Statistical Methods for Analysis of Genetic Data

Christopher R. Cabanski

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2012

Approved by:

Dr. J.S. Marron, Advisor

Dr. D. Neil Hayes, Advisor

Dr. Nilay Argon, Reader

Dr. Yufeng Liu, Reader

Dr. Michael Wu, Reader

© 2012
Christopher R. Cabanski
ALL RIGHTS RESERVED

Abstract

CHRISTOPHER R. CABANSKI: Statistical Methods for Analysis of Genetic Data
(Under the direction of Dr. J.S. Marron and Dr. D. Neil Hayes.)

Genetic studies of gene expression typically aim to identify a set of genes that are associated with a disease, such as a specific cancer type. A single microarray or next generation sequencing experiment can simultaneously measure gene expression for tens of thousands of genes. When analyzing high-dimensional gene expression data, clusters in the data often represent biological quantities of interest, such as tumor subtypes. In this dissertation, we describe Standardized WithIn Class Sum of Squares (SWISS), a statistical tool that quantifies how well a high-dimensional data set clusters into predefined classes. We show SWISS to be very useful in genetic studies for comparing two different processing methods on the same data set by indicating which processing method yields better relative separation between classes. Additionally, we investigate the asymptotic behavior of SWISS in the High Dimension Low Sample Size setting, where the sample size is fixed and the dimension grows.

Next generation sequencing is rapidly becoming the technology of choice for genomic studies. This technology allows millions of fragments of DNA to be simultaneously sequenced. Unfortunately, this technology is not error-free and occasionally will call an incorrect base. When a base is sequenced, a quality score is also provided which corresponds to the probability that the base called is incorrect. In the second half of this dissertation, we show that these quality scores do not accurately represent the probability of a sequencing error. We describe a method that recalibrates these quality scores and show that these recalibrated scores are more accurate and better at discriminating sequencing errors from non-errors.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the past several decades, there has been an increased interest in understanding diseases at the genetic level. The end goal is to translate the results learned from genetic studies into targeted therapies. This increased interest has coincided with the invention of several technologies to measure genetic quantities of interest, such as gene expression. Within the past decade, sequencing the human genome of multiple patients has become feasible, resulting in an influx of new data yet to be mined for interesting biological properties. As the genetic technologies have rapidly developed, there has been an increased need for new statistical analyses that account for and manage the many sources of bias and error present in these technologies. This dissertation describes two different statistical tools, SWISS and ReQON, which were developed to analyze genetic data.

Contemporary high dimensional gene expression assays, such as mRNA expression microarrays, regularly involve multiple data processing steps, such as experimental processing, computational processing, sample selection, or feature selection (i.e., gene selection), prior to deriving any biological conclusions. These steps can dramatically change the interpretation of an experiment. Evaluation of processing steps has received limited attention in the literature. It is not straightforward to evaluate different processing methods and investigators are often unsure of the best method. Chapter 2 describes Standardized WithIn class Sum of Squares (SWISS), a simple statistical tool based on classical ANOVA techniques that gives a quantitative comparison of alternate data processing methods in terms of quality of clustering into predefined biological classes. This chapter also presents two permutation tests for assessing significance of a SWISS score. We also apply SWISS to evaluate different processing methods for two different applications.

In Chapter 3, we investigate mathematical statistical properties of SWISS. Because SWISS has proven itself useful in comparing different data processing methods on high-dimensional datasets, we will explore the high-dimensional asymptotic behavior of SWISS. Specifically, we describe the asymptotic behavior of SWISS in the High Dimension Low Sample Size setting, where the sample size is fixed and the dimension grows. We show that the SWISS score has a useful asymptotic representation in this setting. Additionally, we extend the results to the High Dimension Moderate Sample Size setting, where the sample size grows slowly along with the dimension.

