12-2020

Gene Set Testing by Distance Correlation

Sho-Hsien Su
University of Arkansas, Fayetteville

Gene Set Testing by Distance Correlation

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Statistics and Analytics

by

Sho-Hsien Su
Pittsburg State University
Bachelor of Science in Computer Science, 2006
Pittsburg State University
Master of Science in Technology, 2007
Pittsburg State University
Specialist in Education in Workforce Development and Education, 2009
University of Arkansas
Doctor of Education in Adult and Lifelong Learning, 2015

December 2020
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

_____
Qingyang Zhang, Ph.D.
Thesis Director

_____
Jyotishka Datta, Ph.D.
Committee Member

_____
Yuchun Du, Ph.D.
Committee Member

ABSTRACT

Pathways are the functional building blocks of complex diseases such as cancers. Pathway-level studies may provide insights on some important biological processes. Gene set test is an important tool to study the differential expression of a gene set between two groups, e.g., cancer vs normal. The differential expression of a gene set could be due to the difference in mean, variability, or both. However, most existing gene set tests only target the mean difference but overlook other types of differential expression. In this thesis, we propose to use the recently developed distance correlation for gene set testing. To assess the distance correlation test, simulation studies under different settings are conducted for a comprehensive comparison with the popular Hotelling's T^2 test and rotation gene set test (ROAST). The three gene set tests are also applied to two real datasets for further comparisons. Based on our simulation studies and real data applications, it is found that the distance correlation test has overall better statistical performance than Hotelling's T^2 test and ROAST test, especially for detecting the difference in variability.

This thesis begins with introductions to the problem of gene set testing, and then introduces the prevailing Hotelling's T^2 test and ROAST test. Chapter 2 is a detailed review of the concepts and properties of distance correlation. The results from simulation studies and real data applications were summarized in Chapters 3 and 4 respectively. In Chapter 5, we conclude the thesis with some discussion and future perspectives.

ACKNOWLEDGEMENTS

This thesis would not have been completed without the guidance from my committee, the knowledge taught by the Statistics and Analytics (STAN) Program, the encouragement of my friends, and the support of my family. First, I would like to give my immense appreciation and gratitude to my committee chair, Dr. Qingyang Zhang, for his full support and guidance. He repeatedly met with me to discuss the best direction for my thesis and spent endless hours reviewing my thesis each step of the way. I would also like to express my deep appreciation for my committee members, Dr. Jyotishka Datta and Dr. Yuchun Du, for their guidance and encouragement.

I want to give my special thanks to Dr. Giovanni Petris, Dr. Avishek Chakraborty, and Dr. John Robert Tipton, faculty in the Statistics and Analytics Program. I very much enjoyed learning from those professors. The knowledge I gained from their classes has given me the capability to study independently and work on the research project for this thesis.

My sincere gratitude goes to Dr. Kit Kacirek and Dr. Greg Belcher, who were the advisors in my Ed.D. program and Ed.S. program respectively, for their encouragement and mentorship. They encouraged me to pursue the STAN program when I expressed my interest while I was a student in their program. I truly appreciate the relationship with these two great professors.

Finally, I would like to give my special thanks to my family for their unflinching support. I am intensely grateful for the selfless sacrifices of my family. I would not have been able to successfully complete this degree without the support of my family.

TABLE OF CONTENTS

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

Chapter 1

Introduction

A gene set is a collection of genes that are a priori co-regulated or functionally related (Hejblum, Skinner, & Thiébaut, 2015) and a biological pathway can be defined as a sequence of interactions among molecules in a cell to govern a certain product or a change in a cell (Wikipedia, n.d.). Pathway information provides the facts of biological processes at molecular level (Cerami et al., 2011). For example, the cell-cycle pathway regulates an unreversed and crucial process of cell division. The life of a cell involves two stages that are interphase and M phase (Casem, 2016). M phase embraces all the steps occurred in mitotic cell division and interphase has G1, S, and G2 stages representing every other aspect of a life of cell. Cell cycle pathway is controlled by two classes of proteins known as cyclin-dependent kinases (Cdks) and cyclins. A diverse set of Cdks and cyclins rules each of the stages of the cell cycle. For instance, activation of Cdk2 by cyclin E controls the transition from G1 to S phase. However, the transition from G2 to M phase is controlled by the binding of Cdk1 and cyclin B. Cdk activity will be disabled when the cell has successfully transitioned from one phase to the next phase by destruction of the corresponding cyclin. There are three checkpoints in the process (Bio-Connect, n.d.):

(1) G1 checkpoint: determining if a cell will enter the cell division process

(2) G2 checkpoint: determining if the cell will enter into mitosis

(3) metaphase: ensuring proper chromosome alignment prior to cell division.

If a cell fails to meet the requirements of each checkpoint, it will lead the cell to halt cell cycle progression to next phase. The checkpoints are not often functional in cancer. This will result in genomic instability that is feature of malignant cells.

Gene set test is an important tool for evaluating differential expression of genes representing pathways or other biologically interpretable processes (Wu & Smyth, 2012). Goeman and Bühlmann (2007) classified gene set tests from two aspects: (1) the type of the null hypothesis and (2) the calculation of the p-value. By different null hypotheses, the tests can be classified into competitive and self-contained tests. Competitive gene set tests evaluate the differential expression of the selected genes relative to all other genes, to name a few, the gene set enrichment analysis (GSEA) proposed by Subramanian et al. (2005) and improved GSEA proposed by Efron and Tibshirani (2007). Self-contained gene set tests focus on the gene set or pathway of interest without reference to other genes, for instance, the global test (Goeman et al., 2004), ANCOVA-based approach (Mansmann & Meister, 2005), and the test proposed by Tomfohr, Lu, and Kepler (2005). Suppose G is the gene set of interest and G^c is the complement of G, the null hypothesis for competitive gene set tests can be stated as:

"H_0^{comp}: The genes in G are at most as often differentially expressed as the genes in G^c"

(Goeman and Bühlmann, 2007, p. 981),

and the null hypothesis for self-contained gene set tests can be stated as:

"H_0^{self}: No genes in G are differentially expressed" (Goeman and Bühlmann, 2007, p. 981).

By the method of p-value calculation, gene set tests can be classified into gene sampling methods and subject sampling methods. In gene sampling methods, p-values can be calculated for the gene set on a distribution where the gene is the sampling unit whereas subject sampling methods take subject as the sampling unit. The sampling units in both methods are assumed to be independent and identically distributed. Specifically, p-values can be evaluated by permuting genes in gene sampling methods and permuting subjects in subject sampling methods.

1.1 Purpose of this thesis

Existing gene set tests rely on several key assumptions such as normality and homogeneity of variance, to examine the differential expression of the gene set of interest by comparing the mean vectors. However, the differential expression of a gene set can be in many other forms such as variability difference. The goal of this thesis is to use an existing dependence measure which is capable of detecting differences in both mean and variability of a gene set without any parametric assumption. To validate the performance of distance correlation in gene set testing, two commonly used gene set tests including Hotelling's T^2 test and rotation gene set test (ROAST) (Wu, et al., 2010) are used in the comparison.

The thesis is structured as follows: A review of these two tests is provided in the sections 1.2 and 1.3. The review of distance correlation is provided in Chapter 2. The simulation studies are presented in Chapter 3, and real data applications are given in Chapter 4. Chapter 5 discusses and concludes the thesis

1.2 Hotelling's T^2 test

In 1931, Hotelling proposed the T^2 statistic in his paper entitled "The Generalization of Student's Ratio". Hotelling's T^2 test is a multivariate generation of Student's t test. Hotelling's T^2 test can be used for one-sample and two-sample cases. In this section, we review the concept of Hotelling's T^2 test by contrasting with univariate t-tests.

1.2.1 One-sample test

In the univariate case, suppose a random variable $x \sim N(\mu, \sigma^2)$. For a sample with n subjects, the t statistic can be defined as:

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}},$$

where $\bar{x}$ is the sample mean and s is the sample standard deviation. The t statistic follows the t_{n-1} distribution. One application of the t statistic is one-sample t test:

$$H_0: \mu = \mu_0,$$

where μ_0 is the proposed mean. Under H_0 is true, the test statistic can be defined as:

$$T = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}}. \tag{1.1}$$

It can be shown that T follows a t_{n-1} distribution. Now, we consider the multivariate case.

Suppose a random vector $X_{p\times 1} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \sim MVN\left(\mu_{p\times 1}, \Sigma_{p\times p}\right)$. There is a sample with n subjects:

$$X^{(1)} = \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1p} \end{bmatrix}, X^{(2)} = \begin{bmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2p} \end{bmatrix}, \cdots, X^{(n)} = \begin{bmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{np} \end{bmatrix}$$

For testing $H_0: \mu = \mu_0$, under the H_0 is true, the Hotelling's T^2 statistic in the one-sample case is analogous to the square of T given in (1.1):

$$HT^2 = (\bar{X} - \mu_0)^T \left(\frac{\hat{\Sigma}}{n}\right)^{-1} (\bar{X} - \mu_0),$$

where (1) $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X^{(i)}$ is the sample mean vector,

(2) μ_0 is the proposed population mean vector, and

(3) $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}\left(X^{(i)} - \bar{X}\right)\left(X^{(i)} - \bar{X}\right)^T$ is the sample dispersion matrix by maximum likelihood estimation (Anderson, 2003).

Under H_0 is true, the test statistic can be defined as:

$$T = \frac{n-p}{(n-1)p} HT^2.$$

The test statistic T follows $F_{p,n-p}$ distribution.

1.2.2 Two-sample test

In the univariate case, suppose a random variable x can be categorized by a factor $y = \{1, 2\}$. Let x_1 represent $x|y = 1$, x_2 represent $x|y = 2$, and they follow normal distributions with a common variance:

(1) $x_1 \sim N(\mu_1, \sigma^2)$,

(2) $x_2 \sim N(\mu_2, \sigma^2)$.

Suppose a sample with n subjects includes n_1 subjects from the first population and n_2 subjects from the second population. We are interested in if

$$H_0: \mu_1 = \mu_2.$$

Under H_0 is true, the test statistic can be defined as:

$$T = \frac{\bar{x}_1 - \bar{x}_2}{s_{pooled}\sqrt{\frac{n_1+n_2}{n_1 n_2}}}, \tag{1.2}$$

where (1) $\bar{x}_1$ and $\bar{x}_2$ are the sample means of x when $y = 1$ and $y = 2$ respectively,

(2) $s_{pooled} = \sqrt{\frac{ss_1 + ss_2}{n_1 + n_2 - 2}}$ is the pooled standard deviation (ss_1 and ss_2 are the sums of squares of x when $y = 1$ and $y = 2$ respectively).

This test statistic T follows the $t_{n_1+n_2-2}$ distribution. Now, we consider a multivariate case. Suppose a random vector $X_{p\times 1}$ can be categorized by a factor $y = \{1, 2\}$. Let X_1 represent $X|y = 1$, X_2 represent $X|y = 2$, and they follow two multivariate normal distributions with a common dispersion matrix:

(1) $X_1 \sim MVN(\mu_1, \Sigma_{p\times p})$,

(2) $X_2 \sim MVN(\mu_2, \Sigma_{p\times p})$.

Suppose a sample with n subjects includes n_1 subjects from the first population and n_2 subjects from the second population. For testing H_0: $\mu_1 = \mu_2$, under the H_0 is true, the Hotelling's T^2 in the two-sample test is analogous to the square of T given in (1.2):

$$HT^2 = \frac{n_1 n_2}{n_1 + n_2}(\bar{X}_1 - \bar{X}_2)^T \hat{\Sigma}_{pooled}^{-1}(\bar{X}_1 - \bar{X}_2),$$

where (1) $\bar{X}_1 = \frac{1}{n_1}\sum_{i=1}^{n_1} X^{(i)} \quad \forall X|y = 1,$

(2) $\bar{X}_2 = \frac{1}{n_2}\sum_{j=1}^{n_2} X^{(j)} \quad \forall X|y = 2,$

(3) $\hat{\Sigma}_{pooled}$

$$= \frac{1}{n_1 + n_2 - 2}\left(\sum_{i=1}^{n_1}\left(X_1^{(i)} - \bar{X}_1\right)\left(X_1^{(i)} - \bar{X}_1\right)^T + \sum_{j=1}^{n_2}\left(X_2^{(j)} - \bar{X}_2\right)\left(X_2^{(j)} - \bar{X}_2\right)^T\right)$$

is the pooled sample dispersion matrix (Anderson, 2003).

Under H_0 is true, the test statistic can be defined as:

$$T = \frac{n - p - 1}{(n-2)p} HT^2 \text{ (Izenman, 2008)}.$$

The test statistic T follows the $F_{p,n-p-1}$ distribution.

1.3 Rotation gene set test

Gene set tests based on permutation of probes for computing p-values assume that genes are independent. However, this assumption is unrealistic. Wu et al. (2010) proposed ROAST gene set test that allows for genewise correlation by using rotation which is a Monte Carlo technology for multivariate regression. In this section, we review the concept of ROAST test.

1.3.1 Statistical model

Suppose we have the expression data on G probes in each of n RNA samples. There are $p-1$ different treatments associated with the samples. Let y_{gi} be the log_2-expression value for the i^{th} sample of probe g and $y_g = [y_{g1} \quad y_{g2} \quad \cdots \quad y_{gn}]^T$ is a vector of expression values of probe g for n samples. Assumptions for ROAST are listed below (Wu et al., 2010):

(1) The $y = [y_1 \quad y_2 \quad \cdots \quad y_g \quad \cdots \quad y_G]^T$ follows a multivariate normal distribution with unknown correlations between probes.

(2) An experiment is assumed a linear model:

$$E(y_g) = X\alpha_g,$$

where X is a $n \times p$ design matrix of full column rank to indicate how the treatment factors are assigned to RNA samples and $\alpha_g = [\alpha_{g1} \quad \alpha_{g2} \quad \cdots \quad \alpha_{gj} \quad \cdots \quad \alpha_{gp}]^T$ is an unknown coefficient vector with a length of p. A coefficient α_{gj} represents the $(j-1)^{th}$ treatment effect or difference associated with probe g.

(3) The variance of y_g is assumed:

$$Var(y_g) = W^{-1}\sigma_g^2,$$

where W is a positive definite matrix of weight and σ_g^2 is the unknown probewise variance.

(4) The probewise variance σ_g^2 is assumed that it follows an inverse-χ^2 distribution:

$$\frac{1}{\sigma_g^2} \sim \frac{1}{s_0^2}\chi_{d_0}^2,$$

where s_0^2 is the prior variance represented typical variability and d_0 is the prior degrees of freedom used to control how consistent the variability is across probes.

1.3.2 Probe level test

For a probe level test, suppose we are interested in a contrast of coefficients (Wu et al., 2010):

$$\beta_g = c^T \alpha_g = [c_1 \quad c_2 \quad \cdots \quad c_p] \begin{bmatrix} \alpha_{g1} \\ \alpha_{g2} \\ \vdots \\ \alpha_{gp} \end{bmatrix} = \sum_{j=1}^{p} c_j \alpha_{gj} .$$

To find whether the β_g is nonzero, we state the null hypothesis as:

$$H_0: \beta_g = 0.$$

The test statistic t_g follows a t distribution with degrees of freedom $d = n - p$ under the null hypothesis:

$$t_g = \frac{\hat{\beta}_g}{s_g \sqrt{v}},$$

where $\hat{\beta}_g = c^T \hat{\alpha}_g = \sum_{j=1}^{p} c_j \hat{\alpha}_{gj}$ is the least squares estimator of β_g, s_g is the residual standard deviation for probe g, and $v = c^T (X^T W X)^{-1} c$ is an unscaled standard deviation of $\hat{\beta}_g$. An amended and superior test was derived by using the studies of Wright and Simon (2003) and Smyth (2004) to calculate the posterior variance $\tilde{s}_g^2$ as:

$$\tilde{s}_g^2 = \frac{d_0 s_0^2 + d s_g^2}{d_0 + d}.$$

Then, the moderated test statistic $\tilde{t}_g$ follows a t distribution with degrees of freedom $d_0 + d$ under the null hypothesis:

$$\tilde{t}_g = \frac{\hat{\beta}_g}{\tilde{s}_g \sqrt{v}}.$$

The moderated test statistic $\tilde{t}_g$ can be transformed to an equivalent standard normal random variables z_g as:

$$z_g = F^{-1}(F_t \tilde{t}_g),$$

where F and F_t are the cumulative distribution functions of standard normal and $t_{df=d_0+d}$ distribution respectively.

1.3.3 Gene set test

Suppose S is the set of indices of the probes in the gene set of interest. We can state the null hypothesis as (Wu et al., 2010):

$$H_0: \beta_g = 0 \; \forall g \in S.$$

The alternative hypothesis can be any one of three different statements listed in Table 1.1 based on one-tailed or two-tailed test.

Table 1.1
Three Different Alternative Hypotheses of ROAST

Type of H_1	Statement
H_{up}	$\beta_g > 0$ for at least one $g \in S$
H_{dwon}	$\beta_g < 0$ for at least one $g \in S$
H_{mixed}	$\begin{cases} \beta_g \neq 0 \text{ for at least one } g \in S \text{ or} \\ \text{genes can change in mixed (up or down) directions} \end{cases}$

Let a_g be a weight for probe g and $A = \sum_{g \in S} |a_g|$. Wu et al. (2010) proposed following different summary statistics calculated in term of the z_g:

(1) All genes in the gene set S are differentially expressed by a similar amount:

The test statistics for testing the directional hypotheses H_{up} or H_{dwon} can be given by:

$$T_{mean} = \frac{\sum_{g \in S} a_g z_g}{A}.$$

The test statistic for testing a non-directional hypothesis H_{mixed} the can be obtained by:

$$T_{mean} = \frac{\sum_{g \in S} |a_g z_g|}{A}.$$

(2) Only a few genes in the set S are differentially expressed or some log-fold-changes are much larger than other:

In this case, mean of the squared genewise statistics can better detect differentially expressed genes. The test statistic for testing hypothesis H_{mixed} is defined as:

$$T_{msq} = \frac{\sum_{g \in S} |a_g| z_g^2}{A}.$$

The test statistic for test hypothesis H_{up} is defined as:

$$T_{msq} = \frac{\sum_{a_g z_g > 0} |a_g| z_g^2}{A} \ \forall g \in S.$$

The test statistic for test hypothesis H_{down} is defined as:

$$T_{msq} = \frac{\sum_{a_g z_g < 0} |a_g| z_g^2}{A} \ \forall g \in S.$$

(3) Around a half of genes in the set S are differentially expressed:

In this case, mean-50 statistics can sensitively notice differentially expressed genes. Let $h = \lceil (m+1)/2 \rceil$, where m is the number of genes in the gene set S. The test statistic for testing hypothesis H_{mixed} is defined as:

$$T_{mean50} = \text{the mean of the } h \text{ largest } |a_g z_g| \text{ values.}$$

The test statistic for test hypothesis H_{up} is defined as:

$$T_{mean50} = \text{the mean of the } h \text{ largest } a_g z_g \text{ values.}$$

The test statistic for test hypothesis H_{down} is defined as:

$$T_{msq} = \text{the mean of the } h \text{ samllest } a_g z_g \text{ values.}$$

(4) Floor-mean statistic:

This statistic is motivated by the max-mean statistic proposed by Efron and Tibshirani (2007). The floor-mean statistic works alike to the mean-50 statistic. However, the computation of floor-mean statistic is faster than mean-50. For hypotheses H_{up} and H_{down}, the floored genewise statistics are $f_g = max(z_g, 0)$ and $f_g = min(a_g z_g, 0)$ respectively and their test statistic is defined as:

$$T_{floormean} = \frac{\sum_{g \in S} a_g f_g}{A}.$$

For the hypothesis H_{mixed}, the floored genewise statistic is $f_g = max(|z_g|, 0.67)$ and the test statistic is defined as:

$$T_{floormean} = \frac{\sum_{g \in S} |a_g f_g|}{A}.$$

1.3.4 p-values

Since the correlation between probes is unknown, the distribution of test statistic is unknown. Goeman and Bühlmann (2007) stated that the p-values derived from the methods with an assumption of independence can greatly understate the true p-values. ROAST does not permute samples because permutation needs a large number of replicate samples, cannot test general linear model hypotheses, and assumes that samples are identically distributed and exchangeable. Instead, ROAST utilizes the concepts of rotation tests studied by Langsrud (2005). The p-values is defined as:

$$p\text{-value} = \frac{b + 1}{B + 1},$$

where b is the number that yield a rotation statistic at least as extreme as that observed and B is the total number of rotations.

Chapter 2

Methodology

In this chapter, we review the concept and statistical properties of distance correlation proposed by Székely, Rizzo, and Bakirov in 2007 and derive the distance correlation between a binary variable and a continuous random vector. To begin with, we introduce the notion of characteristic function.

2.1 Characteristic function

If X is a random variable, the cumulative distribution function (CDF) of X is defined as:

$$F_X(x) = P(X \leq x).$$

The CDF contains all the information of the distribution of X. The moment generating function (MGF) of X is defined as:

$$m_X(t) = E(e^{tX}) \qquad t \in \mathbb{R},$$

and also provides information of the distribution of X. One of the major theoretical drawbacks is that MGF may not exist. The problem can be solved by involving imaginary number, $i = \sqrt{-1}$, to make the characteristic function such as (Evans & Rosenthal, 2010):

$$\begin{aligned} C_X(t) &= E\left(e^{itX}\right) \qquad t \in \mathbb{R} \\ &= E\big(cos(tX)\big) + iE\big(sin(tX)\big) \qquad t \in \mathbb{R}. \end{aligned}$$

Let X be a continuous variable with a probability density function $f_X(x)$, the characteristic function of X is:

$$C_X(t) = \int_{-\infty}^{\infty} e^{itx} f_X(x) dx\,.$$

If $X_1, X_2, \cdots, X_p$ are independent random variables, then

$$C_{X_1, X_2, \cdots, X_p}(t) = C_{X_1}(t) C_{X_2}(t) \cdots C_{X_p}(t).$$

Unlike moment generating functions, characteristic functions always exist (Blitzstein & Hwang, 2015). Characteristic functions will be used to define distance correlation in the next section.

2.2 Distance correlation

Distance correlation proposed by Székely, Rizzo, and Bakirov (2007) is an innovative measure of true dependence between random vectors X and Y with arbitrary dimensions. Distance covariance $\mathcal{V}$ and distance correlation $\mathcal{R}$ are analogous to product-moment covariance σ^2 and correlation ρ. The range of distance correlation is $0 \leq \mathcal{R}(X, Y) \leq 1$ and $\mathcal{R}(X, Y) = 0$ if and only if X and Y are independent. These are different from the prevailing product-moment correlation that $-1 \leq \rho \leq 1$ and $\rho = 0$ only indicates that the two random variables are linearly uncorrelated. The notations defined here will be used in illustration of the concepts of distance correlation later.

(1) X is a random vector in a space $\mathbb{R}^p$ where p is a positive integer.

(2) Y is a random vector in a space $\mathbb{R}^q$ where q is a positive integer.

(3) f_X and f_Y denote the characteristic functions of X and Y respectively.

(4) $f_{X,Y}$ denotes the joint characteristic function of X and Y.

(5) $|x|_p$ denotes the Euclidean norm of a vector x in a space $\mathbb{R}^p$. $|x|_p = |x|$ when $p = 1$

(6) $\|\gamma\|_w^2$ denotes the weighted L_2 norm for a complex function γ defined on $\mathbb{R}^p \times \mathbb{R}^q$.

(6) $\mathcal{V}(X, Y)$ denotes the population distance covariance (dCov) between X and Y.

