axis(2)
axis(1, labels=LETTERS[1:6], at=1:6)
lines(cn.out[,2], type="b", pch=2, col='white')
lines(bh.tt.out[,2], type="b", pch=2, lty=2)
lines(bh.cn.out[,2], type="b", pch=2)
lines(by.tt.out[,2], type="b", pch=3, lty=2)
lines(by.cn.out[,2], type="b", pch=3)
lines(qv.tt.out[,2], type="b", pch=4, lty=2)
lines(qv.cn.out[,2], type="b", pch=4)

legend(x=1, y=0.015, legend=c("Benjamini & Hochberg", "Benjamini & Yekuteilli",
"Storeys Q Value"), pch=2:4)
legend(x=1, y=0.011, legend=c("Applied to T-Test", "Applied to CNT"), lty=2:1)

plot(tt.out[,3], type="b", ylim=c(0, .008), axes=F, xlab="Corr. Structure", ylab="Type
I Error Proportion", main="Type I error proportions from
different FDR control methods")
axis(2)
axis(1, labels=LETTERS[1:6], at=1:6)
lines(cn.out[,3], type="b", pch=2)
lines(bh.tt.out[,3], type="b", pch=2, lty=2)
lines(bh.cn.out[,3], type="b", pch=2)
lines(by.tt.out[,3], type="b", pch=3, lty=2)
lines(by.cn.out[,3], type="b", pch=3)
lines(qv.tt.out[,3], type="b", pch=4, lty=2)
lines(qv.cn.out[,3], type="b", pch=4)

legend(x=1, y=0.008, legend=c("Benjamini & Hochberg", "Benjamini & Yekuteilli",
"Storeys Q Value"), pch=2:4)
legend(x=3.5, y=0.008, legend=c("Applied to T-Test", "Applied to CNT"), lty=2:1)

##
plot(tt.out[,4], type="b", ylim=c(0, .0002), axes=F, xlab="Corr. Structure",
ylab="Type I Error Rate", main="Type I error rates from
different FDR control methods")
axis(2)
axis(1, labels=LETTERS[1:6], at=1:6)
lines(cn.out[,4], type="b", pch=2)
lines(bh.tt.out[,4], type="b", pch=2, lty=2)
lines(bh.cn.out[,4], type="b", pch=2)
lines(by.tt.out[,4], type="b", pch=3, lty=2)
lines(by.cn.out[,4], type="b", pch=3)
lines(qv.tt.out[,4], type="b", pch=4, lty=2)
lines(qv.cn.out[,4], type="b", pch=4)
```

```
legend(x=1, y=0.008, legend=c("Benjamini & Hochberg", "Benjamini & Yekuteilli",
"Storeys Q Value"), pch=2:4)
legend(x=3.5, y=0.008, legend=c("Applied to T-Test", "Applied to CNT"), lty=2:1)


##

plot(tt.out[,5], type="b", ylim=c(0, 1), axes=F, xlab="Corr. Structure", ylab="Type II
Error Proportion", main="Type II error proportions from
different FDR control methods", col='white')
axis(2)
axis(1, labels=LETTERS[1:6], at=1:6)
lines(cn.out[,5], type="b", pch=2, col='white')
lines(bh.tt.out[,5], type="b", pch=2, lty=2)
lines(bh.cn.out[,5], type="b", pch=2)
lines(by.tt.out[,5], type="b", pch=3, lty=2)
lines(by.cn.out[,5], type="b", pch=3)
lines(qv.tt.out[,5], type="b", pch=4, lty=2)
lines(qv.cn.out[,5], type="b", pch=4)

legend(x=1, y=1, legend=c("Benjamini & Hochberg", "Benjamini & Yekuteilli", "Storeys Q
Value"), pch=2:4)
legend(x=3.5, y=1, legend=c("Applied to T-Test", "Applied to CNT"), lty=2:1)


plot(tt.out[,6], type="b", ylim=c(0, .04), axes=F, xlab="Corr. Structure",
ylab="Variance of Type II Error proportion", main="Variance of Type II error
proportions
from different FDR control methods", col='white')
axis(2)
axis(1, labels=LETTERS[1:6], at=1:6)
lines(cn.out[,6], type="b", pch=2, col='white')
lines(bh.tt.out[,6], type="b", pch=2, lty=2)
lines(bh.cn.out[,6], type="b", pch=2)
lines(by.tt.out[,6], type="b", pch=3, lty=2)
lines(by.cn.out[,6], type="b", pch=3)
lines(qv.tt.out[,6], type="b", pch=4, lty=2)
lines(qv.cn.out[,6], type="b", pch=4)

legend(x=1, y=0.04, legend=c("Benjamini & Hochberg", "Benjamini & Yekuteilli",
"Storeys Q Value"), pch=2:4)
legend(x=3.5, y=0.04, legend=c("Applied to T-Test", "Applied to CNT"), lty=2:1)
```