Genetic studies where a sample's entire genome is sequenced are becoming increasingly common. For example, sequencing tumor genomes has provided new insight to cancer biology. The amount of sequencing data is rapidly growing by the day. However, statistical and computational tools to analyze this data are not keeping pace. Part of the challenge of analyzing sequencing data is the presence of many different types of error. For example, a sequencer machine will occasionally call an incorrect base, which is referred to as a sequencing error. Chapter 4 describes ReQON, a tool which aims to accurately predict the probability that a base is a sequencing error. Incorporating these probabilities in downstream analyses by more accurately accounting for errors present in the data will guide researchers and increase confidence in their results.

Chapter 2

SWISS: Standardized WithIn Class Sum of Squares

This chapter describes Standardized WithIn class Sum of Squares (SWISS), a tool used to evaluate how well data cluster into predefined classes. SWISS is often used to compare alternate data processing methods on the same dataset. SWISS was first described by Cabanski *et al.* (2010).

This chapter is laid out as follows. Section 2.1 describes the motivation behind SWISS. Section 2.2 describes how to calculate SWISS scores and also how to perform two different permutation tests to assess significance of a SWISS score. Section 2.3 extends SWISS to the multiclass setting. Section 2.4 compares SWISS to other competing methods in the literature. Section 2.5 describes two different applications of SWISS using gene expression data that show the simplicity and usefulness of SWISS. We address possible limitations and biases of SWISS in Section 2.6.

2.1 Motivation

Suppose there is a gene expression dataset (Fan and Ren, 2006) with two or more biological phenotypes (classes), such as tumor/normal or tumor subtypes. The SWISS method assumes that the biological classes are predefined with at least two data points in each class. Frequently, there is interest in comparing different processing methods, such as normalization techniques or gene filterings. There is also interest in comparing differing experimental designs, such as different protocols, microarray platforms, or technologies.

Many problems can arise when trying to evaluate two processing methods or compare different platforms. For instance:

- The best way to compare methods/platforms is not always clear when the data are on different scales or have been normalized in different ways.

- It is important to select the optimal method in an unbiased way. For example, bias could arise from choosing a normalization or other processing method based on which method calls the largest number of significant genes.

- A simple hypothesis may be difficult to formulate. For example, the answer to "Which microarray platform is preferred?" may depend on the normalization method chosen and the amount and type of filtering on the gene set. A more appropriate question is, "Given that the data have been normalized and filtered in a specific manner, which platform is preferred?".

Motivated by these problems, one goal is to develop a more generic approach to comparing processing methods. We propose a new method, Standardized WithIn class Sum of Squares (SWISS), defined in Subsection 2.2.1 on the following page, that uses Euclidean distance to measure which processing method under investigation gives a more effective clustering of data into biological phenotypes. SWISS takes a multivariate approach to determining the best processing method. SWISS tends to be driven by differentially expressed genes (genes with large variation between the classes) and tends to ignore noise genes (genes with little variation across all data points).

We have developed a permutation test based on SWISS (Subsection 2.2.2) to determine if the data are better clustered than expected by random chance. Additionally, we developed a second permutation test (Subsection 2.2.3) for comparing two SWISS scores, possibly from different data processing methods on the same dataset, to determine whether their difference is statistically significant.

2.2 Methods

This section describes how to calculate the SWISS score (Subsection 2.2.1) and two corresponding permutation tests (Subsections 2.2.2 and 2.2.3). A variety of toy examples are presented in this section to show the full range of possible SWISS scores and also to demonstrate the need for the permutation tests.

2.2.1 Standardized WithIn class Sum of Squares (SWISS)

The goal of this chapter is to develop a statistic that quantifies how clustered the data are. The proposed statistic should jointly consider two aspects of the data. First, it should measure how tight the clusters are. Second, this statistic should also measure the separation between clusters. In general, if a dataset is strongly clustered, the clusters should be tight and spread far apart. The proposed statistic should also have the additional property that it allows for easy comparison across datasets, even when the data are on different scales. This subsection describes the SWISS statistic and explains how it satisfies all of the criteria just described.