(7) $\mathcal{V}(X)$ denotes the population distance variance (dVar) of X.

(8) $\mathcal{R}(X, Y)$ denotes the population distance correlation (dCor) between X and Y.

(9) $\mathcal{V}_n(X, Y)$ denotes the sample distance covariance (dCov_n) between X and Y.

(10) $\mathcal{V}_n(X)$ denotes the sample distance variance (dVar_n) of X.

(11) $\mathcal{R}_n(X, Y)$ denotes the sample distance correlation (dCor_n) between X and Y.

(10) $\tilde{\mathcal{V}}_n^2(X, Y)$ denotes the unbiased estimator of squared population distance covariance (dCov^2) between X and Y.

(11) $\mathcal{V}_n^*(X, Y)$ denotes the modified distance covariance statistic between X and Y.

(12) $\mathcal{R}_n^*(X, Y)$ denotes the modified distance correlation statistic between X and Y.

2.2.1 Definition of distance covariance

In their seminal work, Székely, Rizzo, and Bakirov (2007) introduced the distance covariance between X and Y with finite first moments:

$$\mathcal{V}^2(X, Y) = \left\|f_{X,Y}(t, s) - f_X(t)f_Y(s)\right\|^2 = \frac{1}{c_p c_q}\int_{\mathbb{R}^{p+q}} \frac{\left|f_{X,Y}(t, s) - f_X(t)f_Y(s)\right|^2}{|t|_p^{1+p}|s|_q^{1+q}}dtds,$$

where $c_p = \frac{\pi^{\left(\frac{1+p}{2}\right)}}{\Gamma\left(\frac{1+p}{2}\right)}$, $c_q = \frac{\pi^{\left(\frac{1+q}{2}\right)}}{\Gamma\left(\frac{1+q}{2}\right)}$, and $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt \ \forall x > 0$ is the complete gamma function. Similarly, distance variance of X denoted by $\mathcal{V}(X)$ is defined as the square root of:

$$\mathcal{V}^2(X) = \mathcal{V}^2(X, X) = \left\|f_{X,X}(t, s) - f_X(t)f_X(s)\right\|^2.$$

Székely et al (2007) also derive a second definition of distance correlation using inter-point distance (e.g., Euclidean distance) and proved its equivalency to the original definition:

$$\begin{aligned}\mathcal{V}^2(X, Y) &= E\left(|X_1 - X_2|_p|Y_1 - Y_2|_q\right) + E\left(|X_1 - X_2|_p\right)E\left(|Y_1 - Y_2|_q\right) - \\ &\quad 2E\left(|X_1 - X_2|_p|Y_1 - Y_3|_q\right) \\ &= \text{dCov}^2(X, Y) \\ &= \text{Cov}\left(|X_1 - X_2|_p, |Y_1 - Y_2|_q\right) - 2\text{Cov}\left(|X_1 - X_2|_p, |Y_1 - Y_3|_q\right), \qquad (2.1)\end{aligned}$$

where (X_1, Y_1), (X_2, Y_2), and (X_3, Y_3) are independent copies of (X, Y). The detailed proof is provided in Appendix A. In this thesis, we considered a special case of distance correlation where a random vector X following any multivariate distribution and a random variable

$Y \sim Bernoulli(\pi)$ where $Y = 0$, 1 for any two categories and $\pi = P(Y = 1)$. Then, the original formula of the squared distance covariance can be simplified as:

$$\mathcal{V}^2(X, Y)$$

$$= E\big(|X_1 - X_2|_p |Y_1 - Y_2|_q\big) + E\big(|X_1 - X_2|_p\big)E\big(|Y_1 - Y_2|_q\big) - 2E\big(|X_1 - X_2|_p |Y_1 - Y_3|_q\big)$$

$$= 2d_{00}(-\pi^4 + 3\pi^3 - 4\pi^2 + 3\pi - 1) + 2d_{11}(-\pi^4 + \pi^3 - \pi^2) +$$

$$2d_{01}(2\pi^4 - 4\pi^3 + 3\pi^2 - \pi), \tag{2.2}$$

where $d_{00} = E\big(|X_1 - X_2|_p | Y_1 = 0, Y_2 = 0\big)$, $d_{11} = E\big(|X_1 - X_2|_p | Y_1 = 1, Y_2 = 1\big)$, and $d_{01} = E\big(|X_1 - X_2|_p | Y_1 = 0, Y_2 = 1\big) = E\big(|X_1 - X_2|_p | Y_1 = 1, Y_2 = 0\big)$. A detailed proof is provided in Appendix B.

2.2.2 Definition of distance correlation

Székely, Rizzo, and Bakirov (2007) defined the distance correlation between X and Y with finite first moments is the nonnegative number $\mathcal{R}(X, Y)$ as:

$$\mathcal{R}^2(X, Y) = \begin{cases} \dfrac{\mathcal{V}^2(X, Y)}{\sqrt{\mathcal{V}^2(X)\mathcal{V}^2(Y)}}, & \mathcal{V}^2(X)\mathcal{V}^2(Y) > 0; \\ 0, & \mathcal{V}^2(X)\mathcal{V}^2(Y) = 0. \end{cases}$$

2.2.3 Estimated distance covariate and distance correlation

For a random sample of size n, $(X, Y) = \{(X_k, Y_k): k = 1, 2, 3, \cdots, n\}$ from a joint distribution of random vectors X in a space $\mathbb{R}^p$ and Y in a space $\mathbb{R}^q$, Székely, Rizzo, and Bakirov (2007) defined the distance dependence statistics as following:

$$X_{n\times p} = \begin{array}{c} \\ \mathbf{1} \\ \mathbf{2} \\ \vdots \\ \boldsymbol{n} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{1} & \mathbf{2} & \cdots & \boldsymbol{p} \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \end{array},$$

$$(X_1)_{p\times 1} = \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1p} \end{bmatrix}, (X_2)_{p\times 1} = \begin{bmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2p} \end{bmatrix}, \cdots, (X_n)_{p\times 1} = \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1p} \end{bmatrix},$$

$$Y_{n\times q} = \begin{array}{c} \\ \mathbf{1} \\ \mathbf{2} \\ \vdots \\ \boldsymbol{n} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{1} & \mathbf{2} & \cdots & \boldsymbol{q} \end{array} \\ \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1q} \\ y_{21} & y_{22} & \cdots & y_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nq} \end{bmatrix} \end{array},$$

$$(Y_1)_{q\times 1} = \begin{bmatrix} y_{11} \\ y_{12} \\ \vdots \\ y_{1q} \end{bmatrix}, (Y_2)_{q\times 1} = \begin{bmatrix} y_{21} \\ y_{22} \\ \vdots \\ y_{2q} \end{bmatrix}, \cdots, (Y_n)_{q\times 1} = \begin{bmatrix} y_{11} \\ y_{12} \\ \vdots \\ y_{1q} \end{bmatrix}.$$

Let a_{kl} and b_{kl} where $k, l = 1, 2, \cdots, n$ represent Euclidean distances between any two observations of (X, Y). Then, a_{kl} is given by:

$$a_{kl} = |X_k - X_l|_p = \left\| \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{kp} \end{bmatrix} - \begin{bmatrix} x_{l1} \\ x_{l2} \\ \vdots \\ x_{lp} \end{bmatrix} \right\|_p = \left\| \begin{bmatrix} x_{k1} - x_{l1} \\ x_{k2} - x_{l2} \\ \vdots \\ x_{kp} - x_{lp} \end{bmatrix} \right\|_p$$

$$= \sqrt{(x_{k1} - x_{l1})^2 + (x_{k2} - x_{l2})^2 + \cdots + \left(x_{kp} - x_{lp}\right)^2}$$

$$= \sqrt{\sum_{i=1}^{p} (x_{ki} - x_{li})^2}, \tag{2.3}$$

and similarly,

$$b_{kl} = \sqrt{\sum_{i=1}^{q} (y_{ki} - y_{li})^2}. \tag{2.4}$$

Let the distances be collected into two distance matrices. The two distance matrices are shown in Tables 2.1 and 2.2 below:

Table 2.1
Distance Matrix for the Random Vector X

$k\backslash l$	X_1	X_2	$\cdots$	X_n	Average
X_1	$a_{11}=0$	a_{12}	$\cdots$	a_{1n}	$\bar{a}_{1\cdot}=\left(\frac{1}{n}\right)\sum_{l=1}^{n}a_{1l}$
X_2	a_{21}	$a_{22}=0$	$\cdots$	a_{2n}	$\bar{a}_{2\cdot}=\left(\frac{1}{n}\right)\sum_{l=1}^{n}a_{2l}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_n	a_{n1}	a_{n2}	$\cdots$	$a_{nn}=0$	$\bar{a}_{n\cdot}=\left(\frac{1}{n}\right)\sum_{l=1}^{n}a_{nl}$
Average	$\bar{a}_{\cdot 1}=\frac{1}{n}\sum_{k=1}^{n}a_{k1}$	$\bar{a}_{\cdot 2}=\frac{1}{n}\sum_{k=1}^{n}a_{k2}$	$\cdots$	$\bar{a}_{\cdot n}=\frac{1}{n}\sum_{k=1}^{n}a_{kn}$	$\bar{a}_{\cdot\cdot}=\left(\frac{1}{n^2}\right)\sum_{k,l=1}^{n}a_{kl}$

Table 2.2
Distance Matrix for the Random Vector Y

$k\backslash l$	Y_1	Y_2	$\cdots$	Y_n	Average
Y_1	$b_{11}=0$	b_{12}	$\cdots$	b_{1n}	$\bar{b}_{1\cdot}=\left(\frac{1}{n}\right)\sum_{l=1}^{n}b_{1l}$
Y_2	b_{21}	$b_{22}=0$	$\cdots$	b_{2n}	$\bar{b}_{2\cdot}=\left(\frac{1}{n}\right)\sum_{l=1}^{n}b_{2l}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Y_n	b_{n1}	b_{n2}	$\cdots$	$b_{nn}=0$	$\bar{b}_{n\cdot}=\left(\frac{1}{n}\right)\sum_{l=1}^{n}b_{nl}$
Average	$\bar{b}_{\cdot 1}=\frac{1}{n}\sum_{k=1}^{n}b_{k1}$	$\bar{b}_{\cdot 2}=\frac{1}{n}\sum_{k=1}^{n}b_{k2}$	$\cdots$	$\bar{b}_{\cdot n}=\frac{1}{n}\sum_{k=1}^{n}b_{kn}$	$\bar{b}_{\cdot\cdot}=\left(\frac{1}{n^2}\right)\sum_{k,l=1}^{n}b_{kl}$

To make double centered distance matrices from the distance matrices above, double centered elements can be calculated by:

$$A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}$$

$$B_{kl} = b_{kl} - \bar{b}_{k\cdot} - \bar{b}_{\cdot l} + \bar{b}_{\cdot\cdot}$$

In these two double centered distance matrices, row means, column means, and the grand means all equal zero. The relationships among A_{kl}, a_{kl}, $\bar{a}_{k\cdot}$, $\bar{a}_{\cdot l}$, $\bar{a}_{\cdot\cdot}$ and B_{kl}, b_{kl}, $\bar{b}_{k\cdot}$, $\bar{b}_{\cdot l}$, $\bar{b}_{\cdot\cdot}$ are shown in Tables 2.3 and 2.4 below:

Table 2.3

Relationship among A_{kl}, a_{kl}, $\bar{a}_{k\cdot}$, $\bar{a}_{\cdot l}$, and $\bar{a}_{\cdot\cdot}$ in the Distance Matrix for the Random Vector X

$k\backslash l$	X_1	X_2	$\cdots$	X_l	$\cdots$	X_n	Average
X_1	a_{11}	a_{12}	$\cdots$	a_{1l}	$\cdots$	a_{1n}	$\bar{a}_{1\cdot}$
X_2	a_{21}	a_{22}	$\cdots$	a_{2l}	$\cdots$	a_{2n}	$\bar{a}_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
X_k	a_{k1}	a_{k2}	$\cdots$	a_{kl}	$\cdots$	a_{kn}	$\bar{a}_{k\cdot}$ (−)
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$ → A_{kl}
X_n	a_{n1}	a_{n2}	$\cdots$	a_{nl}	$\cdots$	a_{nn}	$\bar{a}_{n\cdot}$ (+)
Average	$\bar{a}_{\cdot 1}$	$\bar{a}_{\cdot 2}$	$\cdots$	$\bar{a}_{\cdot l}$ (−)	$\cdots$	$\bar{a}_{\cdot n}$	$\bar{a}_{\cdot\cdot}$

Table 2.4

Relationship among B_{kl}, b_{kl}, $\bar{b}_{k\cdot}$, $\bar{b}_{\cdot l}$, and $\bar{b}_{\cdot\cdot}$ in the Distance Matrix for the Random Vector Y

$k\backslash l$	Y_1	Y_2	$\cdots$	Y_l	$\cdots$	Y_n	Average
Y_1	b_{11}	b_{12}	$\cdots$	b_{1l}	$\cdots$	b_{1n}	$\bar{b}_{1\cdot}$
Y_2	b_{21}	b_{22}	$\cdots$	b_{2l}	$\cdots$	b_{2n}	$\bar{b}_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Y_k	b_{k1}	b_{k2}	$\cdots$	b_{kl}	$\cdots$	b_{kn}	$\bar{b}_{k\cdot}$ (−)
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$ → B_{kl}
Y_n	b_{n1}	b_{n2}	$\cdots$	b_{nl}	$\cdots$	b_{nn}	$\bar{b}_{n\cdot}$ (+)
Average	$\bar{b}_{\cdot 1}$	$\bar{b}_{\cdot 2}$	$\cdots$	$\bar{b}_{\cdot l}$ (−)	$\cdots$	$\bar{b}_{\cdot n}$	$\bar{b}_{\cdot\cdot}$

There were two issues of distance covariance/correlation: (1) $\mathcal{R}_n^2(X, Y)$ goes to 1 when p and q go infinity even though X and Y are independent for any sample size n and (2) the difference of double centered distance matrices is usually not a double centered distance matrix of any sample. Székely and Rizzo in their later studies defined different versions of A_{kl} and B_{kl} to solve the problems (details are at the end of this chapter). The double centered distance matrices are shown in Tables 2.5 and 2.6 below:

Table 2.5
Double Centered Distance Matrix for the Random Vector X

$\boldsymbol{k\backslash l}$	$\boldsymbol{X_1}$	$\boldsymbol{X_2}$	$\cdots$	$\boldsymbol{X_l}$	$\cdots$	$\boldsymbol{X_n}$	**Average**
$\boldsymbol{X_1}$	A_{11}	A_{12}	$\cdots$	A_{1l}	$\cdots$	A_{1n}	$\bar{A}_{1\cdot} = 0$
$\boldsymbol{X_2}$	A_{21}	A_{22}	$\cdots$	A_{2l}	$\cdots$	A_{2n}	$\bar{A}_{2\cdot} = 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{X_k}$	A_{k1}	A_{k2}	$\cdots$	A_{kl}	$\cdots$	A_{kn}	$\bar{A}_{k\cdot} = 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\boldsymbol{X_n}$	A_{n1}	A_{n2}	$\cdots$	A_{nl}	$\cdots$	A_{nn}	$\bar{A}_{n\cdot} = 0$
Average	$\bar{A}_{\cdot 1} = 0$	$\bar{A}_{\cdot 2} = 0$	$\cdots$	$\bar{A}_{\cdot l} = 0$	$\cdots$	$\bar{A}_{\cdot n} = 0$	$\bar{A}_{\cdot\cdot} = 0$

Table 2.6
Double Centered Distance Matrix for the Random Vector Y

$\boldsymbol{k\backslash l}$	$\boldsymbol{Y_1}$	$\boldsymbol{Y_2}$	$\cdots$	$\boldsymbol{Y_l}$	$\cdots$	$\boldsymbol{Y_n}$	**Average**
$\boldsymbol{Y_1}$	B_{11}	B_{12}	$\cdots$	B_{1l}	$\cdots$	B_{1n}	$\bar{B}_{1\cdot} = 0$
$\boldsymbol{Y_2}$	B_{21}	B_{22}	$\cdots$	B_{2l}	$\cdots$	B_{2n}	$\bar{B}_{2\cdot} = 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{Y_k}$	B_{k1}	B_{k2}	$\cdots$	B_{kl}	$\cdots$	B_{kn}	$\bar{B}_{k\cdot} = 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\boldsymbol{Y_n}$	B_{n1}	B_{n2}	$\cdots$	B_{nl}	$\cdots$	B_{nn}	$\bar{B}_{n\cdot} = 0$
Average	$\bar{B}_{\cdot 1} = 0$	$\bar{B}_{\cdot 2} = 0$	$\cdots$	$\bar{B}_{\cdot l} = 0$	$\cdots$	$\bar{B}_{\cdot n} = 0$	$\bar{B}_{\cdot\cdot} = 0$

In the double centered distance matrix of X, $\bar{A}_{k\cdot} = 0$ for the k^{th} row, where $k = 1, 2, \cdots, n$. The detailed proof is provided in Appendix C. Similarly, we have $\bar{A}_{\cdot l} = 0$, $\bar{B}_{k\cdot} = 0$, and $\bar{B}_{\cdot l} = 0$ for $k, l = 1, 2, \cdots, n$. Therefore, the estimated nonnegative distance covariace $\mathcal{V}_n(X, Y)$ is equal to the square root of the average of elementwise products of these two double centered matrices:

$$\mathcal{V}_n^2(X, Y) = \left(\frac{1}{n^2}\right) \sum_{k,l=1}^{n} (A_{kl} B_{kl}) .$$

Analogously, an estimated nonnegative distance variance of X denoted by $\mathcal{V}_n(X)$ is given by the square root of:

$$\mathcal{V}_n^2(X) = \mathcal{V}_n^2(X, X) = \left(\frac{1}{n^2}\right) \sum_{k,l=1}^{n} A_{kl}^2 .$$

The empirical distance correlation $\mathcal{R}_n(X, Y)$ is defined as the square root of:

$$\mathcal{R}_n^2(X, Y) = \begin{cases} \dfrac{\mathcal{V}_n^2(X, Y)}{\sqrt{\mathcal{V}_n^2(X)\mathcal{V}_n^2(Y)}}, & \text{if } \mathcal{V}_n^2(X)\mathcal{V}_n^2(Y) > 0; \\ 0, & \text{if } \mathcal{V}_n^2(X)\mathcal{V}_n^2(Y) = 0. \end{cases}$$

2.2.4 Implementation of the distance covariance test

The distance covariance test is for testing multivariate independence. For small samples, a reference distribution for $n\mathcal{V}_n^2(X, Y)$ under independence through the observed sample is established since the distribution of $n\mathcal{V}_n^2(X, Y)$ is difficult to calculate (Székely, Rizzo, & Bakirov, 2007). This nonparametric test makes decision from permutation bootstrap with a user defined number of replicates. That is, the distribution for $n\mathcal{V}_n^2(X, Y)$ under independence is constructed by calculating replicates of $n\mathcal{V}_n^2(X, Y)$ under random permutations of the indices of the Y sample.

2.2.5 Modified distance covariance and distance correlation t-test

There is a problem with distance covariance. It is that for any fixed sample size n, $\mathcal{R}_n^2(X, Y)$ goes to 1 when p and q go infinity even though X and Y are independent (Székely & Rizzo, 2013). In the study, modified distance covariance and modified correlation were proposed for testing independence between two random vectors of arbitrary dimensions (possibly high-dimensions). Let A_{kl}^* and B_{kl}^* be modified versions of A_{kl} and B_{kl} and defined by:

$$A_{kl}^* = \begin{cases} \left(\frac{n}{n-1}\right)\left(A_{kl} - \frac{a_{kl}}{n}\right), \text{if } k \neq l; \\ \left(\frac{n}{n-1}\right)(\bar{a}_{k\cdot} - \bar{a}_{\cdot\cdot}), \quad \text{if } k = l, \end{cases}$$

$$B_{kl}^* = \begin{cases} \left(\frac{n}{n-1}\right)\left(B_{kl} - \frac{b_{kl}}{n}\right), \text{if } k \neq l; \\ \left(\frac{n}{n-1}\right)(\bar{b}_{k\cdot} - \bar{b}_{\cdot\cdot}), \quad \text{if } k = l, \end{cases}$$

such that $E(A_{kl}^*) = E(B_{kl}^*) = 0 \; \forall \; k, l$. The modified distance covariance statistic can be defined using A_{kl}^* and B_{kl}^*:

$$\mathcal{V}_n^*(X, Y) = \left(\frac{1}{n(n-3)}\right)\left(\sum_{k,l=1}^{n} A_{kl}^* B_{kl}^* - \left(\frac{n}{n-2}\right)\sum_{k=1}^{n} A_{kk}^* B_{kk}^*\right),$$

where $n \geq 3$. This modified distance covariance is an unbiased estimator of squared population distance covariance. Naturally, the modified distance correlation statistic can be defined by:

$$\mathcal{R}_n^*(X, Y) = \begin{cases} \dfrac{\mathcal{V}_n^*(X, Y)}{\sqrt{\mathcal{V}_n^*(X, X)\mathcal{V}_n^*(Y, Y)}}, \text{if } \mathcal{V}_n^*(X, X)\mathcal{V}_n^*(Y, Y) > 0; \\ 0, \qquad\qquad\qquad\qquad \text{if } \mathcal{V}_n^*(X, X)\mathcal{V}_n^*(Y, Y) = 0. \end{cases}$$

Finally, the test statistic can be defined by:

$$\mathcal{T}_n = \sqrt{\upsilon - 1}\,\frac{\mathcal{R}_n^*(X, Y)}{\sqrt{1 - \mathcal{R}_n^{*\,2}(X, Y)}},$$

where $v = \frac{n(n-3)}{2}$. If p and q go infinity, under the independence hypothesis, the test statistic $\mathcal{T}_n$ converges in distribution to Student's t with $v - 1$ degrees of freedom.