```
#########################################################################
##
##   Program that produces plots for comparing false discovery proportion,
##   type I error, type II errors using plasmode data (Figure 9.8, 9.9 and
#    9.10)
##
##   Uses previously saved simulation data
##
#########################################################################

load("D:\\dilan\\research\\new\\sim26\\sim26.sum.rData")
load("D:\\dilan\\research\\new\\sim27\\sim27.sum.rData")

#FDP

plot(c(1, 2, 4, 5, 7, 8, 12, 13, 15, 16, 17, 18), c(mnbhtfdr4, mnbhcfdr4, mnbytfdr4,
mnbycfdr4, mnqvtfdr4, mnqvcfdr4, mnbhtfdr3, mnbhcfdr3, mnbytfdr3, mnbycfdr3,
mnqvtfdr3, mnqvcfdr3),
```

```
     ylim=c(-0.01, 0.25), pch=1:2,
     col=rep(c(2:4, 2:4), rep(2,6)), xlab="Dataset", ylab="False Discovery
Proportion", main="False Discovery Proportion for Plasmode Data", axes=F)
axis(2)
axis(1, labels=c("","Lung Cancer", "Multiple Myeloma",""), at=c(2, 4.5, 15.5, 17),
col.tick='white')
segments(c(1, 2, 4, 5, 7, 8, 12, 13, 15, 16, 17, 18), c(mnbhtfdr4-sdbhtfdr4,
mnbhcfdr4-sdbhcfdr4, mnbytfdr4-sdbytfdr4, mnbycfdr4-sdbycfdr4, mnqvtfdr4-sdqvtfdr4,
mnqvcfdr4-sdqvcfdr4, mnbhtfdr3-sdbhtfdr3, mnbhcfdr3-sdbhcfdr3, mnbytfdr3-sdbytfdr3,
mnbycfdr3-sdbycfdr3, mnqvtfdr3-sdqvtfdr3, mnqvcfdr3-sdqvcfdr3),
         c(1, 2, 4, 5, 7, 8, 12, 13, 15, 16, 17, 18), c(mnbhtfdr4+sdbhtfdr4,
mnbhcfdr4+sdbhcfdr4, mnbytfdr4+sdbytfdr4, mnbycfdr4+sdbycfdr4, mnqvtfdr4+sdqvtfdr4,
mnqvcfdr4+sdqvcfdr4, mnbhtfdr3+sdbhtfdr3, mnbhcfdr3+sdbhcfdr3, mnbytfdr3+sdbytfdr3,
mnbycfdr3+sdbycfdr3, mnqvtfdr3+sdqvtfdr3, mnqvcfdr3+sdqvcfdr3),
     col=rep(c(2:4, 2:4), rep(2,6)))

legend(x=2, y=0.25, legend=c("Benjamini & Hochberg", "Benjamini & Yekutieli", "Q-
Value"), lty=1, col=2:4)
legend(x=9, y=0.25, legend=c("T-Test", "CNT"), pch=1:2)