For simplicity of presentation, assume that the data consist of two classes, $\mathscr{X}(d) = \{X_1(d), \ldots, X_m(d)\}$ and $\mathscr{Y}(d) = \{Y_1(d), \ldots, Y_n(d)\}$, where $X_i(d) = \left(X_i^{(1)}, \ldots, X_i^{(d)}\right)^{\mathrm{T}}$ and $Y_j(d) = \left(Y_j^{(1)}, \ldots, Y_j^{(d)}\right)^{\mathrm{T}}$ are d-dimensional data vectors. The setting where there are more than two classes is discussed in Section 2.3. Let $N = m+n$ be the total sample size, $\overline{W}$ the d-dimensional overall mean vector of all N data points, and $\overline{X}$ and $\overline{Y}$ the mean vector of class $\mathscr{X}(d)$ and $\mathscr{Y}(d)$, respectively. Following classical analysis of variance (ANOVA) ideas, the Total WithIn class Sum of Squares (TWISS) is defined to be

$$\mathrm{TWISS} = \sum_{i=1}^{m}\sum_{p=1}^{d}\left\{X_i^{(p)} - \overline{X}^{(p)}\right\}^2 + \sum_{j=1}^{n}\sum_{p=1}^{d}\left\{Y_j^{(p)} - \overline{Y}^{(p)}\right\}^2$$

and the the Total Sum of Squares (TSS) is

$$\mathrm{TSS} = \sum_{i=1}^{m}\sum_{p=1}^{d}\left\{X_i^{(p)} - \overline{W}^{(p)}\right\}^2 + \sum_{j=1}^{n}\sum_{p=1}^{d}\left\{Y_j^{(p)} - \overline{W}^{(p)}\right\}^2.$$

The Standardized WithIn Class Sum of Squares (SWISS), which is the proportion of variation unexplained by clustering, is defined as

$$\mathrm{SWISS} = \frac{\mathrm{TWISS}}{\mathrm{TSS}}.$$

The SWISS score will always be between zero and one. This is because (1) TWISS and TSS are both non-negative and (2) TSS is at least as large as TWISS.

Figure 2.1 shows a two gene toy example where the two classes are denoted by red x's and blue o's. Plot A shows the distances used in calculating TWISS. The class means are represented by red

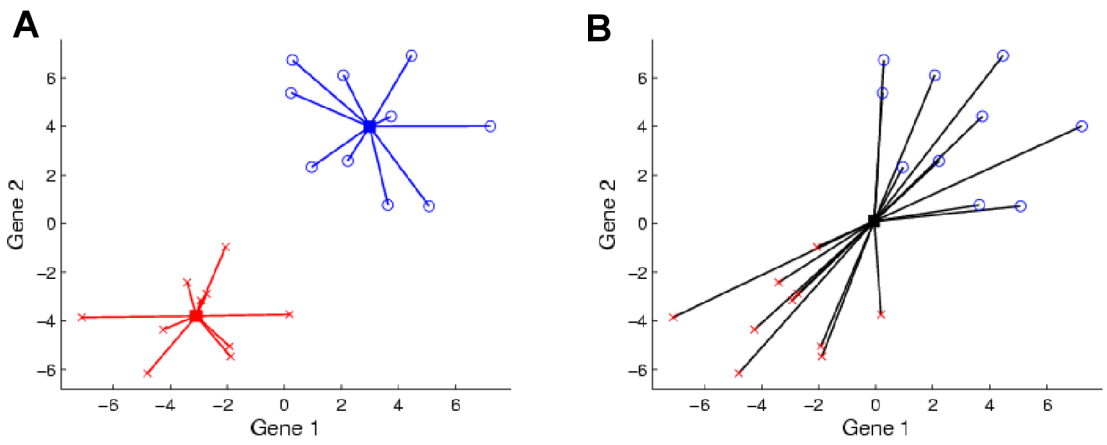

Figure 2.1: Two gene toy example demonstrating distances used in calculating SWISS scores. The red x's and blue o's represent the two classes. The colored lines in plot A show the distance between each point and its respective class mean (red and blue squares). The black lines in plot B show the distance between each point and the overall mean (black square).