2.2.6 $\mathcal{U}$-centering and unbiased distance covariance

In this part, we discussed the aforementioned issue of double centered distance matrices. It is that the difference of double centered distance matrices is usually not a double centered distance matrix of any sample. To solve this problem, Székely and Rizzo (2014) in their study about partial distance correlation defined an alternate type of double centering in the Hilbert space named unbiased or $\mathcal{U}$-centering that formulates an unbiased estimator of squared population distance covariance. Let $\tilde{A}_{kl}$ and $\tilde{B}_{kl}$ be the (k, l)th entry of the $\mathcal{U}$-centered matrices $\tilde{A}$ and $\tilde{B}$ respectively. In the double centering, A_{kl} and B_{kl} have the property that the all rows and columns have zero sums. The $\mathcal{U}$-centering $\tilde{A}_{kl}$ and $\tilde{B}_{kl}$ inherits this property and has the additional property that all expectations are equal to zero, that is, $E(\tilde{A}_{kl}) = 0$ and $(\tilde{B}_{kl}) = 0$ $\forall k, l$. For $n > 2$, the definitions of $\tilde{A}_{kl}$ and $\tilde{B}_{kl}$ are defined by:

$$\tilde{A}_{kl} = \begin{cases} a_{kl} - \left(\frac{1}{n-2}\right)\sum_{h=1}^{n} a_{kh} - \left(\frac{1}{n-2}\right)\sum_{g=1}^{n} a_{gl} + \left(\frac{1}{(n-1)(n-2)}\right)\sum_{g,h=1}^{n} a_{gh}, & \text{if } k \neq l; \\ 0, & \text{if } k = l, \end{cases}$$

$$\tilde{B}_{kl} = \begin{cases} b_{kl} - \left(\frac{1}{n-2}\right)\sum_{h=1}^{n} b_{kh} - \left(\frac{1}{n-2}\right)\sum_{g=1}^{n} b_{gl} + \left(\frac{1}{(n-1)(n-2)}\right)\sum_{g,h=1}^{n} b_{gh}, & \text{if } k \neq l; \\ 0, & \text{if } k = l, \end{cases}$$

where a_{kl} and b_{kl} can be found in equations (2.3) and (2.4). If $E(|X| + |Y|) < \infty$, for $n > 3$, the unbiased estimator of squared population distance covariance is defined by:

$$\tilde{\mathcal{V}}_n^2(X, Y) = (\tilde{A} \cdot \tilde{B}) = \left(\frac{1}{n(n-3)}\right)\sum_{k \neq l} \tilde{A}_{kl}\tilde{B}_{kl}.$$

Chapter 3

Simulation Studies

Simulation studies were conducted to assess and compare type I error rates and statistical power of the distance correlation test, Hotelling's T^2 test, and ROAST test in different scenarios. The R packages for implementation include clusterGeneration v1.3.4, DescTools v0.99.30, energy v1.7-6, limma v3.42.0, and MASS v7.3-51.4. The purposes of the key functions used in the simulation studies are described below:

- The function rcorrmatrix(·) is from the R package clusterGeneration (Qiu & Joe, 2015). The simulations used it with an argument the pre-defined standard deviation vector to generate a random positive definite correlation matrix for constructing a positive definite dispersion matrix for a random vector.
- The function mvrnorm(·) is from the R package MASS (Ripley, Venables, Bates, Hornik, Gebhardt, & Firth, 2019). It was employed with arguments a pre-defined mean vector and a dispersion matrix to randomly generate data for a multivariate normal random vector X.
- The function dcor.test(·) is from the R package energy (Rizzo & Székely, 2019). It is a nonparametric test of multivariate independence. The p-value of this test is found through permutation bootstrap with a specified number of replicates. It was applied to test the independence between a random variable Y and a multivariate normal random vector X.
- The function roast(·) is from the R package limma (Smyth et al., 2019). It was originally proposed and implemented for gene set test but it can generally be applied to any variable set test. It was used with arguments data of a multivariate normal random vector X, design matrix comprised of intercept which was 1 and the random variable Y, and the option contrast of 2 for ROAST testing. This function provides four different p-values:

Down, Up, UporDown (two-sided), and Mixed. In this simulation studies, Mixed p-value was chosen to compare with the significance level α.

- The function HotellingsT2Test(·) is from the R package DescTools (Signorell et al., 2019). Hotelling's T^2 test is the multivariate generalization of the Student's t test. It was used with an argument the formula of $X \sim Y$ to test for a significant difference between the mean vectors of two multivariate datasets $X|Y = 0$ and $X|Y = 1$.

3.1 Simulation settings

In the simulation studies, X was a random vector of continuous type and Y was a dichotomous random variable. The distributions of X and Y are shown below:

$$(X|Y = i) \sim MVN_p(\mu_i, \Sigma_i)\ \forall i = 0 \text{ and } 1,$$

$$Y \sim Bernoulli(\pi),$$

where $Y = 0$ (normal), 1 (cancer) and $\pi = P(Y = 1)$. We generated the datasets under various settings of the mean vectors, dispersion matrices, and sample sizes varied from 40 to 300.

3.2 Hypotheses and significance level

There were three different hypothesis tests in the simulation studies. They included distance correlation test, Hotelling's T^2 test, and ROAST test. All the hypothesis tests were tested at the significance level $\alpha = 0.05$. The hypotheses for these three hypothesis tests are listed in Table 3.1 below:

Table 3.1
Hypotheses for Distance Correlation, Hotelling's T^2, and ROAST Tests

	Distance Correlation Test	Hotelling's T^2 Test	ROAST Test
H_0	$\mathcal{R} = 0$	$\mu_0 = \mu_1$	$\beta = 0$
H_1	$\mathcal{R} \neq 0$	$\mu_0 \neq \mu_1$	$\beta \neq 0$

3.3 Assessing type I error rates (simulation study I):

A type I error occurs when a true null hypothesis is rejected. A type I error rate is the probability that a null hypothesis is rejected when it is true:

$$P(\text{reject } H_0 | H_0 \text{ is true}).$$

One important property of distance correlation is that $\mathcal{R}(X, Y) = 0$ only if X and Y are independent. It implies that $P(X|Y = 0) = P(X|Y = 1) = P(X)$ if the null hypothesis is true. In other words, to make the null hypothesis be true, both $(X|Y = 0)$ and $(X|Y = 1)$ should follow an identical multivariate normal distribution $MVN_p(\mu, \Sigma)$ where μ and Σ are the common mean vector and the common dispersion matrix of $(X|Y = 0)$ and $(X|Y = 1)$. Similarly, this concept can be also applied to a Hotelling's T^2 test and ROAST test. In this part of simulation studies, the common mean vector μ was set as:

$$\mu^T = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0],$$

the common dispersion matrix Σ was calculated based on the common population standard deviation vector:

$$\sigma^T = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 0.5 \quad 1.5 \quad 2.5],$$

and the common population correlation matrix R which was randomly generated by the function rcorrmatrix$(\cdot)$. The random variable Y was set to follow the Bernoulli distribution with $\pi = 0.5$. Moreover, the number of replicates was set as 1000. In each replication, two new datasets of the random variable Y and the random vector of X were randomly generated respectively for distance correlation test, Hotelling's T^2 test, and ROAST test. The null hypothesis was rejected if a p-value was less than $\alpha = 0.05$. The numbers of times that a null hypotheses was rejected in distance correlation tests, Hotelling's T^2 tests, and ROAST tests within 1000 replicates were recorded respectively. A type I error rate can be obtained by:

$$\text{Type I error rate} = \frac{\text{The number of times that the } H_0 \text{ was rejected}}{1000}.$$

The whole process above was performed ten replicates for each different sample size. The average type I error rates and standard deviations for different sample sizes were summarized based on the ten replicates. The R source code can be found in Appendix D. It can be seen from Figure 3.1 that for all settings, the type I error rates are very close to the nominal level of 0.05 with a standard deviation less than 0.01 (see Table 6.1).

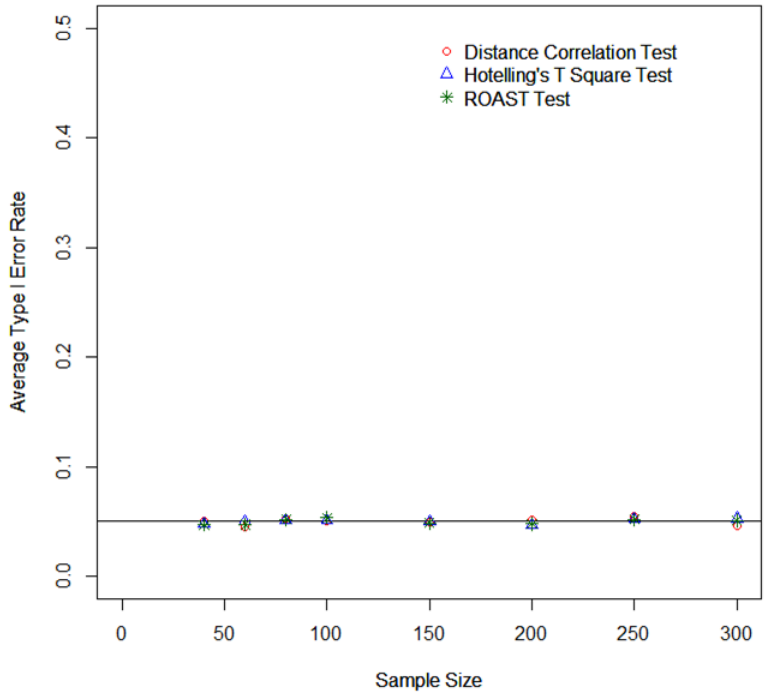

Figure 3.1. Average type I error rate versus sample size in different hypothesis tests.

3.4 Assessing statistical power

Power is the probability that a null hypothesis is rejected when it is false:

$$P(\text{reject } H_0 | H_0 \text{ is false}).$$

Considering a distance correlation test, the null hypothesis is true only if X and Y are independent. To make the null hypothesis be false, X and Y should be dependent each other. If X and Y are dependent, $P(X|Y = 0) \neq P(X)$ and $P(X|Y = 1) \neq P(X)$ that implies $P(X|Y = 0) \neq$

$P(X|Y=1)$. Therefore, let $(X|Y=0)$ and $(X|Y=1)$ follow different multivariate normal distributions $MVN_p(\mu_0, \Sigma_0)$ and $MVN_p(\mu_1, \Sigma_1)$ respectively. Similarly, the concept can be also applied to a Hotelling's T^2 test and ROAST test. The statistical power was assessed under four conditions based on the settings of mean vectors, standard deviation vectors, and correlation matrices. The four conditions are listed in Table 3.2.

Table 3.2
Settings of the Four Conditions for Assessing Powers

Condition	Mean	Standard Deviation	Correlation
1	$\mu_0 \neq \mu_1$	$\sigma_0 = \sigma_1$	$R_0 = R_1$
2	$\mu_0 = \mu_1$	$\sigma_0 \neq \sigma_1$	$R_0 \neq R_1$
3	$\mu_0 = \mu_1$	$\sigma_0 = \sigma_1$	$R_0 \neq R_1$
4	$\mu_0 \neq \mu_1$	$\sigma_0 \neq \sigma_1$	$R_0 \neq R_1$

The random variable Y was set to follow the Bernoulli distribution with $\pi = 0.5$. The number of replicates was set as 1000. New datasets for random variable Y and random vector X were randomly generated based on different settings of mean vectors, standard deviation vector, and correlation matrices for distance correlation test, Hotelling's T^2 test, and ROAST test in each replication. The null hypothesis was rejected if a p-value was less than $\alpha = 0.05$. The number of times that a null hypothesis was rejected in distance correlation tests, Hotelling's T^2 tests, and ROAST tests within 1000 replicates were recorded respectively. A statistical power can be calculated by:

$$\text{Power} = \frac{\text{The number of times that the } H_0 \text{ was rejected}}{1000}$$

The whole process above was executed ten replicates for each sample size. An average power and a standard deviation for each sample size were summarized based on the ten replicates.

3.4.1 Simulation study II

The setting of the simulation study II had the same standard deviation vector and the same correlation matrix but different mean vectors for the two groups of the random vector X categorized by the dichotomous random variable Y. The dispersion matrices Σ_0 and Σ_1 were constrained to be same. The common standard deviation vector was set as:

$$\sigma^T = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 0.5 \quad 1.5 \quad 2.5].$$

The common correlation matrix R was randomly generated by the rcorrmatrix$(\cdot)$ function. The common dispersion matrix Σ was calculated based on the common standard deviation vector σ and the common correlation matrix R. The μ_0 and μ_1 were set in three different cases listed in Table 3.3. The R problem and means and standard deviations of statistical power under different tests and sample sizes for this part of simulation studies can be found in Appendix E. The results under the first case showing in Figure 3.2 indicate that both Hotelling's T^2 test and ROAST test have means of statistical power around 1 with standard deviations around 0 (see Table 6.2) for all sample sizes. However, distance correlation test has means of statistical power below 0.8 with standard deviations around 0.01 (see Table 6.2) when sample sizes are less than 80 and means of statistical power around 1 with standard deviations below 0.01 (see Table 6.2) when sample sizes are 150 or above.

Table 3.3
Settings of μ_0 and μ_1 for Different Three Cases in the Simulation Study II

	μ_0^T	μ_1^T
Case 1	$[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$	$[1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$
Case 2	$[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$	$[0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5]$
Case 3	$[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$	$[0.5 \quad -0.5 \quad 0.5 \quad -0.5 \quad 0.5 \quad -0.5 \quad 0.5 \quad -0.5]$

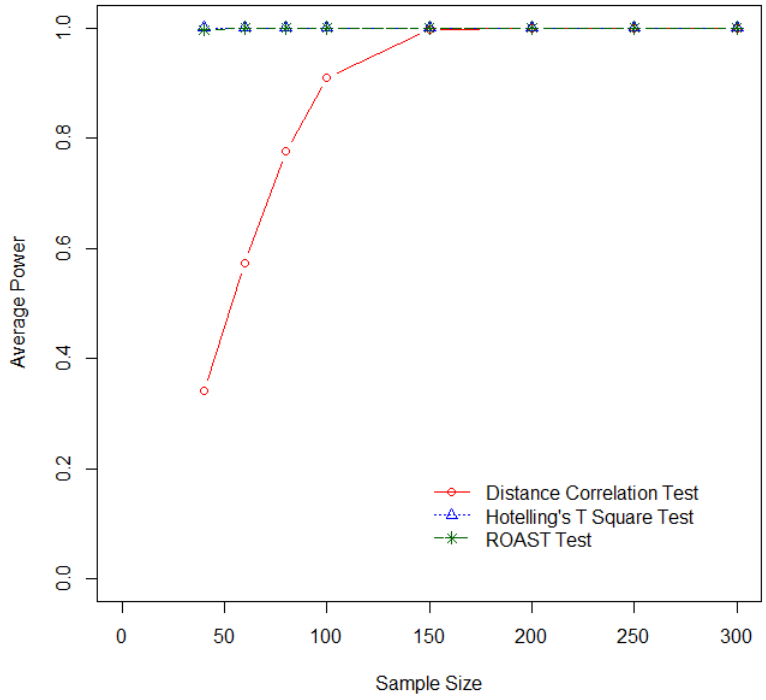

Figure 3.2. Average power versus sample size in different hypothesis tests under the first case in the simulation study II.

The results under the second and third cases showing in Figures 3.3 and 3.4 are similar. Hotelling's T^2 test has the best performance followed by ROAST test. The means of statistical power of Hotelling's T^2 test are around 1 with standard deviations around 0 (see Tables 6.3 and 6.4) for all sample sizes. The means of statistical power of ROAST test are under 0.8 with standard deviations around 0.01 (see Tables 6.3 and 6.4) when sample sizes below 80 and the means of statistical power around 1 with standard deviations around 0 (see Tables 6.3 and 6.4) when sample sizes are 150 or above. In distance correlation test, the mean of statistical power is getting larger when the sample size is increased. Specifically, distance correlation test has a mean of statistical power around 0.1 with a standard deviation around 0.01 when sample size is 40 and a mean of statistical power around 0.8 with a standard deviation around 0.01 when sample size becomes 300.

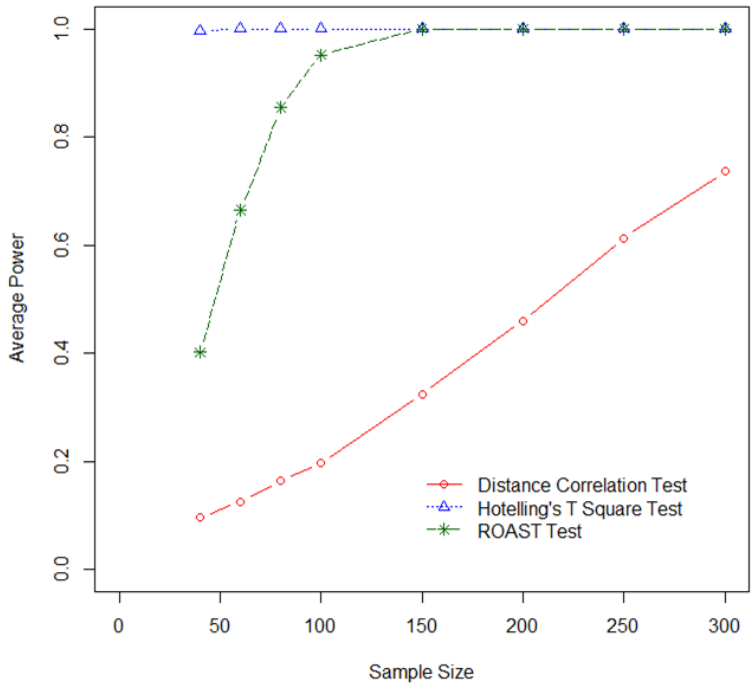

Figure 3.3. Average power versus sample size in different hypothesis tests under the second case in the simulation study II.

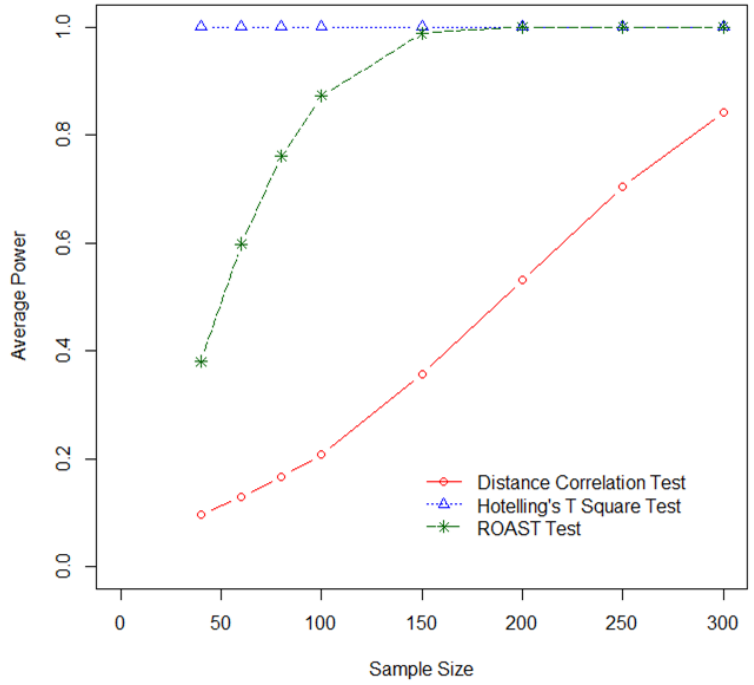

Figure 3.4. Average power versus sample size in different hypothesis tests under the third case in the simulation study II.

3.4.2 Simulation study III

The setting of the simulation study III had the same mean vector but different standard deviation vectors and different correlation matrices for the two groups of the random vector X categorized by the dichotomous random variable Y. In this simulation study, we set $\mu_0 = \mu_1$ and used two random correlation matrices. The common mean vector, standard deviation vectors were set in two different cases listed in Table 3.4. The R problem and means and standard deviations of statistical power under different tests and sample sizes for the simulation study III can be found in Appendix F. The results under the both cases showing in Figures 3.5 and 3.6 are similar. Both Hotelling's T^2 test and ROAST test have means of statistical power around 0 with standard deviations around 0 (see Tables 6.5 and 6.6) for all sample sizes. However, distance correlation test has means of statistical power under 0.8 with standard deviations around 0 (see Tables 6.5 and 6.6) when sample sizes are below 80 and means of statistical power around 1 with standard deviations around 0 when the sample sizes are 100 or above.

Table 3.4
Settings of μ_0, μ_1, σ_0, and σ_1 for Different Cases in the Simulation Study III

	Case 1	Case 2
$\mu_0^T = \mu_1^T$	$[0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$	$[0.5 \;\; -0.5 \;\; 0.5 \;\; -0.5 \;\; 0.5 \;\; -0.5 \;\; 0.5 \;\; -0.5]$
σ_0^T	$[1 \;\; 2 \;\; 3 \;\; 4 \;\; 5 \;\; 0.5 \;\; 1.5 \;\; 2.5]$	$[1 \;\; 2 \;\; 3 \;\; 4 \;\; 5 \;\; 0.5 \;\; 1.5 \;\; 2.5]$
σ_1^T	$[5 \;\; 4 \;\; 0.5 \;\; 3 \;\; 2.5 \;\; 1.5 \;\; 2 \;\; 1]$	$[5 \;\; 4 \;\; 0.5 \;\; 3 \;\; 2.5 \;\; 1.5 \;\; 2 \;\; 1]$

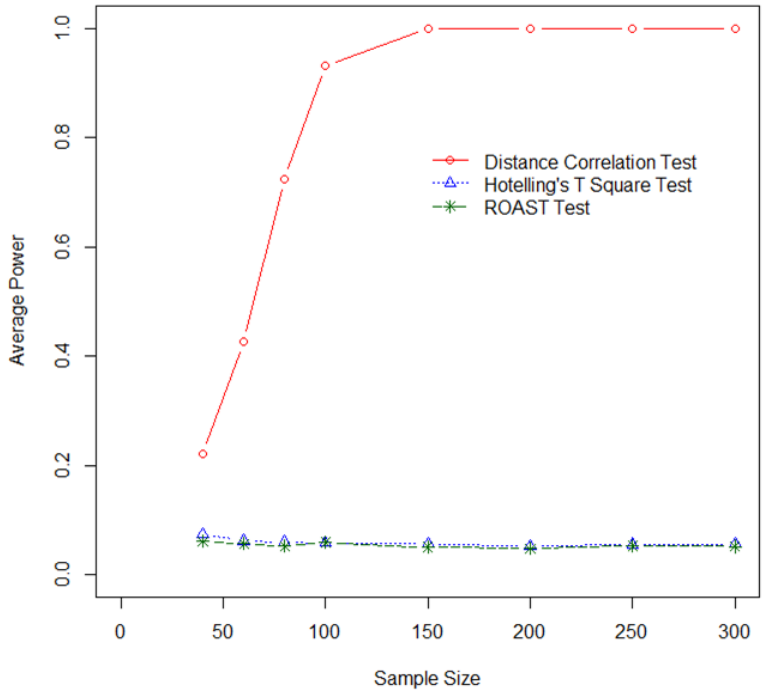

Figure 3.5. Average power versus sample size in different hypothesis tests under the first case in the simulation study III.

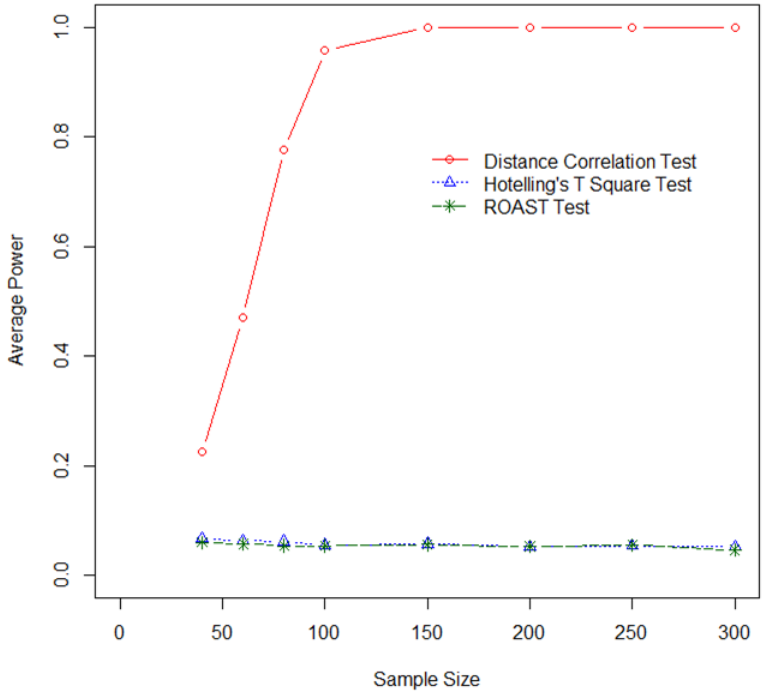

Figure 3.6. Average power versus sample size in different hypothesis tests under the second case in the simulation study III.