# T1E

plot(c(1, 2, 4, 5, 7, 8, 12, 13, 15, 16, 17, 18), c(mnbhtt1e4, mnbhct1e4, mnbytt1e4,
mnbyct1e4, mnqvtt1e4, mnqvct1e4,     mnbhtt1e3, mnbhct1e3, mnbytt1e3, mnbyct1e3,
mnqvtt1e3, mnqvct1e3),
     ylim=c(-0.005, 0.03), pch=1:2,
     col=rep(c(2:4, 2:4), rep(2,6)), xlab="Dataset", ylab="Type I Error Proportion",
main="Type I Error Proportion for Plasmode Data", axes=F)
axis(2)
axis(1, labels=c("","Lung Cancer", "Multiple Myeloma",""), at=c(2, 4.5, 15.5, 17),
col.tick='white')
segments(c(1, 2, 4, 5, 7, 8, 12, 13, 15, 16, 17, 18), c(mnbhtt1e4-sdbhtt1e4,
mnbhct1e4-sdbhct1e4, mnbytt1e4-sdbytt1e4, mnbyct1e4-sdbyct1e4, mnqvtt1e4-sdqvtt1e4,
mnqvct1e4-sdqvct1e4, mnbhtt1e3-sdbhtt1e3, mnbhct1e3-sdbhct1e3, mnbytt1e3-sdbytt1e3,
mnbyct1e3-sdbyct1e3, mnqvtt1e3-sdqvtt1e3, mnqvct1e3-sdqvct1e3),
         c(1, 2, 4, 5, 7, 8, 12, 13, 15, 16, 17, 18), c(mnbhtt1e4+sdbhtt1e4,
mnbhct1e4+sdbhct1e4, mnbytt1e4+sdbytt1e4, mnbyct1e4+sdbyct1e4, mnqvtt1e4+sdqvtt1e4,
mnqvct1e4+sdqvct1e4, mnbhtt1e3+sdbhtt1e3, mnbhct1e3+sdbhct1e3, mnbytt1e3+sdbytt1e3,
mnbyct1e3+sdbyct1e3, mnqvtt1e3+sdqvtt1e3, mnqvct1e3+sdqvct1e3),
     col=rep(c(2:4, 2:4), rep(2,6)))

legend(x=1, y=0.03, legend=c("Benjamini & Hochberg", "Benjamini & Yekutieli", "Q-
Value"), lty=1, col=2:4)
legend(x=9, y=0.03, legend=c("T-Test", "CNT"), pch=1:2)
abline(h=0, lty=2, col=8)



# T2E

plot(c(1, 2, 4, 5, 7, 8, 12, 13, 15, 16, 17, 18), c(mnbhtt2e4, mnbhct2e4, mnbytt2e4,
mnbyct2e4, mnqvtt2e4, mnqvct2e4,     mnbhtt2e3, mnbhct2e3, mnbytt2e3, mnbyct2e3,
mnqvtt2e3, mnqvct2e3),
     ylim=c(0, 0.25), pch=1:2,
     col=rep(c(2:4, 2:4), rep(2,6)), xlab="Dataset", ylab="Type II Error Proportion",
main="Type II Error Proportion for Plasmode Data", axes=F)
axis(2)
axis(1, labels=c("","Lung Cancer", "Multiple Myeloma",""), at=c(2, 4.5, 15.5, 17),
col.tick='white')
segments(c(1, 2, 4, 5, 7, 8, 12, 13, 15, 16, 17, 18), c(mnbhtt2e4-sdbhtt2e4,
mnbhct2e4-sdbhct2e4, mnbytt2e4-sdbytt2e4, mnbyct2e4-sdbyct2e4, mnqvtt2e4-sdqvtt2e4,
```

```
mnqvct2e4-sdqvct2e4, mnbhtt2e3-sdbhtt2e3, mnbhct2e3-sdbhct2e3, mnbytt2e3-sdbytt2e3,
mnbyct2e3-sdbyct2e3, mnqvtt2e3-sdqvtt2e3, mnqvct2e3-sdqvct2e3),
         c(1, 2, 4, 5, 7, 8, 12, 13, 15, 16, 17, 18), c(mnbhtt2e4+sdbhtt2e4,
mnbhct2e4+sdbhct2e4, mnbytt2e4+sdbytt2e4, mnbyct2e4+sdbyct2e4, mnqvtt2e4+sdqvtt2e4,
mnqvct2e4+sdqvct2e4, mnbhtt2e3+sdbhtt2e3, mnbhct2e3+sdbhct2e3, mnbytt2e3+sdbytt2e3,
mnbyct2e3+sdbyct2e3, mnqvtt2e3+sdqvtt2e3, mnqvct2e3+sdqvct2e3),
     col=rep(c(2:4, 2:4), rep(2,6)))

legend(x=2, y=0.25, legend=c("Benjamini & Hochberg", "Benjamini & Yekutieli", "Q-
Value"), lty=1, col=2:4)
legend(x=9, y=0.25, legend=c("T-Test", "CNT"), pch=1:2)
```