and blue squares. TWISS, the numerator of SWISS, is calculated by taking the distance between each point and the class mean (denoted by colored lines), squaring this distance, then summing the squared distances over all points. TWISS measures the amount of spread in the clusters. Plot B shows the distances used in calculating TSS, the denominator of SWISS. The overall mean is represented by a black square. TSS is calculated by taking the distance between each point and the overall mean (denoted by black lines), squaring this distance, then summing the squared distances over all points. TSS is a measure of the overall amount of variation, or energy, present in the data. TSS is affected by the spread between the two classes, and as the separation between the classes increases, TSS will also increase. Normalizing TWISS by TSS allows for comparison between multiple SWISS scores, even when the data are on different scales. For the data shown in Figure 2.1, TWISS is 151.9 and TSS is 640.4. Thus, the SWISS score of this data is $\frac{151.9}{640.4} = 0.24$.

SWISS has the following properties:

- *Unit free*. SWISS does not have any units and is comparable between datasets that are on different scales. This is because SWISS is normalized by dividing by TSS.

- *Shift invariant*. Shifting the data by a fixed amount does not change SWISS because the distances between each point and its class mean and overall mean remain constant.

- *Scale invariant*. Multiplying the data by a common factor, say c, does not change SWISS. TWISS and TSS will both change by a factor of c^2, which will cancel out when taking the ratio to calculate SWISS.

- *Rotation invariant*. SWISS is unchanged when the data are arbitrarily rotated around a fixed point for the same reason that SWISS is shift invariant.

One-dimensional Toy Examples Demonstrating the Range of SWISS

Figures 2.2 through 2.4 show how changes in the class means, class standard deviations and proportion of points in each class, respectively, are reflected in the SWISS score. Each figure shows a sequence of one-dimensional (1D) distribution plots. Each dataset was generated from a mixture of two Gaussian distributions. The data have been colored according to mixture component (class) and the heights have been jittered to provide visual separation of the points. The red and blue curves are smooth histograms of the red and blue points, respectively, and the black curves are smooth histograms of the entire population. SWISS scores are reported at the top of each plot.

Figure 2.2 shows how changing the standard deviation of the classes while keeping the class means constant drives SWISS scores. The first plot shows the data when the standard deviations are both zero, and this results in a SWISS score of zero. The SWISS score will always be zero when both of the class standard deviations are zero because the distance between each point and its class mean is zero and, hence, TWISS is zero. Then, as the standard deviations of the classes increase, the SWISS score also increases. This makes sense because as the standard deviations increase, TWISS increases at a faster rate than TSS, which results in a larger SWISS score. Notice that for the first plot in the third row, the black curve no longer dips between the red and blue curves. This means that there is no longer a clear separation between the two classes and this corresponds to a SWISS score of 0.31. In the final plot, there is no distinction between the the two classes and, therefore, the SWISS score is very close to one.

Figure 2.3 shows how changing the distance between class means, while keeping the standard deviation constant, is captured by the SWISS score. The first plot shows the data when class means are very far apart and there is no overlap between the two classes, and this results in a SWISS score