3.4.3 Simulation study IV

The setting of the simulation study IV had the same mean vector and standard deviation vector but different correlation matrices for the two groups of the random vector X categorized by the dichotomous random variable Y. We set $\mu_0 = \mu_1$ and $\Sigma_0 = \Sigma_1$. The common mean vector μ was set as:

$$\mu^T = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0].$$

The common standard deviation vector σ was set as:

$$\sigma^T = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 0.5 \quad 1.5 \quad 2.5].$$

Additionally, we used two random correlation matrices. The R problem and means and standard deviations of statistical power under different tests and sample sizes for this simulation study can be found in Appendix G. the results showing in Figure 3.7 indicate that both Hotelling's T^2 and ROASST tests have means of statistical power around 0 with standard deviations around 0 (see Table 6.7) for all sample sizes. However, distance correlation test has an s-shaped curve. It has means of statistical power above 0.8 with standard deviations around 0 (see Table 6.7) when the sample sizes are greater 200.

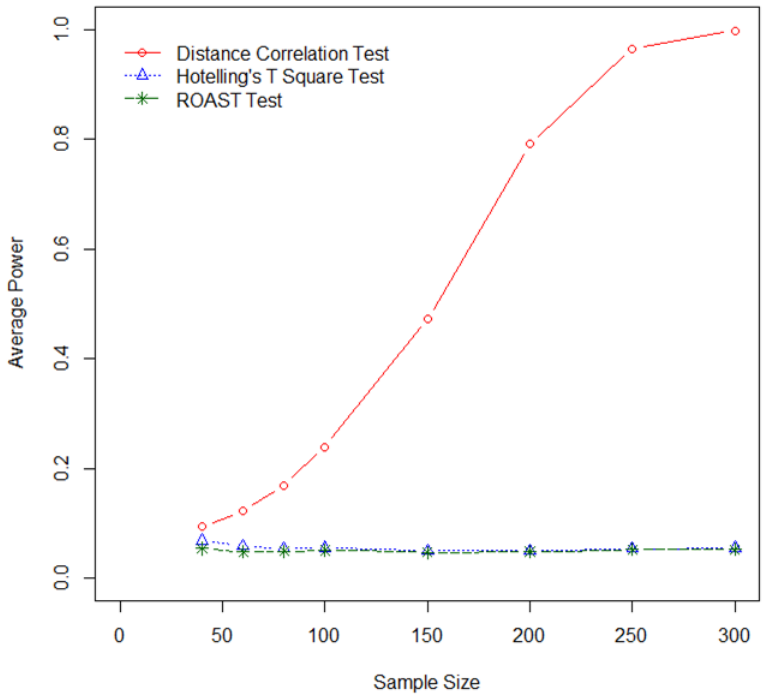

Figure 3.7. Average power versus sample size in different hypothesis tests in the simulation study IV.

3.4.4 Simulation study V

In the setting of the simulation study V, mean vectors, standard deviation vectors, and correlation matrices were different for the two groups of the random vector X categorized by the dichotomous random variable Y. The mean vectors and standard deviation vectors were set in three different cases listed in Table 3.5. The R program and means and standard deviations of statistical power under different tests and sample sizes for this simulation study can be found in Appendix G. The results under the first case showing in Figure 3.8 indicate that all three hypothesis test are similar. The lowest means of statistical power are around 0.75 with standard deviations around 0 (see Tables 6.8 – 6.10) for all three hypothesis tests when the sample size is 40. The means of statistical power are around 1 with standard deviations around 0 for all three hypothesis tests when the sample sizes are 60 or above. The results under both the second and third cases showing in Figures 3.9 and 3.10 indicate distance correlation has the best statistical

power followed by Hotelling's T^2 test for all sample sizes. In both cases, distance correlation test has means of statistical power above 0.9 with standard deviations around 0 (see Tables 6.9 and 6.10) when sample sizes are 80 or above. However, Hotelling's T^2 and ROAST tests have means of statistical power around 0.8 or above with standard deviations around 0 (see Tables 6.9 and 6.10) when sample sizes are 150 or above.

Table 3.5
Settings of μ_0, μ_1, σ_0, and σ_1 for Different Cases in the Simulation Study V

	Case 1	Case 2
μ_0^T	$[0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$	$[0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$
μ_1^T	$[1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1]$	$[0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5]$
σ_0^T	$[1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 0.5\ \ 1.5\ \ 2.5]$	$[1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 0.5\ \ 1.5\ \ 2.5]$
σ_1^T	$[5\ \ 4\ \ 0.5\ \ 3\ \ 2.5\ \ 1.5\ \ 2\ \ 1]$	$[5\ \ 4\ \ 0.5\ \ 3\ \ 2.5\ \ 1.5\ \ 2\ \ 1]$

Table 3.5 (Cont.)

	Case 3
μ_0^T	$[0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$
μ_1^T	$[0.5\ \ -0.5\ \ 0.5\ \ -0.5\ \ 0.5\ \ -0.5\ \ 0.5\ \ -0.5]$
σ_0^T	$[1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 0.5\ \ 1.5\ \ 2.5]$
σ_1^T	$[5\ \ 4\ \ 0.5\ \ 3\ \ 2.5\ \ 1.5\ \ 2\ \ 1]$

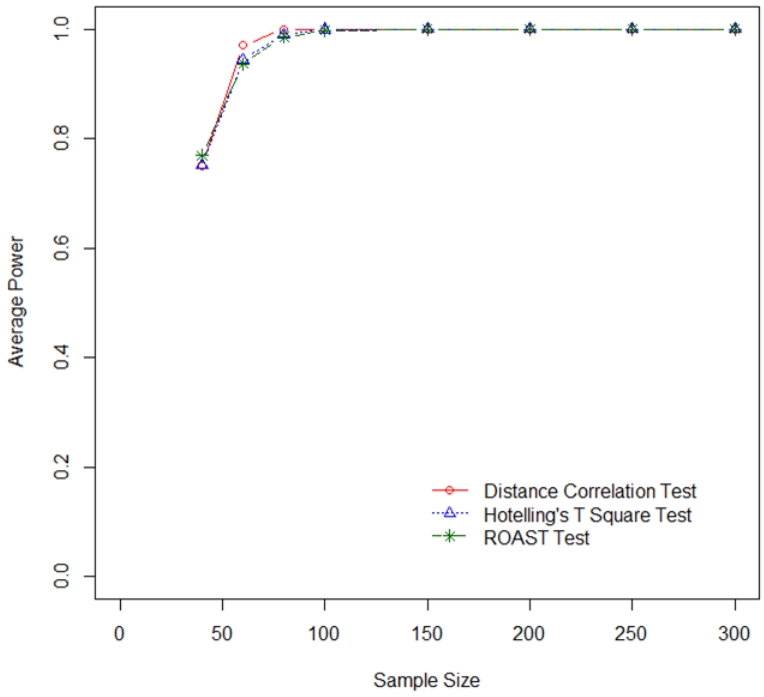

Figure 3.8. Average power versus sample size in different hypothesis tests under the first case in the simulation study V.

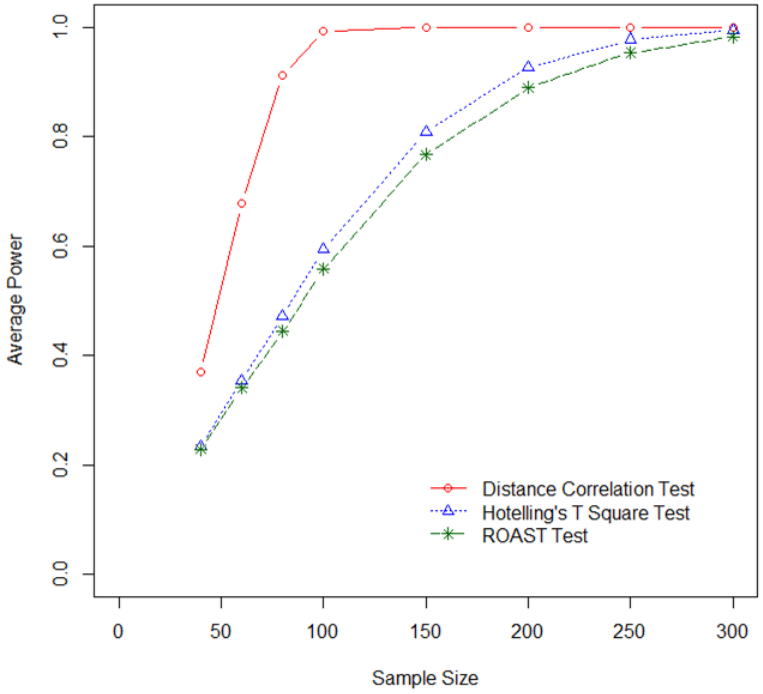

Figure 3.9. Average power versus sample size in different hypothesis tests under the second case in the simulation study V.

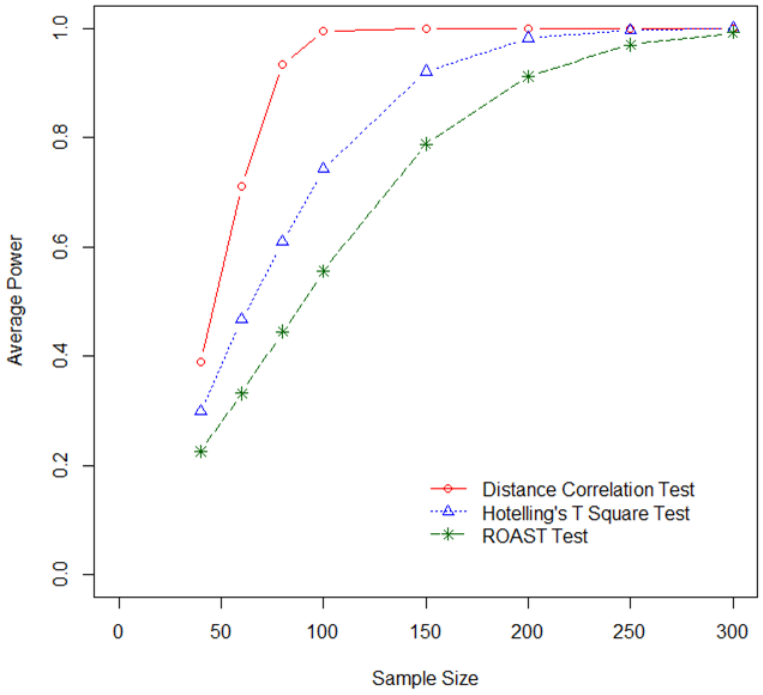

Figure 3.10. Average power versus sample size in different hypothesis tests under the third case in the simulation study V.

Chapter 4

Real Data Applications

In this chapter, we compared the three tests, namely, distance correlation test, Hotelling's T^2 test and ROAST test, on two real data applications. The first dataset is a RNA-seq dataset about cervical tumors and matched controls. It was used in the study by Witten, Tibshirani, Gu, Fire, and Lui (2010). The second dataset is phylogenetic microarray data matrix about microbiota in the human intestine. It was used in the study by Lahti, Salojärvi, Salonen, Scheffer, and Vos (2014). For each dataset, we employed the three methods to test independence between a variable set and a group variable. All the hypothesis testing were based on the significance level of 0.05.

4.1 First real data application

The dataset used in this application is an expression profile of 714 miRNAs from 58 samples including 29 cervical tumor samples and 29 normal controls (Witten, Tibshirani, Gu, Fire, & Lui, 2010) downloaded from Gynecologic Oncology Group Tissue Bank. An examination about how miRNA and the types of tissue are associated was conducted. The following notations will be used in illustration of this real data application:

(1) X_{tumor} and X_{normal} denote the expression data of miRNAs with a cervical tumor and a normal tissue respectively,

(2) $\bar{X}_{tumor}$ and s_{tumer} denote the mean and the standard deviation vectors of the expression data of miRNAs with a cervical tumor in the sample,

(3) $\bar{X}_{normal}$ and s_{normal} denote the mean and the sample standard deviation vectors of the expression data of miRNAs with a normal tissue in the sample.

4.1.1 Dataset summary

The expression level of each miRNA was measured by RNA-sequencing technique, where the abundance can be represented by the number of sort reads produced by the assay. To stabilize the variances, a natural log transformation was performed. All the counts were added by 1 to avoid log of zero. A mean difference $(\bar{X}_{normal} - \bar{X}_{tumor})$ and a ratio of standard deviations (s_{normal}/s_{tumer}) for each miRNA were used to express how the differences in mean and standard deviation between the normal and the tumor groups.

Figure 4.1 shows the distribution of mean differences between the normal and the tumor groups according to the log-scaled dataset with a normal curve of $N\left(mean\left((\bar{X}_{normal} - \bar{X}_{tumor})\right), sd(\bar{X}_{normal} - \bar{X}_{tumor})\right)$. This distribution is fairly symmetrical, skewness = 0.341.

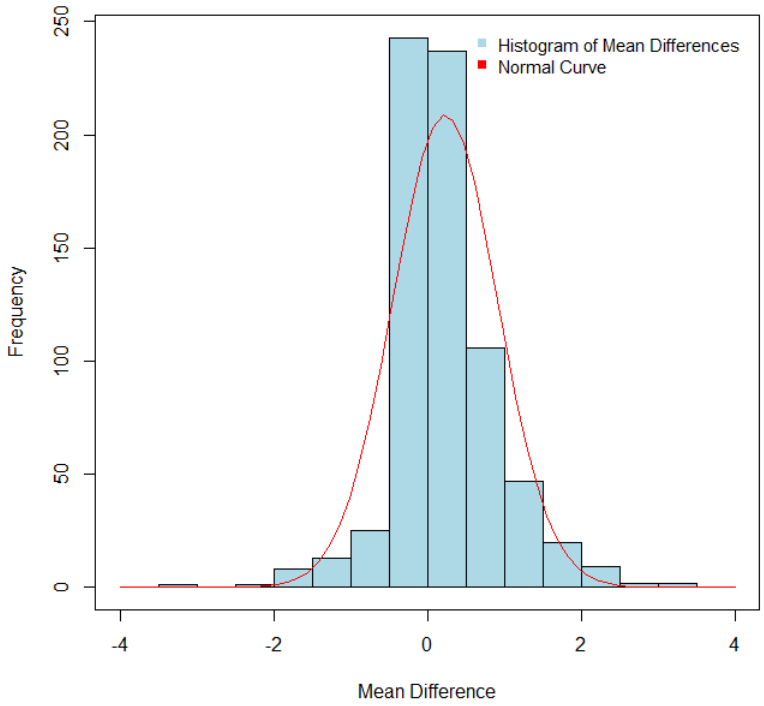

Figure 4.1. Distribution of mean differences between the normal and the tumor groups for each miRNA with a normal curve.

The distribution of standard deviation ratios of the normal and the tumor groups for each miRNA according to the log-scaled dataset is represented in Figure 4.2. Thirty miRNAs have a ratio of infinity and is excluded in Figure 4.2. That is caused by those miRNAs have all subjects with a zero expression in the tumor group. The distribution is positively skewed.

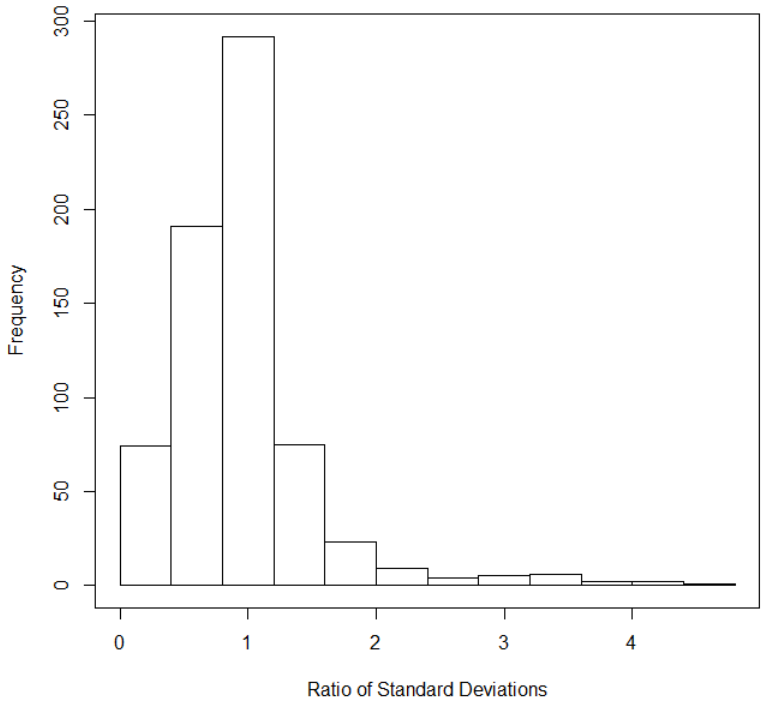

Figure 4.2. Distribution of standard deviation ratios between the normal and the tumor groups for each miRNA.

As we can see, some miRNAs have large differences in both mean and standard deviation between the normal and the tumor groups. This is a similar condition to the conditions in the simulation study V. We expect that the expression level of miRNAs will be associated with the two types of tissues in the population.

4.1.2 Preprocessing of raw data

The total count of each miRNA was calculated by summing all corresponding values of 58 subjects (tissues). The values of the total count were between 1 and 2253073. A set of sequence thresholds was set from 40 to 129000 with an interval of 20. Distance correlation test

and ROAST test were performed under each threshold. Hotelling's T^2 test is not appropriate for this dataset since the number of miRNAs is greater than the number of subjects. The thresholds were used to determine which miRNAs were included for distance correlation test and ROAST test. Any miRNAs were included in the random vector X for a specific threshold if the corresponding total count were greater than the specific threshold. The numbers of miRNAs included in a random vector X under different thresholds are represented in Figure 4.3. As we can see, more than 600 miRNAs were included when the thresholds were between 0 and 12000. Then, the numbers of miRNAs rapidly dropped for the rest of thresholds. The data of the random variable Y were set as a vector of size 59 with a value of 0 to represent the corresponding subject with a normal tissue and the other value of 1 to represent the corresponding subject with a tumor tissue. The R program for the first real data application can be found in Appendix I.

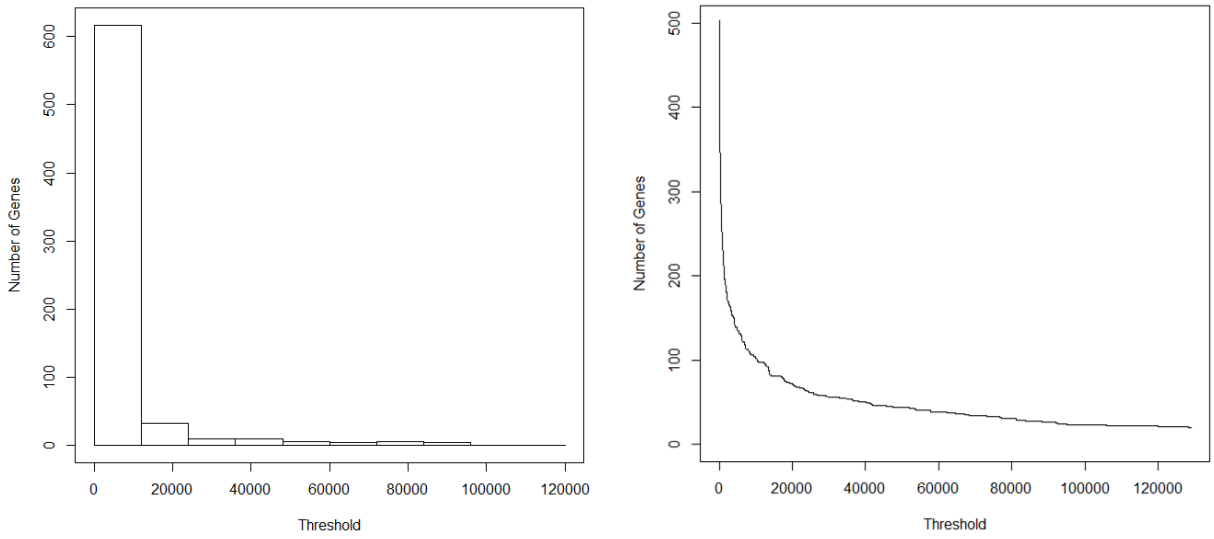

Figure 4.3. Histogram and scatterplot of different thresholds versus number of selected miRNAs.

4.1.3 Results

The distributions of p-values for distance correlation test and ROAST test under all difference thresholds were summarized in Figure 4.4. The null hypotheses were rejected in both distance correlation test and ROAST test under all thresholds. We conclude that at least one

transcript count of a miRNA is associated with the types of tissues, cervical cancer and normal, in the populations. However, in terms of p-values, distance correlation test is smaller and more concentrated than ROAST test. The descriptive statistics of the p-values are listed in Table 4.1.

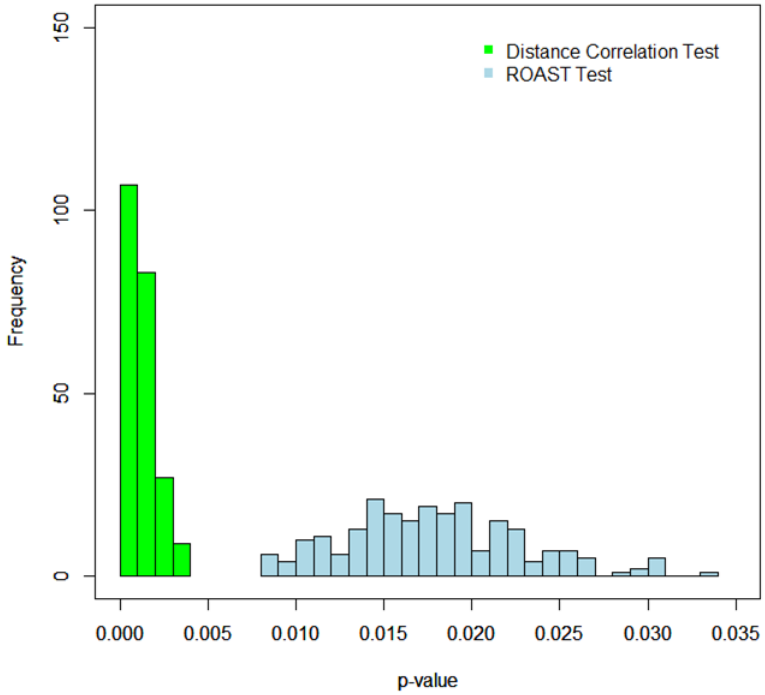

Figure 4.4. Distributions of p-value for distance correlation test and ROAST test under all different thresholds.

Table 4.1
Descriptive Statistics of p-values for Distance Correlation Test and ROAST Test under Different Thresholds

	Min	*Max*	*M*	*SD*
Distance Correlation	0.001	0.004	0.002	0.001
ROAST	0.008	0.034	0.018	0.005

4.2 Second real data application

The dataset used in this real data application including two files named HITChip.tab and Metadata.tab from the study titled "Tipping Elements in the Human Intestinal Ecosystem" by Lahti, Salojärvi, Salonen, Scheffer, and Vos (2014). The HITChip.tab encompasses HITChip

phylogenetic microarray data matrix with a dimension of 1172 samples by 130 genus-like groups related microbial communities in human intestine. Those data are continuous. The Metadata.tab holds the metadata for the samples in the HITChip data matrix. It includes 10 variables. The information of the variables is listed in Table 4.2.

Table 4.2
Information of the Variables in the Metadata File

Variable	Explanation
SampleID	Unique ample identified corresponding to the sample in HITChip data matrix
Age	Age in years
Sex	Male/Female
Nationality	At the level of geographic regions: US, Central Europe, Eastern Europe, South Europe, Scandinavia, UKIE
DNA _extraction_method	DNA extraction method: r: Repeated Bead Beating o: Other
ProjectID	Project identifier
Diversity	Shannon diversity index based on probe-level signals

Table 4.2 (Cont.)

Variable	Explanation
BMI_group	Standard body-mass index classification: underweight: < 18.5 lean: 18.5 – 25 overweight: 25 – 30 obese: 30 – 35 severe: 35 – 40 morbid obese 40 – 45 superobese: > 45
SubjectID	Subject identifier
Time	Time point from the baseline in months

4.2.1 Descriptive statistics of the preprocessed dataset

The range of the microarray data in HITChip.tab was between 32.29489 and 944063.8. To stabilize the variances, a natural log transformation was performed by taking a natural loge of the data in HITChip.tab directly. The second real data application examined whether HITChip phylogenetic microarray data is associated with age, sex, nationality, and BMI group respectively. The variables Age, Sex, Nationality, and BMI_group were re-categorized into four dichotomous random variables Ys. There was no missing value in HITChip.tab. However, there were some missing values indicated by NAs in Metadata.tab. The cases were eliminated from both HITChip.tab and Metadata.tab if the corresponding values were NA in the variables Age, Sex, Nationality, BMI_group respectively. The conditions of categorization and group sample sizes are listed in Table 4.3. The dataset of random vector X and the corresponding random variable Y were constructed based on and represented the re-categorized variables Age, Sex, Nationality, and BMI_group individually. The following notations will be used in illustration later:

(1) $X_{Y=0}$ and $X_{Y=1}$ denote the microarray data with $Y = 0$ and $Y = 1$ respectively.

(2) $\bar{X}_{Y=0}$ and $s_{Y=0}$ denote the mean and the standard deviation vectors of the microarray data with $Y = 0$ in the sample,

(3) $\bar{X}_{Y=1}$ and $s_{Y=1}$ denote the mean and the sample standard deviation vectors of the data with $Y = 1$ in the sample.

Then, distance correlation test, Hotelling's T^2 test, and ROAST test were applied to each pair of X and Y. The R program for the second real data application can be found in Appendix J.

Table 4.3
Conditions and Group Sample Sizes for Re-categorized Age, Sex, Nationality, and BMI_group

Variable	$Y = 0$		$Y = 1$	
	Condition	Sample Size	Condition	Sample Size
Age	≤ 40	415	otherwise	701
Sex	male	455	female	680
Nationality	US	44	otherwise	1096
BMI_group	lean	493	otherwise	573

The distributions of mean differences $(\bar{X}_{Y=0} - \bar{X}_{Y=1})$ and standard deviation ratios $(s_{Y=0}/s_{Y=1})$ between two groups of re-categorized variables Age, Sex, Nationality, and BMI _group respectively are exhibited in Figures 4.5 – 4.8. The skewnesses of those distributions are listed in Table 4.4. The distributions of mean differences are fairly symmetrical except the variable Sex that is highly and positively skewed. The distributions of standard deviation ratios tend to be highly and positively skewed for all variables. As we can see, some genus-like groups have large differences in both mean and standard deviation between the two groups of the re-categorized variables Age, Sex, Nationality, and BMI_group respectively. Furthermore, the sample sizes are large in this application. This is also a similar condition to the conditions in the simulation study

V. We expect that the null hypotheses will be rejected in distance correlation test, Hotelling's T^2 test, and ROAST test.

Table 4.4
Skewnesses of the Distributions of Mean Differences and Standard Deviation Ratios for Re-categorized Age, Sex, Nationality, and BMI_group

	Variables			
	Age	Sex	Nationality	BMI_group
Mean Difference $(\bar{X}_{Y=0} - \bar{X}_{Y=1})$	0.030	1.672	-0.559	-0.115
Standard Deviation Ratio $(s_{Y=0}/s_{Y=1})$	0.808	1.899	1.696	1.076

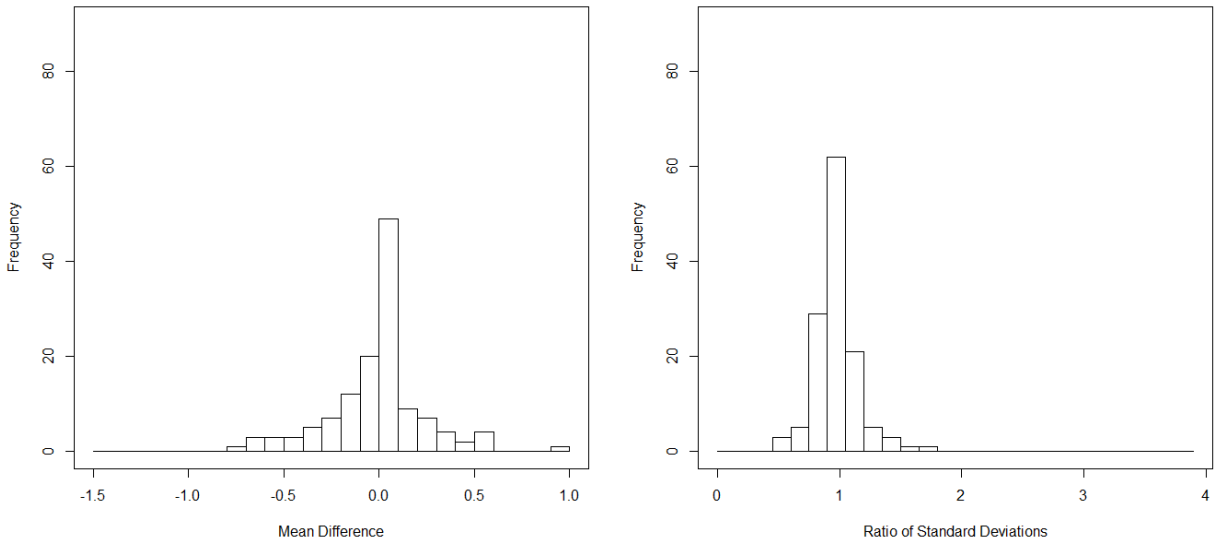

Figure 4.5. Distributions of mean differences and standard deviation ratios between the two groups of Age for all genus-like groups.

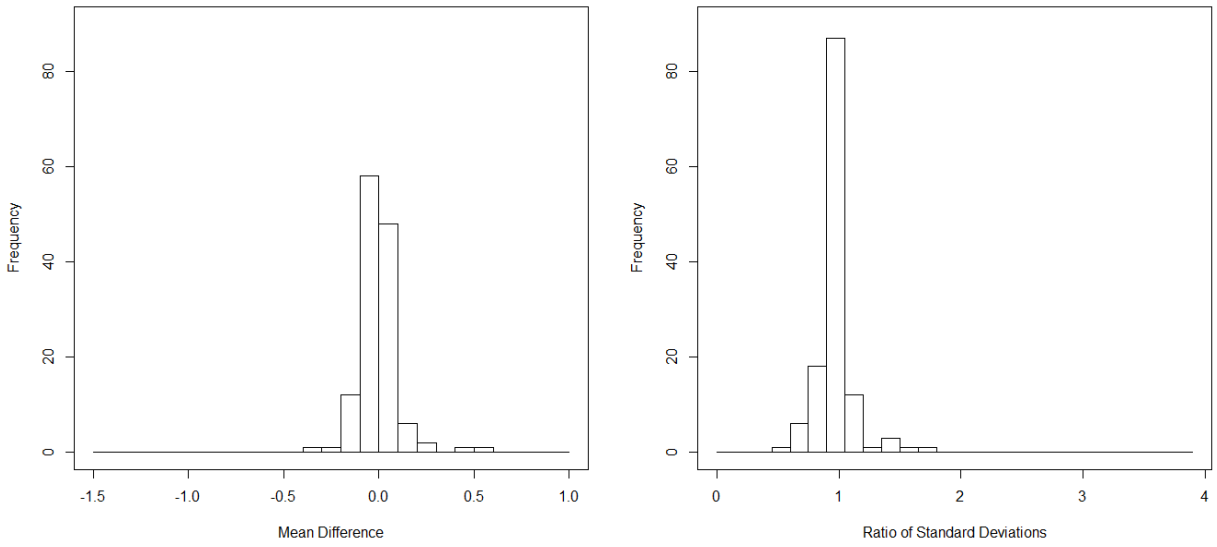

Figure 4.6. Distributions of mean differences and standard deviation ratios between the two groups of Sex for all genus-like groups.

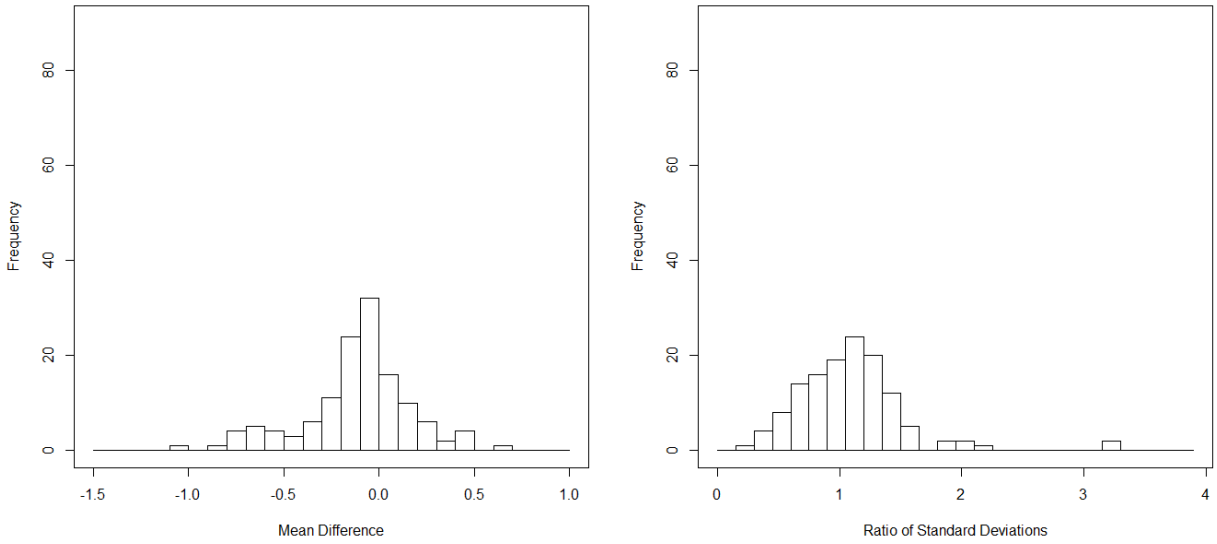

Figure 4.7. Distributions of mean differences and standard deviation ratios between the two groups of Nationality for all genus-like groups.

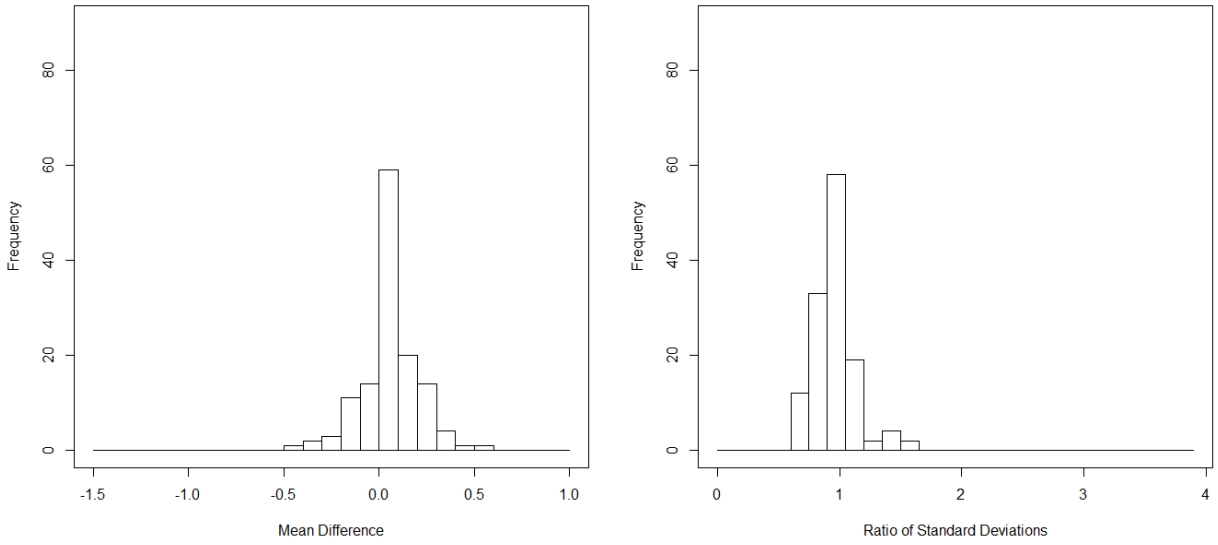

Figure 4.8. Distributions of mean differences and standard deviation ratios between the two groups of BMI_group for all genus-like groups.

4.2.2 Results

According to the *p*-values listed in Table 4.4, the null hypotheses were rejected by distance correlation test, Hotelling's T^2 test, and ROAST test for factors, the re-categorized variables age, sex, nationality, and BMI group, respectively. We can conclude that at least one genus-like group is associated with the re-categorized variables age, sex, nationality, and BMI group respectively in the populations. In other words, the multivariate distributions of the microarray data categorized by factors re-categorized age, sex, nationality, and BMI group respectively are different.

Table 4.5

The *p-values for Distance Correlation Test, Hotelling's T^2 Test, and ROAST Test in Age, Sex, Nationality, and BMI Group*

	Distance Correlation Test	Hotelling's T^2 test	ROAST
Age	0.001	0.000	0.001
Sex	0.001	0.000	0.016
Nationality	0.001	0.000	0.001
BMI Group	0.001	0.000	0.001

Chapter 5

Discussion and Conclusions

This chapter concludes the thesis and discusses some advantages and shortcomings of the three gene set tests being compared. In addition, we discuss some possible extensions and future directions.

5.1 Summary

Many statistical methods have been recently developed to test the differential expression of a gene set. However, most of the gene set tests emphasis on detecting mean differences instead of distributional differences. To this end, we proposed to use a novel dependence measure, namely distance correlation, for gene set testing because it targets not only the mean difference but also other forms of difference. To validate the distance correlation test, simulation studies under different settings were conducted for a comprehensive comparison with two popular multivariate tests including Hotelling's T^2 test and ROAST test. Furthermore, these three tests were applied to two real data sets.

5.2 Discussion

Both Hotelling's T^2 test and ROAST test detect a differential expression of a gene set based on mean vectors. The primary drawback of these two tests is that they cannot detect the difference in variability. If a null hypothesis is failed to reject in Hotelling's T^2 test and ROAST test, it tells us that the gene set has the same population mean vector between two groups. However, these two populations might or might not have a difference in variability. If a null hypothesis is rejected by Hotelling's T^2 test and ROAST test, it tells us that the gene set has a difference in population mean vector but it is unknown that whether these two populations have the same variability or not. The other drawback of Hotelling's T^2 test and ROAST test is that

they rely on several assumptions that we mentioned in Chapter 1. In contrast with Hotelling's T^2 test and ROAST test, distance correlation test detects a differential expression of a gene set based on their distributions. Distance correlation test does not require any parametric assumptions. If a null hypothesis fails to be rejected in distance correlation test, it tells us the two groups of the gene set have the same population distribution. The major drawback of distance correlation test is that it does not tell us the differential expression of a gene set is caused by population mean vectors or population variabilities if the null hypothesis is rejected in distance correlation test.

Based on the settings of mean vectors, standard deviation vectors, and correlation matrices, simulation studies were conducted under one condition for assessing type I error rates and four conditions with nine cases for assessing powers. The results can be summarized as:

(1) The differential expression of a gene set is purely caused by differences in mean:

Hotelling's T^2 test has the best powers that are close to 1 for all specified sample sizes. Then, it is followed by ROAST test and distance correlation test. Both ROAST test and distance correlation test have a larger power if the sample size is increased. Specifically for distance correlation test, the powers are greater than 0.7 when the sample sizes are greater than 250.

(2) The differential expression of a gene set is caused by differences in correlation with/without differences in standard deviation:

Both Hotelling's T^2 test and ROAST test have powers that are close to 0 for all specified sample sizes. Distance correlation test has the best powers for all specified sample sizes. The power of distance correlation test is increased if the sample size is increased. Overall, the powers of distance correlation test are greater than 0.8 when the sample sizes are greater than 200.

(3) The differential expression of a gene set is caused by differences in mean, standard deviation, and correlation:

When the mean difference is large, the three tests are similar. All three tests have powers greater than 0.7 for all specified sample sizes. When the mean difference is small, distance correlation test has the best powers for all specified sample sizes. Then, it is followed by Hotelling's T^2 test and ROAST test. All three tests have a greater power when the sample size is increased. Specifically for distance correlation test, the powers are greater than 0.7 when the sample sizes are greater 60.

The two real data applications have similar conditions to the simulation study V and the findings from these two real data applications support the results from the simulation study V.

5.3 Conclusions

According to our simulation studies, the distance correlation test works better than Hotelling's T^2 test and ROAST test on detecting a differential expression of a gene set caused by differences in variability. However, when a null hypothesis is rejected in a distance correlation test, it does not tell us that the differential expression of the gene set is caused by mean differences, variability differences, or both mean and variability differences.

Both Hotelling's T^2 test and ROAST test can only detect the differential expression of a gene set due to mean differences. If the null hypothesis is failed to reject in Hotelling's T^2 test or ROAST test, distance correlation test can be used for further examinations.

5.4 Future work

In this thesis, we focus on a dichotomous response Y in distance correlation test. In fact, this can be extended to any number of categories, nominal or ordinal. For example, the random variable Y can be the BMI_group in the second real data application, which has eight categories

including underweight, lean, overweight, obese, severe, morbid obese, and superobese. Distance correlation test can be directly applied to such multi-category response simply by defining the dummy variables.

In this thesis, we chose distance correlation for gene set test. However, many other novel correlation measures can be used. For instance, Zhu, Xu, Li, and Zhong (2017) proposed the projection correlation, which is a measure of dependence between two random vectors. Projection correlation is equal to zero if and only if the two random vectors are independent. Furthermore, projection correlation does not require moment restriction for (X, Y), which is more flexible distance correlation. A possible extension is to apply those similar correlation measures to the problem of gene set testing.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*. New Jersey: John Wiley.

Bio-Connect. (n.d.). Cell Cycle Pathway. Retrieved May 19, 2020 from https://www.bio-connect.nl/cell-cycle-pathway/cnt/page/4814

Blitzstein, J. K., & Hwang, J. (2015). *Introduction to probability*.

Casem, M. L. (2016). *Case Studies in Cell Biology*. Cambridge, MA: Academic Press

Cerami, E. G., Gross, B. E., Demir, E., Rodchenkov, I., Babur, Ö., Anwar, N., ..., Sander, C. (2011). Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Research*, *39* (Database issue), D685–D690. doi: 10.1093/nar/gkq1039

Efron, B., & Tibshirani, R. (2007). On testing the significance of sets of genes. *Annals of Applied Statistics, 1*, 107–129. doi: 10.1214/07-AOAS101

Evans, M. J. & Rosenthal, J. S. (2010). *Probability and Statistics: The Science of Uncertainty*, New York, NY: W. H. Freeman.

Goeman, J. J., & Bühlmann, P. (2007). Analyzing gene expression data in terms of gene sets: Methodological issues, *Bioinformatics*, *23*, 980–987. https://doi.org/10.1093/bioinformatics/btm051

Goeman, J. J., Van De Geer, S. A., De Kort, F., & Van Houwelingen, H. C. (2004). A global test for groups of genes: Testing association with a clinical outcome. *Bioinformatics*, *20*, 93–99. doi: 10.1093/bioinformatics/btg382

Hejblum, B. P., Skinner, J., & Thiébaut, R. (2015). Time-course gene set analysis for longitudinal gene expression data, *PLoS Computational Biology*, *11*(6). doi: 10.1371/journal.pcbi.1004310

Hotelling, H. (1931). The generalization of student's ratio. *Annals of Mathematical Statistics, 2*, 360–378.

Izenman, A. J. (2008). *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. New York, NY: Springer.

Lahti, L., Salojärvi, J., Salonen, A., Scheffer, M., & Vos, W. M. (2014). Tipping elements in the human intestinal ecosystem. *Nature Communications, 5*.

Langsrud, Ø. (2005). Rotation tests. *Statistics and Computing*, *15*(1), 53–60. doi: 10.1007/s11222-005-4789-5

Mansmann, U., & Meister, R. (2005). Testing differential gene expression in functional groups: Goemano´s global test versus an ANCOVA approach. *Methods of Information in Medicine, 44*(3), 449–453.

Qiu, W., & Joe, H. (2015). Package 'clusterGeneration' [PDF file]. Retrieved October 23, 2019, from https://cran.r-project.org/web/packages/clusterGeneration/clusterGeneration.pdf

Ripley, B., Venables, B., Bates, D. M., Hornik, K., Gebhardt, A., & Firth, D. (2019). Package 'MASS' [PDF file]. Retrieved October 23, 2019, from https://cran.r-project.org/web/packages/MASS/MASS.pdf

Rizzo, M., & Szekely, G. (2019). Package 'eneergy' [PDF file]. Retrieved October 23, 2019, from https://cran.r-project.org/web/packages/energy/energy.pdf

Signorell, A., Aho, K., Alfons, A., Anderegg, N., Aragon, T., Arppe, A., …, Zeileis, A. (2019). Package 'DescTools' [PDF file]. Retrieved November 1, 2019, from https://cran.r-project.org/web/packages/DescTools/DescTools.pdf

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology, 3*, Article3.

Smyth, G., Hu, Y., Ritchie, M., Silver, J., Wettenhall, J., McCarthy, D., …, Choi, D. (2019). Package 'limma' [PDF file]. Retrieved November 2, 2019, from https://bioconductor.org/packages/release/bioc/manuals/limma/man/limma.pdf

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillete, M. A., …, Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences, 102*, 15545–15550

Székely, G. J., & Rizzo M. L. (2013). The distance correlation t-test of independence in high dimension. *Journal of Multivariate Analysis*, 117, 193-213.

Székely, G. J., & Rizzo M. L. (2014). Partial distance correlation with methods for dissimilarities, *Annals of Statistics*, *42*(6), 2382-2412.

Székely, G. J., Rizzo, M. L., & Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *Annals of Statistics, 35*(6), 2769-2794.

Tomfohr, J., Lu, J., & Kepler, T. B. (2005). Pathway level analysis of gene expression using singular value decomposition. *BMC Bioinformatics*, 6:225. doi: 10.1186/1471-2105-6-225

Wikipedia. (n.d.). Biological pathway. Retrieved March 9, 2020 from https://en.wikipedia.org/wiki/Biological_pathway

Witten, D., Tibshirani, R., Gu, S. G., Fire, A., & Lui, WO. (2010). Ultra-high throughput sequencing-based small RNA discovery and discrete statistical biomarker analysis in a collection of cervical tumours and matched controls. *BMC Biology*, *8*.

Wright, G. W., & Simon, R. M. (2003). A random variance model for detection of differential gene expression in small microarray experiments. *Bioinformatics*, *19*, 2448–2455. doi: 10.1093/bioinformatics/btg345

Wu, D., & Smyth, G. K. (2012). Camera: A competitive gene set test accounting for inter-gene correlation. *Nucleic Acids Research*, *40*, e133. doi: 10.1093/nar/gks461

Wu, D., Lim, E., Vaillant, F., Asselin-Labat, M. –L., Visvader, J. E. & Smyth, G. K. (2010). ROAST: Rotation gene set tests for complex microarray experiments, *Bioinformatics*, 26(17), 2176–2182. doi: 10.1093/bioinformatics/btq401

Zhu, L., Xu, K., Li, R., & Zhong, W. (2017). Projection correlation between two random vectors. *Biometrika*, *104*(4), 829 – 843. doi: 10.1093/biomet/asx043

Appendices

Appendix A: Proof of equation (2.1)

We know that $\mathrm{Cov}(U, V) = E(UV) - E(U)E(V)$.

$$\mathrm{dCov}^2(X, Y) = \mathrm{Cov}\big(|X_1 - X_2|_p, |Y_1 - Y_2|_q\big) - 2\mathrm{Cov}\big(|X_1 - X_2|_p, |Y_1 - Y_3|_q\big)$$

$$= E\big(|X_1 - X_2|_p|Y_1 - Y_2|_q\big) - E\big(|X_1 - X_2|_p\big)E\big(\,|Y_1 - Y_2|_q\big) -$$

$$2\left(E\big(|X_1 - X_2|_p|Y_1 - Y_3|_q\big) - E\big(|X_1 - X_2|_p\big)E\big(\,|Y_1 - Y_3|_q\big)\right)$$

$$= E\big(|X_1 - X_2|_p|Y_1 - Y_2|_q\big) - E\big(|X_1 - X_2|_p\big)E\big(\,|Y_1 - Y_2|_q\big) -$$

$$2E\big(|X_1 - X_2|_p|Y_1 - Y_3|_q\big) + 2\left(E\big(|X_1 - X_2|_p\big)E\big(\,|Y_1 - Y_3|_q\big)\right)$$

$$= E\big(|X_1 - X_2|_p|Y_1 - Y_2|_q\big) + E\big(|X_1 - X_2|_p\big)E\big(|Y_1 - Y_2|_q\big) -$$

$$2E\big(|X_1 - X_2|_p|Y_1 - Y_3|_q\big)$$

$$\begin{pmatrix} \because Y_1, Y_2, Y_3 \text{ are independent and identically distributed} \\ \therefore E\big(\,|Y_1 - Y_2|_q\big) = E\big(\,|Y_1 - Y_3|_q\big) \end{pmatrix}$$

$$= \mathcal{V}^2(X, Y)$$

Appendix B: Proof of equation (2.2)

Suppose $\begin{cases} X|Y=0 \sim f(x)\ \forall X=(-\infty,\infty) \\ X|Y=1 \sim g(x)\ \forall X=(-\infty,\infty) \\ Y \sim Bernoulli(\pi)\ \forall Y=0 \text{ (normal), } 1 \text{ (cancel) where } \pi=P(Y=1) \end{cases}$.

$$\mathcal{V}^2(X,Y)=E\big(|X_1-X_2|_p|Y_1-Y_2|\big)+E\big(|X_1-X_2|_p\big)E(|Y_1-Y_2|)-$$
$$2E\big(|X_1-X_2|_p|Y_1-Y_3|\big) \qquad \text{(by definition)}$$

Let $\begin{cases} d_{00}=E\big(|X_1-X_2|_p|Y_1=0,Y_2=0\big) \\ d_{11}=E\big(|X_1-X_2|_p|Y_1=1,Y_2=1\big) \\ d_{01}=E\big(|X_1-X_2|_p|Y_1=0,Y_2=1\big)=E\big(|X_1-X_2|_p|Y_1=1,Y_2=0\big) \end{cases}$

(1) For the first term:

$$E\big(|X_1-X_2|_p|Y_1-Y_2|\big)$$
$$=E\big(|X_1-X_2|_p|Y_1=0,Y_2=0\big)P(Y_1=0,Y_2=0)+$$
$$E\big(|X_1-X_2|_p|Y_1=1,Y_2=1\big)P(Y_1=1,Y_2=1)+$$
$$E\big(|X_1-X_2|_p|Y_1=0,Y_2=1\big)P(Y_1=0,Y_2=1)+$$
$$E\big(|X_1-X_2|_p|Y_1=1,Y_2=0\big)P(Y_1=1,Y_2=0)$$

(by law of total probability & conditional expectation)

$$=0P(Y_1=0,Y_2=0)+0P(Y_1=1,Y_2=1)+d_{01}P(Y_1=0,Y_2=1)+$$
$$d_{01}P(Y_1=1,Y_2=0)$$

$$\left(\text{as } |Y_1-Y_2|=\begin{cases} 0 \text{ if } Y_1=Y_2=0 \\ 0 \text{ if } Y_1=Y_2=1 \\ 1 \text{ if } Y_1=0 \text{ and } Y_2=1 \\ 1 \text{ if } Y_1=1 \text{ and } Y_2=0 \end{cases}\right)$$

$$=d_{01}P(Y_1=0)P(Y_2=1)+d_{01}P(Y_1=1)P(Y_2=0) \qquad (\text{as } Y_1 \perp Y_2)$$

$$=2d_{01}\pi(1-\pi) \qquad \left(\text{as } \begin{cases} P(Y=1)=\pi \\ P(Y=0)=1-\pi \end{cases}\right).$$

(2) For the second term:

$$E\big(|X_1 - X_2|_p\big)$$

$$= E\big(|X_1 - X_2|_p | Y_1 = 0, Y_2 = 0\big)P(Y_1 = 0, Y_2 = 0) +$$

$$E\big(|X_1 - X_2|_p | Y_1 = 1, Y_2 = 1\big)P(Y_1 = 1, Y_2 = 1) +$$

$$E\big(|X_1 - X_2|_p | Y_1 = 0, Y_2 = 1\big)P(Y_1 = 0, Y_2 = 1) +$$

$$E\big(|X_1 - X_2|_p | Y_1 = 1, Y_2 = 0\big)P(Y_1 = 1, Y_2 = 0)$$

(by law of total probability & conditional expectation)

$$= d_{00}P(Y_1 = 0)P(Y_2 = 0) + d_{11}P(Y_1 = 1)P(Y_2 = 1) +$$

$$d_{01}P(Y_1 = 0)P(Y_2 = 1) + d_{01}P(Y_1 = 1)P(Y_2 = 0) \qquad (\text{as } Y_1 \perp Y_2)$$

$$= d_{00}(1-\pi)^2 + d_{11}\pi^2 + 2d_{01}\pi(1-\pi) \qquad \left(\text{as } \begin{cases} P(Y=1) = \pi \\ P(Y=0) = 1-\pi \end{cases}\right),$$

$$E(|Y_1 - Y_2|)$$

$$= 0P(Y_1 = 0, Y_2 = 0) + 0P(Y_1 = 1, Y_2 = 1) + 1P(Y_1 = 0, Y_2 = 1) +$$

$$1P(Y_1 = 1, Y_2 = 0) \qquad \left(\text{as } |Y_1 - Y_2| = \begin{cases} 0 \text{ if } Y_1 = Y_2 = 0 \\ 0 \text{ if } Y_1 = Y_2 = 1 \\ 1 \text{ if } Y_1 = 0 \text{ and } Y_2 = 1 \\ 1 \text{ if } Y_1 = 1 \text{ and } Y_2 = 0 \end{cases}\right)$$

$$= P(Y_1 = 0, Y_2 = 1) + P(Y_1 = 1, Y_2 = 0)$$

$$= P(Y_1 = 0)P(Y_2 = 1) + P(Y_1 = 1)P(Y_2 = 0) \qquad (\text{as } Y_1 \perp Y_2)$$

$$= \pi(1-\pi) + \pi(1-\pi) \qquad \left(\text{as } \begin{cases} P(Y=1) = \pi \\ P(Y=0) = 1-\pi \end{cases}\right)$$

$$= 2\pi(1-\pi),$$

$$E\big(|X_1 - X_2|_p\big)E(|Y_1 - Y_2|)$$

$$= \big(d_{00}(1-\pi)^2 + d_{11}\pi^2 + 2d_{01}\pi(1-\pi)\big)2\pi(1-\pi)$$

$$= 2d_{00}\pi(1-\pi)^3 + 2d_{11}\pi^3(1-\pi) + 4d_{01}\pi^2(1-\pi)^2.$$

(3) For the third term:

$$E\big(|X_1 - X_2|_p|Y_1 - Y_3|\big)$$

$$= E\big(|X_1 - X_2|_p|Y_1 = 0, Y_2 = 0, Y_3 = 0\big)P(Y_1 = 0, Y_2 = 0, Y_3 = 0) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 0, Y_2 = 1, Y_3 = 0\big)P(Y_1 = 0, Y_2 = 1, Y_3 = 0) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 1, Y_2 = 0, Y_3 = 1\big)P(Y_1 = 1, Y_2 = 0, Y_3 = 1) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 1, Y_2 = 1, Y_3 = 1\big)P(Y_1 = 1, Y_2 = 1, Y_3 = 1) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 0, Y_2 = 0, Y_3 = 1\big)P(Y_1 = 0, Y_2 = 0, Y_3 = 1) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 0, Y_2 = 1, Y_3 = 1\big)P(Y_1 = 0, Y_2 = 1, Y_3 = 1) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 1, Y_2 = 0, Y_3 = 0\big)P(Y_1 = 1, Y_2 = 0, Y_3 = 0) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 1, Y_2 = 1, Y_3 = 0\big)P(Y_1 = 1, Y_2 = 1, Y_3 = 0)$$

(by law of total probability & conditional expectation)

$$= 0P(Y_1 = 0, Y_2 = 0, Y_3 = 0) + 0P(Y_1 = 0, Y_2 = 1, Y_3 = 0) +$$

$$0P(Y_1 = 1, Y_2 = 0, Y_3 = 1) + 0P(Y_1 = 1, Y_2 = 1, Y_3 = 1) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 0, Y_2 = 0, Y_3 = 1\big)P(Y_1 = 0, Y_2 = 0, Y_3 = 1) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 0, Y_2 = 1, Y_3 = 1\big)P(Y_1 = 0, Y_2 = 1, Y_3 = 1) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 1, Y_2 = 0, Y_3 = 0\big)P(Y_1 = 1, Y_2 = 0, Y_3 = 0) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 1, Y_2 = 1, Y_3 = 0\big)P(Y_1 = 1, Y_2 = 1, Y_3 = 0)$$

$$\left(\text{as } |Y_1 - Y_3| = \begin{cases} 0 \text{ if } Y_1 = Y_3 = 0 \\ 0 \text{ if } Y_1 = Y_3 = 1 \\ 1 \text{ if } Y_1 = 0 \text{ and } Y_3 = 1 \\ 1 \text{ if } Y_1 = 1 \text{ and } Y_3 = 0 \end{cases}\right)$$

$$= E\big(|X_1 - X_2|_p|Y_1 = 0, Y_2 = 0\big)P(Y_1 = 0, Y_2 = 0) +$$

$$E\big(|X_1 - X_2|_p|Y_1 = 0, Y_2 = 1\big)P(Y_1 = 0, Y_2 = 1) +$$

$$E\big(|X_1 - X_2|_p | Y_1 = 1, Y_2 = 0\big)P(Y_1 = 1, Y_2 = 0) +$$

$$E\big(|X_1 - X_2|_p | Y_1 = 1, Y_2 = 1\big)P(Y_1 = 1, Y_2 = 1)$$

$$\left(\text{as } \begin{cases} E\big(|X_1 - X_2|_p | Y_1 = 0, Y_2 = 0, Y_3 = 0\big)P(Y_1 = 0, Y_2 = 0, Y_3 = 0) + \\ E\big(|X_1 - X_2|_p | Y_1 = 0, Y_2 = 0, Y_3 = 1\big)P(Y_1 = 0, Y_2 = 0, Y_3 = 1) \\ = E\big(|X_1 - X_2|_p | Y_1 = 0, Y_2 = 0\big)P(Y_1 = 0, Y_2 = 0) \\ E\big(|X_1 - X_2|_p | Y_1 = 0, Y_2 = 1, Y_3 = 0\big)P(Y_1 = 0, Y_2 = 1, Y_3 = 0) + \\ E\big(|X_1 - X_2|_p | Y_1 = 0, Y_2 = 1, Y_3 = 1\big)P(Y_1 = 0, Y_2 = 1, Y_3 = 1) \\ = E\big(|X_1 - X_2|_p | Y_1 = 0, Y_2 = 1\big)P(Y_1 = 0, Y_2 = 1) \\ \vdots \end{cases} \right)$$

$$= d_{00}P(Y_1 = 0, Y_2 = 0) + d_{01}P(Y_1 = 0, Y_2 = 1) + d_{01}P(Y_1 = 1, Y_2 = 0) +$$

$$d_{11}P(Y_1 = 1, Y_2 = 1)$$

$$= d_{00}P(Y_1 = 0)P(Y_2 = 0) + d_{01}P(Y_1 = 0)P(Y_2 = 1) + d_{01}P(Y_1 = 1)P(Y_2 = 0) +$$

$$d_{11}P(Y_1 = 1)P(Y_2 = 1) \qquad (\text{as } Y_1 \perp Y_2)$$

$$= d_{00}(1-\pi)^2 + 2d_{01}\pi(1-\pi) + d_{11}\pi^2 \qquad \left(\text{as } \begin{cases} P(Y = 1) = \pi \\ P(Y = 0) = 1 - \pi \end{cases}\right),$$

$$2E\big(|X_1 - X_2|_p | Y_1 - Y_3|\big) = 2d_{00}(1-\pi)^2 + 4d_{01}\pi(1-\pi) + 2d_{11}\pi^2.$$

$$\mathcal{V}^2(X, Y)$$

$$= E\big(|X_1 - X_2|_p | Y_1 - Y_2|\big) + E\big(|X_1 - X_2|_p\big)E(|Y_1 - Y_2|) - 2E\big(|X_1 - X_2|_p | Y_1 - Y_3|\big)$$

$$= 2d_{01}\pi(1-\pi) + 2d_{00}\pi(1-\pi)^3 + 2d_{11}\pi^3(1-\pi) + 4d_{01}\pi^2(1-\pi)^2 -$$

$$(2d_{00}(1-\pi)^2 + 4d_{01}\pi(1-\pi) + 2d_{11}\pi^2)$$

$$= 2d_{00}(1-\pi)^2(\pi(1-\pi) - 1) + 2d_{11}\pi^2(\pi(1-\pi) - 1) +$$

$$2d_{01}\pi(1-\pi)(1 + 2\pi(1-\pi) - 2)$$

$$= 2d_{00}(1-\pi)^2(\pi(1-\pi) - 1) + 2d_{11}\pi^2(\pi(1-\pi) - 1) +$$

$$2d_{01}\pi(1-\pi)(2\pi(1-\pi) - 1)$$

$$= 2d_{00}(-\pi^4 + 3\pi^3 - 4\pi^2 + 3\pi - 1) + 2d_{11}(-\pi^4 + \pi^3 - \pi^2) +$$

$$2d_{01}(2\pi^4 - 4\pi^3 + 3\pi^2 - \pi)$$

Appendix C: Proof of $\bar{A}_{k\cdot} = 0$ in the double centered distance matrix of X

$$\bar{A}_{k\cdot} = \left(\frac{1}{n}\right)\sum_{i=1}^{n} A_{ki}$$

$$= \left(\frac{1}{n}\right)(A_{k1} + A_{k2} + \cdots + A_{kn})$$

$$= \left(\frac{1}{n}\right)\big((a_{k1} - \bar{a}_{k\cdot} - \bar{a}_{\cdot 1} + \bar{a}_{\cdot\cdot}) + (a_{k2} - \bar{a}_{k\cdot} - \bar{a}_{\cdot 2} + \bar{a}_{\cdot\cdot}) + \cdots +$$

$$(a_{kn} - \bar{a}_{k\cdot} - \bar{a}_{\cdot n} + \bar{a}_{\cdot\cdot})\big)$$

$$= \left(\frac{1}{n}\right)\big((a_{k1} + a_{k2} + \cdots + a_{kn}) - n\bar{a}_{k\cdot} - (\bar{a}_{\cdot 1} + \bar{a}_{\cdot 2} + \cdots + \bar{a}_{\cdot n}) + n\bar{a}_{\cdot\cdot}\big)$$

$$= \left(\frac{1}{n}\right)(a_{k1} + a_{k2} + \cdots + a_{kn}) - \bar{a}_{k\cdot} - \left(\frac{1}{n}\right)(\bar{a}_{\cdot 1} + \bar{a}_{\cdot 2} + \cdots + \bar{a}_{\cdot n}) + \bar{a}_{\cdot\cdot}$$

$$= \bar{a}_{k\cdot} - \bar{a}_{k\cdot} - \bar{a}_{\cdot\cdot} + \bar{a}_{\cdot\cdot}$$

$$= 0.$$

Appendix D: R program for assessing type I error rates (simulation study I)

```
library(MASS)
library(energy)
library(clusterGeneration)
library(DescTools)
library(limma)

alpha <- 0.05
R <- 1000
runs <- 10
n <- c(40, 60, 80, 100, 150, 200, 250, 300)
mu <- rep(0, 8)
ng <- length(mu)
sigma <- c(1,2,3,4,5,0.5,1.5,2.5)

DispersionFunction <- function(sigma, correlation)
  {
    ng <- length(sigma)
    DispersionMatrix <- matrix(NA, nrow=ng, ncol=ng)
    for (i in 1:ng)
      {
        for (j in i:ng)
          {
            DispersionMatrix[i, j] <- correlation[i, j]*sigma[i]*sigma[j]
          }
        for (k in 1:i)
          {
            if (i < ng)
              DispersionMatrix[i + 1, k] <- DispersionMatrix[k, i + 1]
          }
      }
    return(DispersionMatrix)
  }

TypeIErrorSimulation <- function(n, mu, DispersionMatrix, alpha)
  {
    X <- matrix(NA, nrow=n, ncol=ng)
    Y <- matrix(NA, nrow=n, ncol=1)
    NrejectH0_dcor <- 0
    NrejectH0_HotellingsT2 <- 0
    NrejectH0_roast <- 0
    for (r in 1:R)
      {
        X <- mvrnorm(n, mu, DispersionMatrix)
        Y[, 1] <- rbinom(n, size=1, prob=0.5)

        if (dcor.test(X, Y, R=R)$p.value <= alpha)
          {
            NrejectH0_dcor <- NrejectH0_dcor + 1
          }

        if (HotellingsT2Test(X~Y)$p.value <= alpha)
          {
```

```
            NrejectH0_HotellingsT2 <- NrejectH0_HotellingsT2 + 1
          }

        DesignMatrix <- cbind(Intercept=1, Group=Y)
        if (roast(t(X), design=DesignMatrix, contrast=2)$p.value[[2]][4] <= alpha)
          {
            NrejectH0_roast <- NrejectH0_roast + 1
          }
      }
    return(c(NrejectH0_dcor / R, NrejectH0_HotellingsT2 / R, NrejectH0_roast / R))
  }

set.seed(1)
CorrelationMatrix <- rcorrmatrix(length(sigma))
DispersionMatrix <- DispersionFunction(sigma, CorrelationMatrix)

TypeIErrors <- array(NA, dim=c(length(n), runs, 3), dimnames=list(n, 1:runs,
c("dcor.test", "HotellingsT2Test", "roast")))
AverageTypeIErrors <- matrix(NA, nrow=3, ncol=length(n))
SDTypeIErrors <- matrix(NA, nrow=3, ncol=length(n))
for (SampleSize in 1:length(n))
  {
    for (run in 1:runs)
      {
        TypeIErrors[SampleSize, run,] <- TypeIErrorSimulation(n[SampleSize], mu,
DispersionMatrix, alpha)
      }
    for (TestType in 1:3)
      {
        AverageTypeIErrors[TestType, SampleSize] <- mean(TypeIErrors[SampleSize,,
TestType])
        SDTypeIErrors[TestType, SampleSize] <- sd(TypeIErrors[SampleSize,,
TestType])
      }
  }

AverageTypeIErrors
SDTypeIErrors

x11()
plot(n, AverageTypeIErrors[1,], xlim=c(0, 300), ylim=c(0, 0.5), pch=1,
xlab="Sample Size", ylab="Average Type I Error Rate", col="red")
abline(h=0.05)
par(new=T)
plot(n, AverageTypeIErrors[2,], xlim=c(0, 300), ylim=c(0, 0.5), pch=2, xlab="",
ylab="", axes=F, col="blue")
par(new=T)
plot(n, AverageTypeIErrors[3,], xlim=c(0, 300), ylim=c(0, 0.5), pch=8, xlab="",
ylab="", axes=F, col="darkgreen")
legend(150, 0.5, legend=c("Distance Correlation Test", "Hotelling's T Square
Test", "ROAST Test"), col=c("red", "blue", "darkgreen"), pch=c(1, 2, 8), bty="n")
```

Table 6.1

Means and Standard Deviations of Type I Error Rates for Different Sample Sizes and Different Hypothesis Tests

Sample Size	Type I Error Rate					
	Distance Correlation		Hotelling's T^2		ROAST	
	M	*SD*	*M*	*SD*	*M*	*SD*
40	0.051	0.008	0.048	0.005	0.047	0.006
60	0.046	0.003	0.050	0.007	0.047	0.008
80	0.053	0.005	0.052	0.008	0.052	0.004
100	0.051	0.005	0.052	0.007	0.054	0.006
150	0.049	0.008	0.050	0.005	0.049	0.007
200	0.051	0.005	0.047	0.004	0.048	0.006
250	0.055	0.007	0.053	0.009	0.052	0.005
300	0.046	0.006	0.053	0.008	0.051	0.006

Appendix E: R program for assessing powers in the simulation study II

```
library(MASS)
library(energy)
library(clusterGeneration)
library(DescTools)
library(limma)

alpha <- 0.05
R <- 1000
runs <- 10
n <- c(40, 60, 80, 100, 150, 200, 250, 300)

DispersionMatrix <- function(sigma, correlation)
  {
    ng <- length(sigma)
    DispersionMatrix <- matrix(NA, nrow=ng, ncol=ng)
    for (i in 1:ng)
      {
        for (j in i:ng)
          {
            DispersionMatrix[i, j] <- correlation[i, j]*sigma[i]*sigma[j]
          }
        for (k in 1:i)
          {
            if (i < ng)
              DispersionMatrix[i + 1, k] <- DispersionMatrix[k, i + 1]
          }
      }
    return(DispersionMatrix)
  }

PowerSimulation <- function(n, mu_0, mu_1, DispersionMatrix_0, DispersionMatrix_1,
alpha)
  {
    NrejectH0_dcor <- 0
    NrejectH0_HotellingsT2 <- 0
    NrejectH0_roast <- 0
    ng <- length(mu_0)
    X <- matrix(NA, nrow=n, ncol=ng)
    Y <- matrix(NA, nrow=n, ncol=1)
    for (r in 1:R)
      {
        Y[, 1] <- rbinom(n, size=1, prob=0.5)
        for (i in 1:n)
          {
            if (Y[i, 1] == 0)
              X[i,] <- mvrnorm(1, mu_0, DispersionMatrix_0)
            else
              X[i,] <- mvrnorm(1, mu_1, DispersionMatrix_1)
          }

        if (dcor.test(X, Y, R=R)$p.value <= alpha)
          {
```

```
            NrejectH0_dcor <- NrejectH0_dcor + 1
          }

        if (HotellingsT2Test(X~Y)$p.value <= alpha)
          {
            NrejectH0_HotellingsT2 <- NrejectH0_HotellingsT2 + 1
          }

        DesignMatrix <- cbind(Intercept=1, Group=Y)
        if (roast(t(X), design=DesignMatrix, contrast=2)$p.value[[2]][4] <= alpha)
          {
            NrejectH0_roast <- NrejectH0_roast + 1
          }
      }
    return(c(NrejectH0_dcor / R, NrejectH0_HotellingsT2 / R, NrejectH0_roast / R))
  }

#*****************************************************************
#*   Same standard deviation, same correlation, different means   *
#*****************************************************************
sigma_0 <- c(1,2,3,4,5,0.5,1.5,2.5)
sigma_1 <- sigma_0

set.seed(1)
CorrelationMatrix_0 <- rcorrmatrix(length(sigma_0))
CorrelationMatrix_1 <- CorrelationMatrix_0

DispersionMatrix_0 <- DispersionMatrix(sigma_0, CorrelationMatrix_0)
DispersionMatrix_1 <- DispersionMatrix(sigma_1, CorrelationMatrix_1)

#=========================================================
#=   Case 1: mu_0 = {0,0,...,0} and mu_1 = {1,1,...,1}   =
#=========================================================
mu_0 <- rep(0, 8)
mu_1 <- rep(1, 8)

Powers_A1 <- array(NA, dim=c(length(n), runs, 3), dimnames=list(n, 1:runs,
c("dcor.test", "HotellingsT2Test", "roast")))
AveragePowers_A1 <- matrix(NA, nrow=3, ncol=length(n))
SDPowers_A1 <- matrix(NA, nrow=3, ncol=length(n))
for (SampleSize in 1:length(n))
  {
    for (run in 1:runs)
       {
         Powers_A1[SampleSize, run,] <- PowerSimulation(n[SampleSize], mu_0, mu_1,
DispersionMatrix_0, DispersionMatrix_1, alpha)
       }
  }
for (TestType in 1:3)
  {
    for (run in 1:runs)
      {
        for (SampleSize in 1:length(n))
          {
```

```
            AveragePowers_A1[TestType, SampleSize] <- mean(Powers_A1[SampleSize,,
TestType])
            SDPowers_A1[TestType, SampleSize] <- sd(Powers_A1[SampleSize,,
TestType])
          }
      }
  }

AveragePowers_A1
SDPowers_A1
plot(n, AveragePowers_A1[1,], type="b", lty=1, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=1, xlab="Sample Size", ylab="Average Power", col="red")
par(new=T)
plot(n, AveragePowers_A1[2,], type="b", lty=3, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=2, xlab="", ylab="", axes=F, col="blue")
par(new=T)
plot(n, AveragePowers_A1[3,], type="b", lty=5, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=8, xlab="", ylab="", axes=F, col="darkgreen")
legend(150, 0.2, legend=c("Distance Correlation Test", "Hotelling's T Square
Test", "ROAST Test"), col=c("red", "blue", "darkgreen"), lty=c(1, 3, 5), pch=c(1,
2, 8), bty="n")

#=============================================================
#=   Case 2: mu_0 = {0,0,...,0} and mu_1 = {0.5,0.5,...,0.5}   =
#=============================================================
mu_0 <- rep(0, 8)
mu_1 <- rep(0.5, 8)

Powers_A2 <- array(NA, dim=c(length(n), runs, 3), dimnames=list(n, 1:runs,
c("dcor.test", "HotellingsT2Test", "roast")))
AveragePowers_A2 <- matrix(NA, nrow=3, ncol=length(n))
SDPowers_A2 <- matrix(NA, nrow=3, ncol=length(n))
for (SampleSize in 1:length(n))
  {
    for (run in 1:runs)
       {
         Powers_A2[SampleSize, run,] <- PowerSimulation(n[SampleSize], mu_0, mu_1,
DispersionMatrix_0, DispersionMatrix_1, alpha)
       }
  }
for (TestType in 1:3)
  {
    for (run in 1:runs)
      {
        for (SampleSize in 1:length(n))
          {
            AveragePowers_A2[TestType, SampleSize] <-
mean(Powers_A2[SampleSize,,TestType])
            SDPowers_A2[TestType, SampleSize] <-
sd(Powers_A2[SampleSize,,TestType])
          }
      }
  }

AveragePowers_A2
```

```
SDPowers_A2
x11()
plot(n, AveragePowers_A2[1,], type="b", lty=1, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=1, xlab="Sample Size", ylab="Average Power", col="red")
par(new=T)
plot(n, AveragePowers_A2[2,], type="b", lty=3, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=2, xlab="", ylab="", axes=F, col="blue")
par(new=T)
plot(n, AveragePowers_A2[3,], type="b", lty=5, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=8, xlab="", ylab="", axes=F, col="darkgreen")
legend(150, 0.2, legend=c("Distance Correlation Test", "Hotelling's T Square
Test", "ROAST Test"), col=c("red", "blue", "darkgreen"), lty=c(1, 3, 5), pch=c(1,
2, 8), bty="n")

#=========================================================================
#=   Case 3: mu_0 = {0,0,...,0} and mu_1 = {0.5,-0.5,0.5,-0.5,...,-0.5}   =
#=========================================================================
mu_0 <- rep(0, 8)
mu_1 <- c(0.5, -0.5, 0.5, -0.5, 0.5, -0.5, 0.5, -0.5)

Powers_A3 <- array(NA, dim=c(length(n), runs, 3), dimnames=list(n, 1:runs,
c("dcor.test", "HotellingsT2Test", "roast")))
AveragePowers_A3 <- matrix(NA, nrow=3, ncol=length(n))
SDPowers_A3 <- matrix(NA, nrow=3, ncol=length(n))
for (SampleSize in 1:length(n))
  {
    for (run in 1:runs)
       {
         Powers_A3[SampleSize, run,] <- PowerSimulation(n[SampleSize], mu_0, mu_1,
DispersionMatrix_0, DispersionMatrix_1, alpha)
       }
  }
for (TestType in 1:3)
  {
    for (run in 1:runs)
      {
        for (SampleSize in 1:length(n))
          {
            AveragePowers_A3[TestType, SampleSize] <-
mean(Powers_A3[SampleSize,,TestType])
            SDPowers_A3[TestType, SampleSize] <-
sd(Powers_A3[SampleSize,,TestType])
          }
      }
  }

AveragePowers_A3
SDPowers_A3
x11()
plot(n, AveragePowers_A3[1,], type="b", lty=1, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=1, xlab="Sample Size", ylab="Average Power", col="red")
par(new=T)
plot(n, AveragePowers_A3[2,], type="b", lty=3, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=2, xlab="", ylab="", axes=F, col="blue")
par(new=T)
```

```
plot(n, AveragePowers_A3[3,], type="b", lty=5, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=8, xlab="", ylab="", axes=F, col="darkgreen")
legend(150, 0.2, legend=c("Distance Correlation Test", "Hotelling's T Square
Test", "ROAST Test"), col=c("red", "blue", "darkgreen"), lty=c(1, 3, 5), pch=c(1,
2, 8), bty="n")
```

Table 6.2

Means and Standard Deviations of Powers under Case 1 in the Simulation Study II for Different Sample Sizes and Different Hypothesis Tests

Sample Size	Power					
	Distance Correlation		Hotelling's T^2		ROAST	
	M	*SD*	*M*	*SD*	*M*	*SD*
40	0.342	0.008	1.000	0.000	0.996	0.001
60	0.573	0.015	1.000	0.000	1.000	0.000
80	0.776	0.013	1.000	0.000	1.000	0.000
100	0.910	0.015	1.000	0.000	1.000	0.000
150	0.998	0.001	1.000	0.000	1.000	0.000
200	1.000	0.000	1.000	0.000	1.000	0.000
250	1.000	0.000	1.000	0.000	1.000	0.000
300	1.000	0.000	1.000	0.000	1.000	0.000

Table 6.3

Means and Standard Deviations of Powers under Case 2 in the Simulation Study II for Different Sample Sizes and Different Hypothesis Tests

Sample Size	Power					
	Distance Correlation		Hotelling's T^2		ROAST	
	M	*SD*	*M*	*SD*	*M*	*SD*
40	0.096	0.012	0.996	0.003	0.402	0.011
60	0.126	0.005	1.000	0.000	0.665	0.011
80	0.164	0.004	1.000	0.000	0.855	0.008
100	0.197	0.010	1.000	0.000	0.951	0.007
150	0.325	0.017	1.000	0.000	0.999	0.001
200	0.460	0.022	1.000	0.000	1.000	0.000
250	0.613	0.012	1.000	0.000	1.000	0.000
300	0.736	0.015	1.000	0.000	1.000	0.000

Table 6.4

Means and Standard Deviations of Powers under Case 3 in the Simulation Study II for Different Sample Sizes and Different Hypothesis Tests

Sample Size	Power					
	Distance Correlation		Hotelling's T^2		ROAST	
	M	*SD*	*M*	*SD*	*M*	*SD*
40	0.097	0.013	1.000	0.000	0.380	0.014
60	0.130	0.013	1.000	0.000	0.598	0.012
80	0.166	0.015	1.000	0.000	0.761	0.008
100	0.208	0.011	1.000	0.000	0.873	0.009
150	0.357	0.013	1.000	0.000	0.988	0.003
200	0.531	0.013	1.000	0.000	1.000	0.001
250	0.704	0.010	1.000	0.000	1.000	0.000
300	0.841	0.011	1.000	0.000	1.000	0.000

Appendix F: R program for assessing powers in the simulation study III

```
library(MASS)
library(energy)
library(clusterGeneration)
library(DescTools)
library(limma)
alpha <- 0.05
R <- 1000
runs <- 10
n <- c(40, 60, 80, 100, 150, 200, 250, 300)

DispersionFunction <- function(sigma, correlation)
  {
    ng <- length(sigma)
    DispersionMatrix <- matrix(NA, nrow=ng, ncol=ng)
    for (i in 1:ng)
      {
        for (j in i:ng)
          {
            DispersionMatrix[i, j] <- correlation[i, j]*sigma[i]*sigma[j]
          }
        for (k in 1:i)
          {
            if (i < ng)
              DispersionMatrix[i + 1, k] <- DispersionMatrix[k, i + 1]
          }
      }
    return(DispersionMatrix)
  }

PowerSimulation <- function(n, mu_0, mu_1, DispersionMatrix_0, DispersionMatrix_1,
alpha)
  {
    NrejectH0_dcor <- 0
    NrejectH0_HotellingsT2 <- 0
    NrejectH0_roast <- 0
    ng <- length(mu_0)
    X <- matrix(NA, nrow=n, ncol=ng)
    Y <- matrix(NA, nrow=n, ncol=1)
    for (r in 1:R)
      {
        Y[, 1] <- rbinom(n, size=1, prob=0.5)
        for (i in 1:n)
          {
            if (Y[i, 1] == 0)
              X[i,] <- mvrnorm(1, mu_0, DispersionMatrix_0)
            else
              X[i,] <- mvrnorm(1, mu_1, DispersionMatrix_1)
          }

        if (dcor.test(X, Y, R=R)$p.value <= alpha)
          {
            NrejectH0_dcor <- NrejectH0_dcor + 1
```

```
          }

        if (HotellingsT2Test(X~Y)$p.value <= alpha)
          {
            NrejectH0_HotellingsT2 <- NrejectH0_HotellingsT2 + 1
          }

        DesignMatrix <- cbind(Intercept=1, Group=Y)
        if (roast(t(X), design=DesignMatrix, contrast=2)$p.value[[2]][4] <= alpha)
          {
            NrejectH0_roast <- NrejectH0_roast + 1
          }
      }
    return(c(NrejectH0_dcor / R, NrejectH0_HotellingsT2 / R, NrejectH0_roast / R))
  }

#*************************************************************************
#*   Same mean, different standard deviations, and different correlations   *
#*************************************************************************
#=========================================================================
=======
#=   Case 1: mu_0 = mu_1 = {0,0,...,0}
=
#=           sigma_0 = {1,2,3,4,5,0.5,1.5,2.5} and sigma_1 =
{5,4,0.5,3,2.5,1.5,2,1}   =
#=========================================================================
=======
mu_0 <- rep(0, 8)
mu_1 <- mu_0
sigma_0 <- c(1,2,3,4,5,0.5,1.5,2.5)
sigma_1 <- c(5,4,0.5,3,2.5,1.5,2,1)

set.seed(1)
CorrelationMatrix_0 <- rcorrmatrix(length(sigma_0))
CorrelationMatrix_1 <- rcorrmatrix(length(sigma_1))

DispersionMatrix_0 <- DispersionFunction(sigma_0, CorrelationMatrix_0)
DispersionMatrix_1 <- DispersionFunction(sigma_1, CorrelationMatrix_1)

Powers_B1 <- array(NA, dim=c(length(n), runs, 3), dimnames=list(n, 1:runs,
c("dcor.test", "HotellingsT2Test", "roast")))
AveragePowers_B1 <- matrix(NA, nrow=3, ncol=length(n))
SDPowers_B1 <- matrix(NA, nrow=3, ncol=length(n))
for (SampleSize in 1:length(n))
  {
    for (run in 1:runs)
       {
         Powers_B1[SampleSize, run,] <- PowerSimulation(n[SampleSize], mu_0, mu_1,
DispersionMatrix_0, DispersionMatrix_1, alpha)
       }
  }
for (TestType in 1:3)
  {
    for (run in 1:runs)
      {
```

```
        for (SampleSize in 1:length(n))
          {
            AveragePowers_B1[TestType, SampleSize] <- mean(Powers_B1[SampleSize,,
TestType])
            SDPowers_B1[TestType, SampleSize] <- sd(Powers_B1[SampleSize,,
TestType])
          }
      }
  }

AveragePowers_B1
SDPowers_B1
plot(n, AveragePowers_B1[1,], type="b", lty=1, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=1, xlab="Sample Size", ylab="Average Power", col="red")
par(new=T)
plot(n, AveragePowers_B1[2,], type="b", lty=3, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=2, xlab="", ylab="", axes=F, col="blue")
par(new=T)
plot(n, AveragePowers_B1[3,], type="b", lty=5, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=8, xlab="", ylab="", axes=F, col="darkgreen")
legend(150, 0.8, legend=c("Distance Correlation Test", "Hotelling's T Square
Test", "ROAST Test"), col=c("red", "blue", "darkgreen"), lty=c(1, 3, 5), pch=c(1,
2, 8), bty="n")

#=============================================================================
=======
#=   Case 2: mu_0 = mu_1 = {0.5,-0.5,0.5,-0.5,...,-0.5}
=
#=           sigma_0 = {1,2,3,4,5,0.5,1.5,2.5} and sigma_1 =
{5,4,0.5,3,2.5,1.5,2,1}   =
#=============================================================================
=======
mu_0 <- c(0.5, -0.5, 0.5, -0.5, 0.5, -0.5, 0.5, -0.5)   #? ENTER the population
means for X|Y=0
mu_1 <- mu_0        #? ENTER the population means for X|Y=1

sigma_0 <- c(1,2,3,4,5,0.5,1.5,2.5)
sigma_1 <- c(5,4,0.5,3,2.5,1.5,2,1)

CorrelationMatrix_0 <- rcorrmatrix(length(sigma_0))
CorrelationMatrix_1 <- rcorrmatrix(length(sigma_1))

DispersionMatrix_0 <- DispersionFunction(sigma_0, CorrelationMatrix_0)
DispersionMatrix_1 <- DispersionFunction(sigma_1, CorrelationMatrix_1)

Powers_B2 <- array(NA, dim=c(length(n), runs, 3), dimnames=list(n, 1:runs,
c("dcor.test", "HotellingsT2Test", "roast")))
AveragePowers_B2 <- matrix(NA, nrow=3, ncol=length(n))
SDPowers_B2 <- matrix(NA, nrow=3, ncol=length(n))
for (SampleSize in 1:length(n))
  {
    for (run in 1:runs)
       {
         Powers_B2[SampleSize, run,] <- PowerSimulation(n[SampleSize], mu_0, mu_1,
DispersionMatrix_0, DispersionMatrix_1, alpha)
```

```
      }
  }
for (TestType in 1:3)
  {
    for (run in 1:runs)
      {
        for (SampleSize in 1:length(n))
          {
            AveragePowers_B2[TestType, SampleSize] <- mean(Powers_B2[SampleSize,,
TestType])
            SDPowers_B2[TestType, SampleSize] <- sd(Powers_B2[SampleSize,,
TestType])
          }
      }
  }

AveragePowers_B2
SDPowers_B2
x11()
plot(n, AveragePowers_B2[1,], type="b", lty=1, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=1, xlab="Sample Size", ylab="Average Power", col="red")
par(new=T)
plot(n, AveragePowers_B2[2,], type="b", lty=3, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=2, xlab="", ylab="", axes=F, col="blue")
par(new=T)
plot(n, AveragePowers_B2[3,], type="b", lty=5, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=8, xlab="", ylab="", axes=F, col="darkgreen")
legend(150, 0.8, legend=c("Distance Correlation Test", "Hotelling's T Square
Test", "ROAST Test"), col=c("red", "blue", "darkgreen"), lty=c(1, 3, 5), pch=c(1,
2, 8), bty="n")
```

Table 6.5

Means and Standard Deviations of Powers under Case 1 in the Simulation Study III for Different Sample Sizes and Different Hypothesis Tests

Sample Size	Power					
	Distance Correlation		Hotelling's T^2		ROAST	
	M	*SD*	*M*	*SD*	*M*	*SD*
40	0.220	0.016	0.074	0.007	0.061	0.010
60	0.427	0.016	0.062	0.010	0.057	0.006
80	0.724	0.013	0.059	0.007	0.053	0.006
100	0.931	0.008	0.058	0.009	0.059	0.008
150	1.000	0.000	0.056	0.006	0.051	0.006
200	1.000	0.000	0.052	0.007	0.048	0.006
250	1.000	0.000	0.055	0.009	0.053	0.008
300	1.000	0.000	0.056	0.006	0.052	0.008

Table 6.6

Means and Standard Deviations of Powers under Case 2 in the Simulation Study III for Different Sample Sizes and Different Hypothesis Tests

Sample Size	Power					
	Distance Correlation		Hotelling's T^2		ROAST	
	M	*SD*	*M*	*SD*	*M*	*SD*
40	0.225	0.012	0.068	0.010	0.060	0.010
60	0.469	0.011	0.063	0.005	0.057	0.006
80	0.777	0.009	0.062	0.010	0.054	0.006
100	0.957	0.005	0.056	0.007	0.054	0.009
150	1.000	0.000	0.058	0.003	0.056	0.009
200	1.000	0.000	0.053	0.010	0.053	0.009
250	1.000	0.000	0.054	0.006	0.056	0.006
300	1.000	0.000	0.053	0.004	0.046	0.005

Appendix G: R program for assessing powers in the simulation study IV

```
library(MASS)
library(energy)
library(clusterGeneration)
library(DescTools)
library(limma)
alpha <- 0.05
R <- 1000
runs <- 10
n <- c(40, 60, 80, 100, 150, 200, 250, 300)

DispersionMatrix <- function(sigma, correlation)
  {
    ng <- length(sigma)
    DispersionMatrix <- matrix(NA, nrow=ng, ncol=ng)
    for (i in 1:ng)
      {
        for (j in i:ng)
          {
            DispersionMatrix[i, j] <- correlation[i, j]*sigma[i]*sigma[j]
          }
        for (k in 1:i)
          {
            if (i < ng)
              DispersionMatrix[i + 1, k] <- DispersionMatrix[k, i + 1]
          }
      }
    return(DispersionMatrix)
  }

PowerSimulation <- function(n, mu_0, mu_1, DispersionMatrix_0, DispersionMatrix_1,
alpha)
  {
    NrejectH0_dcor <- 0
    NrejectH0_HotellingsT2 <- 0
    NrejectH0_roast <- 0
    ng <- length(mu_0)
    X <- matrix(NA, nrow=n, ncol=ng)
    Y <- matrix(NA, nrow=n, ncol=1)
    for (r in 1:R)
      {
        Y[, 1] <- rbinom(n, size=1, prob=0.5)
        for (i in 1:n)
          {
            if (Y[i, 1] == 0)
              X[i,] <- mvrnorm(1, mu_0, DispersionMatrix_0)
            else
              X[i,] <- mvrnorm(1, mu_1, DispersionMatrix_1)
          }

        if (dcor.test(X, Y, R=R)$p.value <= alpha)
          {
            NrejectH0_dcor <- NrejectH0_dcor + 1
```

```
          }

        if (HotellingsT2Test(X~Y)$p.value <= alpha)
          {
            NrejectH0_HotellingsT2 <- NrejectH0_HotellingsT2 + 1
          }

        DesignMatrix <- cbind(Intercept=1, Group=Y)
        if (roast(t(X), design=DesignMatrix, contrast=2)$p.value[[2]][4] <= alpha)
          {
            NrejectH0_roast <- NrejectH0_roast + 1
          }
      }
    return(c(NrejectH0_dcor / R, NrejectH0_HotellingsT2 / R, NrejectH0_roast / R))
  }

#*****************************************************************
#*   Same dispersion matrix, same mean, different correlations   *
#*****************************************************************
mu_0 <- rep(0, 8)
mu_1 <- mu_0
sigma_0 <- c(1,2,3,4,5,0.5,1.5,2.5)
sigma_1 <- sigma_0

set.seed(1)
CorrelationMatrix_0 <- rcorrmatrix(length(sigma_0))
CorrelationMatrix_1 <- rcorrmatrix(length(sigma_1))

DispersionMatrix_0 <- DispersionMatrix(sigma_0, CorrelationMatrix_0)
DispersionMatrix_1 <- DispersionMatrix(sigma_1, CorrelationMatrix_1)

Powers_C <- array(NA, dim=c(length(n), runs, 3), dimnames=list(n, 1:runs,
c("dcor.test", "HotellingsT2Test", "roast")))
AveragePowers_C <- matrix(NA, nrow=3, ncol=length(n))
SDPowers_C <- matrix(NA, nrow=3, ncol=length(n))
for (SampleSize in 1:length(n))
  {
    for (run in 1:runs)
       {
         Powers_C[SampleSize, run,] <- PowerSimulation(n[SampleSize], mu_0, mu_1,
DispersionMatrix_0, DispersionMatrix_1, alpha)
       }
  }
for (TestType in 1:3)
  {
    for (run in 1:runs)
      {
        for (SampleSize in 1:length(n))
          {
            AveragePowers_C[TestType, SampleSize] <- mean(Powers_C[SampleSize,,
TestType])
            SDPowers_C[TestType, SampleSize] <- sd(Powers_C[SampleSize,,
TestType])
          }
      }
```

```
  }

AveragePowers_C
SDPowers_C
plot(n, AveragePowers_C[1,], type="b", lty=1, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=1, xlab="Sample Size", ylab="Average Power", col="red")
par(new=T)
plot(n, AveragePowers_C[2,], type="b", lty=3, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=2, xlab="", ylab="", axes=F, col="blue")
par(new=T)
plot(n, AveragePowers_C[3,], type="b", lty=5, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=8, xlab="", ylab="", axes=F, col="darkgreen")
legend(0, 1, legend=c("Distance Correlation Test", "Hotelling's T Square Test",
"ROAST Test"), col=c("red", "blue", "darkgreen"), lty=c(1, 3, 5), pch=c(1, 2, 8),
bty="n")
```

Table 6.7

Means and Standard Deviations of Powers in the Simulation Study IV for Different Sample Sizes and Different Hypothesis Tests

Sample Size	Power					
	Distance Correlation		Hotelling's T^2		ROAST	
	M	*SD*	*M*	*SD*	*M*	*SD*
40	0.094	0.010	0.069	0.009	0.054	0.009
60	0.122	0.010	0.058	0.008	0.048	0.007
80	0.170	0.010	0.054	0.006	0.049	0.007
100	0.240	0.012	0.055	0.007	0.051	0.007
150	0.473	0.012	0.050	0.007	0.047	0.008
200	0.791	0.014	0.050	0.006	0.049	0.007
250	0.965	0.004	0.054	0.010	0.052	0.006
300	0.998	0.001	0.056	0.005	0.053	0.007

Appendix H: R program for assessing powers in the simulation study V

```
library(MASS)
library(energy)
library(clusterGeneration)
library(DescTools)
library(limma)
alpha <- 0.05
R <- 1000
runs <- 10
n <- c(40, 60, 80, 100, 150, 200, 250, 300)

DispersionMatrix <- function(sigma, correlation)
  {
    ng <- length(sigma)
    DispersionMatrix <- matrix(NA, nrow=ng, ncol=ng)
    for (i in 1:ng)
      {
        for (j in i:ng)
          {
            DispersionMatrix[i, j] <- correlation[i, j]*sigma[i]*sigma[j]
          }
        for (k in 1:i)
          {
            if (i < ng)
              DispersionMatrix[i + 1, k] <- DispersionMatrix[k, i + 1]
          }
      }
    return(DispersionMatrix)
  }

PowerSimulation <- function(n, mu_0, mu_1, DispersionMatrix_0, DispersionMatrix_1,
alpha)
  {
    NrejectH0_dcor <- 0
    NrejectH0_HotellingsT2 <- 0
    NrejectH0_roast <- 0
    ng <- length(mu_0)
    X <- matrix(NA, nrow=n, ncol=ng)
    Y <- matrix(NA, nrow=n, ncol=1)
    for (r in 1:R)
      {
        # Generate data for X and Y
        Y[, 1] <- rbinom(n, size=1, prob=0.5)
        for (i in 1:n)
          {
            if (Y[i, 1] == 0)
              X[i,] <- mvrnorm(1, mu_0, DispersionMatrix_0)
            else
              X[i,] <- mvrnorm(1, mu_1, DispersionMatrix_1)
          }

        if (dcor.test(X, Y, R=R)$p.value <= alpha)
          {
```

```
            NrejectH0_dcor <- NrejectH0_dcor + 1
          }

        if (HotellingsT2Test(X~Y)$p.value <= alpha)
          {
            NrejectH0_HotellingsT2 <- NrejectH0_HotellingsT2 + 1
          }

        DesignMatrix <- cbind(Intercept=1, Group=Y)
        if (roast(t(X), design=DesignMatrix, contrast=2)$p.value[[2]][4] <= alpha)
          {
            NrejectH0_roast <- NrejectH0_roast + 1
          }
      }
    return(c(NrejectH0_dcor / R, NrejectH0_HotellingsT2 / R, NrejectH0_roast / R))
  }

#**************************************************************************
#*   Different means, different standard deviations, different correlations   *
#**************************************************************************
sigma_0 <- c(1,2,3,4,5,0.5,1.5,2.5)
sigma_1 <- c(5,4,0.5,3,2.5,1.5,2,1)

set.seed(1)
CorrelationMatrix_0 <- rcorrmatrix(length(sigma_0))
CorrelationMatrix_1 <- rcorrmatrix(length(sigma_1))

DispersionMatrix_0 <- DispersionMatrix(sigma_0, CorrelationMatrix_0)
DispersionMatrix_1 <- DispersionMatrix(sigma_1, CorrelationMatrix_1)

#=========================================================================
=======
#=   Case 1: mu_0 = {0,0,...,0} and mu_1 = {1,1,...,1}
=
#=           sigma_0 = {1,2,3,4,5,0.5,1.5,2.5} and sigma_1 =
{5,4,0.5,3,2.5,1.5,2,1}   =
#=========================================================================
=======
mu_0 <- rep(0, 8)
mu_1 <- rep(1, 8)

Powers_D1 <- array(NA, dim=c(length(n), runs, 3), dimnames=list(n, 1:runs,
c("dcor.test", "HotellingsT2Test", "roast")))
AveragePowers_D1 <- matrix(NA, nrow=3, ncol=length(n))
SDPowers_D1 <- matrix(NA, nrow=3, ncol=length(n))
for (SampleSize in 1:length(n))
  {
    for (run in 1:runs)
       {
         Powers_D1[SampleSize, run,] <- PowerSimulation(n[SampleSize], mu_0, mu_1,
DispersionMatrix_0, DispersionMatrix_1, alpha)
       }
  }
for (TestType in 1:3)
  {
```

```
    for (run in 1:runs)
      {
        for (SampleSize in 1:length(n))
          {
            AveragePowers_D1[TestType, SampleSize] <- mean(Powers_D1[SampleSize,,
TestType])
            SDPowers_D1[TestType, SampleSize] <- sd(Powers_D1[SampleSize,,
TestType])
          }
      }
  }

AveragePowers_D1
SDPowers_D1
plot(n, AveragePowers_D1[1,], type="b", lty=1, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=1, xlab="Sample Size", ylab="Average Power", col="red")
par(new=T)
plot(n, AveragePowers_D1[2,], type="b", lty=3, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=2, xlab="", ylab="", axes=F, col="blue")
par(new=T)
plot(n, AveragePowers_D1[3,], type="b", lty=5, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=8, xlab="", ylab="", axes=F, col="darkgreen")
legend(150, 0.2, legend=c("Distance Correlation Test", "Hotelling's T Square
Test", "ROAST Test"), col=c("red", "blue", "darkgreen"), lty=c(1, 3, 5), pch=c(1,
2, 8), bty="n")

#=============================================================================
=======
#=   Case 2: mu_0 = {0,0,...,0} and mu_1 = {0.5,0.5,...,0.5}
=
#=           sigma_0 = {1,2,3,4,5,0.5,1.5,2.5} and sigma_1 =
{5,4,0.5,3,2.5,1.5,2,1}   =
#=============================================================================
=======
mu_0 <- rep(0, 8)   #? ENTER the population means for X|Y=0
mu_1 <- rep(0.5, 8)   #? ENTER the population means for X|Y=1

Powers_D2 <- array(NA, dim=c(length(n), runs, 3), dimnames=list(n, 1:runs,
c("dcor.test", "HotellingsT2Test", "roast")))
AveragePowers_D2 <- matrix(NA, nrow=3, ncol=length(n))
SDPowers_D2 <- matrix(NA, nrow=3, ncol=length(n))
for (SampleSize in 1:length(n))
  {
    for (run in 1:runs)
       {
         Powers_D2[SampleSize, run,] <- PowerSimulation(n[SampleSize], mu_0, mu_1,
DispersionMatrix_0, DispersionMatrix_1, alpha)
       }
  }
for (TestType in 1:3)
  {
    for (run in 1:runs)
      {
        for (SampleSize in 1:length(n))
          {
```

```
            AveragePowers_D2[TestType, SampleSize] <- mean(Powers_D2[SampleSize,,
TestType])
            SDPowers_D2[TestType, SampleSize] <- sd(Powers_D2[SampleSize,,
TestType])
          }
      }
  }

AveragePowers_D2
SDPowers_D2
x11()
plot(n, AveragePowers_D2[1,], type="b", lty=1, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=1, xlab="Sample Size", ylab="Average Power", col="red")
par(new=T)
plot(n, AveragePowers_D2[2,], type="b", lty=3, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=2, xlab="", ylab="", axes=F, col="blue")
par(new=T)
plot(n, AveragePowers_D2[3,], type="b", lty=5, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=8, xlab="", ylab="", axes=F, col="darkgreen")
legend(150, 0.2, legend=c("Distance Correlation Test", "Hotelling's T Square
Test", "ROAST Test"), col=c("red", "blue", "darkgreen"), lty=c(1, 3, 5), pch=c(1,
2, 8), bty="n")

#=============================================================================
=======
#=   Case 3: mu_0 = {0,0,...,0} and mu_1 = {0.5,-0.5,0.5,-0.5,...,-0.5}
=
#=           sigma_0 = {1,2,3,4,5,0.5,1.5,2.5} and sigma_1 =
{5,4,0.5,3,2.5,1.5,2,1}   =
#=============================================================================
=======
mu_0 <- rep(0, 8)   #? ENTER the population means for X|Y=0
mu_1 <- c(0.5, -0.5, 0.5, -0.5, 0.5, -0.5, 0.5, -0.5)   #? ENTER the population
means for X|Y=1

Powers_D3 <- array(NA, dim=c(length(n), runs, 3), dimnames=list(n, 1:runs,
c("dcor.test", "HotellingsT2Test", "roast")))
AveragePowers_D3 <- matrix(NA, nrow=3, ncol=length(n))
SDPowers_D3 <- matrix(NA, nrow=3, ncol=length(n))
for (SampleSize in 1:length(n))
  {
    for (run in 1:runs)
       {
         Powers_D3[SampleSize, run,] <- PowerSimulation(n[SampleSize], mu_0, mu_1,
DispersionMatrix_0, DispersionMatrix_1, alpha)
       }
  }
for (TestType in 1:3)
  {
    for (run in 1:runs)
      {
        for (SampleSize in 1:length(n))
          {
            AveragePowers_D3[TestType, SampleSize] <- mean(Powers_D3[SampleSize,,
TestType])
```

```
            SDPowers_D3[TestType, SampleSize] <- sd(Powers_D3[SampleSize,,
TestType])
          }
      }
  }

AveragePowers_D3
SDPowers_D3
x11()
plot(n, AveragePowers_D3[1,], type="b", lty=1, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=1, xlab="Sample Size", ylab="Average Power", col="red")
par(new=T)
plot(n, AveragePowers_D3[2,], type="b", lty=3, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=2, xlab="", ylab="", axes=F, col="blue")
par(new=T)
plot(n, AveragePowers_D3[3,], type="b", lty=5, xlim=c(0, 300), ylim=c(0.0, 1.0),
pch=8, xlab="", ylab="", axes=F, col="darkgreen")
legend(150, 0.2, legend=c("Distance Correlation Test", "Hotelling's T Square
Test", "ROAST Test"), col=c("red", "blue", "darkgreen"), lty=c(1, 3, 5), pch=c(1,
2, 8), bty="n")
```

Table 6.8

Means and Standard Deviations of Powers under Case 1 in the Simulation Study V for Different Sample Sizes and Different hypothesis Tests

Sample Size	Power					
	Distance Correlation		Hotelling's T^2		ROAST	
	M	*SD*	*M*	*SD*	*M*	*SD*
40	0.749	0.014	0.751	0.014	0.769	0.016
60	0.971	0.004	0.943	0.009	0.937	0.007
80	0.999	0.001	0.990	0.004	0.986	0.003
100	1.000	0.000	0.999	0.001	0.997	0.002
150	1.000	0.000	1.000	0.000	1.000	0.000
200	1.000	0.000	1.000	0.000	1.000	0.000
250	1.000	0.000	1.000	0.000	1.000	0.000
300	1.000	0.000	1.000	0.000	1.000	0.000

Table 6.9

Means and Standard Deviations of Powers under Case 2 in the Simulation Study V for Different Sample Sizes and Different hypothesis Tests

Sample Size	Power					
	Distance Correlation		Hotelling's T^2		ROAST	
	M	*SD*	*M*	*SD*	*M*	*SD*
40	0.370	0.021	0.233	0.019	0.228	0.015
60	0.678	0.016	0.354	0.020	0.341	0.017
80	0.911	0.008	0.471	0.013	0.445	0.016
100	0.993	0.003	0.594	0.012	0.558	0.014
150	1.000	0.000	0.808	0.011	0.767	0.018
200	1.000	0.000	0.956	0.005	0.889	0.012
250	1.000	0.000	0.977	0.005	0.953	0.006
300	1.000	0.000	0.994	0.002	0.982	0.002

Table 6.10

Means and Standard Deviations of Powers under Case 3 in the Simulation Study V for Different Sample Sizes and Different Hypothesis Tests

Sample Size	Power					
	Distance Correlation		Hotelling's T^2		ROAST	
	M	*SD*	*M*	*SD*	*M*	*SD*
40	0.389	0.011	0.299	0.014	0.225	0.011
60	0.711	0.008	0.467	0.015	0.332	0.017
80	0.933	0.005	0.610	0.016	0.445	0.018
100	0.994	0.003	0.743	0.013	0.556	0.007
150	1.000	0.000	0.921	0.007	0.788	0.008
200	1.000	0.000	0.982	0.004	0.911	0.008
250	1.000	0.000	0.996	0.001	0.969	0.005
300	1.000	0.000	0.999	0.001	0.992	0.003

Appendix I: R program for the first real data application

```
library(energy)
library(DescTools)
library(limma)
library(moments)

# The original dataset is preprocessed to cancer_dataA.csv
CancerData <- read.csv("cancer_dataA.csv", header=F)

n <- nrow(CancerData)
p <- ncol(CancerData) - 1

Y <- rep(NA, n)
for (i in 1:n)
  {
    ifelse (CancerData[i, 1] == "N", Y[i] <- 0, Y[i] <- 1)
  }

X <- CancerData[, 2:715]
total <- colSums(X)
logX <- log(X + 1)

threshold <- seq(from=40, to=129000, by=20)
I <- length(threshold)
R <- 1000
DesignMatrix <- cbind(Intercept=1, Group=Y)
p_values_temp <- matrix(NA, nrow=I, ncol=3, dimnames = list(threshold,
c("NumberOfGenes","dcor", "ROAST")))

set.seed(1)
for (i in 1:I)
  {
    index_t <- total > threshold[i]
    XData <- logX[, index_t]
    p_values_temp[i, 1] <- ncol(XData)
    p_values_temp[i, 2] <- dcor.test(XData, Y, R=R)$p.value
    p_values_temp[i, 3] <- roast(t(XData), design=DesignMatrix,
contrast=2)$p.value[[2]][4]
  }
index_p <- rep(NA, I)
index_p[1] = T
for (j in 1:(I - 1))
  {
    index_p[j + 1] <- p_values_temp[j, 1] > p_values_temp[j + 1, 1]
  }
p_values <- p_values_temp[index_p,]

hist(p_values[,2], breaks=4, xlim=c(0, 0.035), ylim=c(0,150), xlab="p-value",
ylab="Frequency", main=NULL, col="green")
par(new=T)
hist(p_values[,3], nclass=20, xlim=c(0, 0.035), ylim=c(0,150), xlab="p-value",
ylab="Frequency", main=NULL, col="lightblue")
```

```
legend(0.02, 150, legend=c("Distance Correlation Test", "ROAST Test"),
col=c("green", "lightblue"), pch=c(15, 15),bty="n")
box()

x11()
plot(threshold, p_values_temp[, 1], pch=20, ylim=c(0,500), xlab="Threshold",
ylab="Number of Genes", type="l", main=NULL)
x11()
NumberofGenes <- hist(total[total>=0 & total<120000], breaks=seq(from=0,
to=120000, by=12000), xlab="Threshold", ylab="Number of Genes", main=NULL)
box()
NumberofGenes$counts
NumberofGenes$breaks
NumberofGenes$mids

max(p_values[,2])
min(p_values[,2])
mean(p_values[,2])
sd(p_values[,2])

max(p_values[,3])
min(p_values[,3])
mean(p_values[,3])
sd(p_values[,3])

x11()
MeanDifference <- hist((colMeans(logX[!Y,])-colMeans(logX[as.logical(Y),])),
breaks=seq(-4, 4, by=0.5), xlab="Mean Difference", main=NULL, col="lightblue")
box()
xfit <- seq(-4, 4, length=60)
yfit <- dnorm(xfit, mean(colMeans(logX[!Y,])-colMeans(logX[as.logical(Y),])),
sd(colMeans(logX[!Y,])-colMeans(logX[as.logical(Y),])))
yfit <- yfit*diff(MeanDifference$mids[1:2])*length(colMeans(logX[!Y,])-
colMeans(logX[as.logical(Y),]))
lines(xfit, yfit, col="red")
legend(0.5, 250, legend=c("Histogram of Mean Differences", "Normal Curve"),
col=c("lightblue", "red"), pch=c(15, 15),bty="n", cex=0.95)

skewness((colMeans(logX[!Y,])-colMeans(logX[as.logical(Y),])))
MeanDifference$counts
sum(MeanDifference$counts)
sum(colMeans(logX[!Y,])-colMeans(logX[as.logical(Y),])==0)
which(colMeans(logX[!Y,])-colMeans(logX[as.logical(Y),])==0)
MeanDifference$breaks
MeanDifference$mids

x11()
Ratio_SD <- hist(apply(logX[!Y,], 2, sd)/apply(logX[as.logical(Y),], 2, sd),
breaks=seq(0,5, by=0.4), xlab="Ratio of Standard Deviations", main=NULL)
box()
Ratio_SD$counts
sum(Ratio_SD$counts)
sum(apply(logX[!Y,], 2, sd)/apply(logX[as.logical(Y),], 2, sd)==0)
logX[!Y,which(apply(logX[!Y,], 2, sd)/apply(logX[as.logical(Y),], 2, sd)==0)]
sum(apply(logX[!Y,], 2, sd)/apply(logX[as.logical(Y),], 2, sd)==Inf)
```

```
logX[as.logical(Y),which(apply(logX[!Y,], 2, sd)/apply(logX[as.logical(Y),], 2,
sd)==Inf)]
Ratio_SD$breaks
Ratio_SD$mids

counts <- matrix(NA, nrow=6, ncol=1, dimnames=list(c("(X0bar-X1bar)<0", "(X0bar-
X1bar)=0", "(X0bar-X1bar)>0", "(s0/s1)<1", "(s0/s1)=0", "(s0/s1)>1"), "counts"))
counts[1, 1] <- sum((colMeans(logX[!Y,])-colMeans(logX[as.logical(Y),])) < 0)
counts[2, 1] <- sum((colMeans(logX[!Y,])-colMeans(logX[as.logical(Y),])) == 0)
counts[3, 1] <- sum((colMeans(logX[!Y,])-colMeans(logX[as.logical(Y),])) > 0)
counts[4, 1] <- sum((apply(logX[!Y,], 2, sd)/apply(logX[as.logical(Y),], 2, sd))
< 1)
counts[5, 1] <- sum((apply(logX[!Y,], 2, sd)/apply(logX[as.logical(Y),], 2, sd))
== 1)
counts[6,1] <- sum((apply(logX[!Y,], 2, sd)/apply(logX[as.logical(Y),], 2, sd)) >
1)
counts
```

Appendix J: R program for the second real data application

```
library(energy)
library(DescTools)
library(limma)
library(moments)

Metadata <- read.table("Metadata.tab", header=T)
HITChip <- read.table("HITChip.tab", sep="\t", row.names=1, header=T)
dim(HITChip)
max(HITChip)
min(HITChip)

n <- nrow(HITChip)
p <- ncol(HITChip)

ColumnNumberOfMetadata <- c(2, 3, 4, 8)
index <- matrix(NA, nrow=n, ncol=4,dimnames=list(NULL, c("Age", "Sex",
"Nationality", "BMI")))
for (j in 1:length(ColumnNumberOfMetadata))
  {
    for (i in 1:n)
      {
        index[i, j] <- !is.na(Metadata[i, ColumnNumberOfMetadata[j]])
      }
  }

n <- matrix(NA, nrow=4, ncol=1, dimnames=list(c("Age", "Sex", "Nationality",
"BMI"), "Sample Size"))
for (i in 1:length(ColumnNumberOfMetadata))
  {
    n[i] <- sum(index[, i])
  }
n

Y <- list(rep(NA, n[1, 1]), rep(NA, n[2, 1]), rep(NA, n[3, 1]), rep(NA, n[4, 1]))
for (i in 1:n[1, 1])
  {
    ifelse (Metadata[index[, 1], 2][i] <= 40, Y[[1]][i] <- 0, Y[[1]][i] <- 1)
  }

Age0 <- sum(!Y[[1]])
Age0
Age1 <- sum(Y[[1]])
Age1

Conditions <- c("male", "US", "lean")
for (j in 2:length(ColumnNumberOfMetadata))
  {
    for (i in 1:n[j, 1])
      {
        ifelse (Metadata[index[, j], ColumnNumberOfMetadata[j]][i] ==
Conditions[j - 1], Y[[j]][i] <- 0, Y[[j]][i] <- 1)
      }
```

```
  }

Sex0 <- sum(!Y[[2]])
Sex0
Sex1 <- sum(Y[[2]])
Sex1

Nationality0 <- sum(!Y[[3]])
Nationality0
Nationality1 <- sum(Y[[3]])
Nationality1

BMI0 <- sum(!Y[[4]])
BMI0
BMI1 <- sum(Y[[4]])
BMI1

X <- list(log(HITChip[index[, 1],]), log(HITChip[index[, 2],]),
log(HITChip[index[, 3],]), log(HITChip[index[, 4],]))

set.seed(1)
p_values <- matrix(NA, nrow=4, ncol=3, dimnames=list(c("Age", "Sex",
"Nationality", "BMI"), c("dcor", "Hotteling's T2", "ROAST")))
R <- 1000
for (i in 1:length(ColumnNumberOfMetadata))
  {
    p_values[i, 1] <- dcor.test(X[[i]], Y[[i]], R=R)$p.value
    p_values[i, 2] <-
HotellingsT2Test(as.matrix(X[[i]])~as.matrix(Y[[i]]))$p.value
    DesignMatrix <- cbind(Intercept=1, Group=Y[[i]])
    p_values[i, 3] <- roast(t(X[[i]]), design=DesignMatrix,
contrast=2)$p.value[[2]][4]
  }
p_values

labels <- c("Age", "Sex", "Nationality", "BMI")
counts <- matrix(NA, nrow=6, ncol=4, dimnames=list(c("(X0bar-X1bar)<0", "(X0bar-
X1bar)=0", "(X0bar-X1bar)>0", "(s0/s1)<1", "(s0/s1)=1", "(s0/s1)>1"), c("Age",
"Sex", "Nationality", "BMI")))
Skewness <- matrix(NA, nrow=2, ncol=4, dimnames=list(c("MeanDifference", "SD
Ratio"), c("Age", "Sex", "Nationality", "BMI")))
for (i in 1:length(ColumnNumberOfMetadata))
  {
    x11(width=14.125,height=7.0625)
    par(mfrow=c(1, 2))
    histogram <- hist((colMeans(X[[i]][!Y[[i]],])-
colMeans(X[[i]][as.logical(Y[[i]]),])), breaks=seq(-1.5, 1, by=0.1), ylim=c(0,
90), main=labels[i], xlab="Mean Difference")
    Skewness[1, i] <- skewness((colMeans(X[[i]][!Y[[i]],])-
colMeans(X[[i]][as.logical(Y[[i]]),])))
    box()
    #xfit <- seq(-1.5, 1, length=60)
    #yN01 <- dnorm(xfit, 0, 1)
    #yfit <- yN01*diff(histogram$mids[1:2])*sum(histogram$counts)
    #lines(xfit, yfit, col="red")
```

```
    hist(apply(X[[i]][!Y[[i]],], 2, sd)/apply(X[[i]][as.logical(Y[[i]]),], 2, sd),
breaks=seq(0, 4, by=0.15), ylim=c(0, 90), main=labels[i], xlab="Ratio of Standard
Deviations")
    Skewness[2, i] <- skewness(apply(X[[i]][!Y[[i]],], 2,
sd)/apply(X[[i]][as.logical(Y[[i]]),], 2, sd))
    box()
    counts[1, i] <- sum((colMeans(X[[i]][!Y[[i]],])-
colMeans(X[[i]][as.logical(Y[[i]]),])) < 0)
    counts[2, i] <- sum((colMeans(X[[i]][!Y[[i]],])-
colMeans(X[[i]][as.logical(Y[[i]]),])) == 0)
    counts[3, i] <- sum((colMeans(X[[i]][!Y[[i]],])-
colMeans(X[[i]][as.logical(Y[[i]]),])) > 0)
    counts[4, i] <- sum(apply(X[[i]][!Y[[i]],], 2,
sd)/apply(X[[i]][as.logical(Y[[i]]),], 2, sd) < 1)
    counts[5, i] <- sum(apply(X[[i]][!Y[[i]],], 2,
sd)/apply(X[[i]][as.logical(Y[[i]]),], 2, sd) == 1)
    counts[6, i] <- sum(apply(X[[i]][!Y[[i]],], 2,
sd)/apply(X[[i]][as.logical(Y[[i]]),], 2, sd) > 1)
  }
counts
Skewness
```