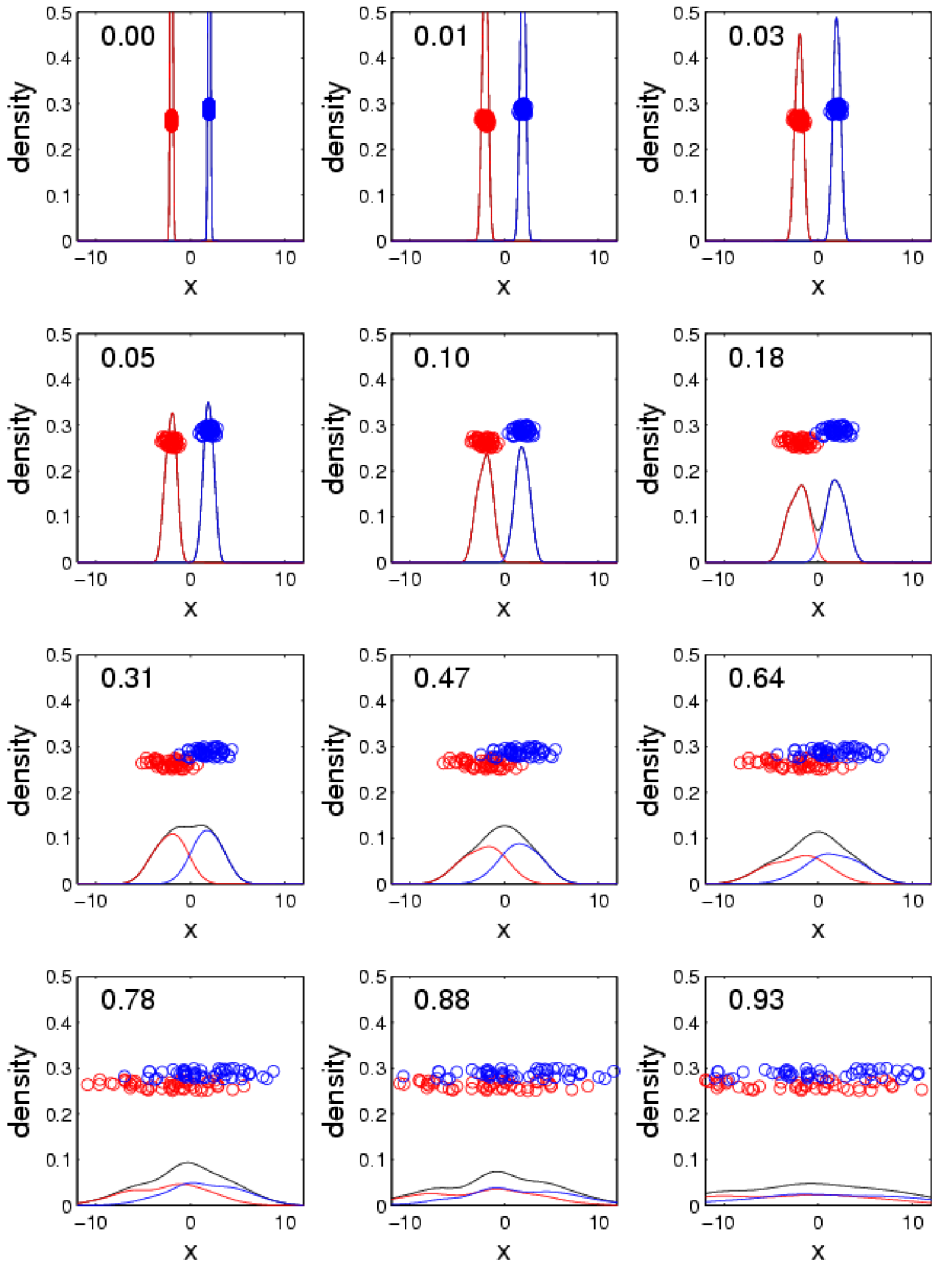

Figure 2.2: Sequence of 1D scatter plots with jittered heights. The data have been split into two classes, denoted by different colors. The red and blue curves are smooth histograms of the red and blue points, respectively. The means of the two classes are fixed and the standard deviations are varied. The SWISS scores are reported at the top of each plot. As the standard deviation increases, the SWISS score also increases.

very close to zero. Then, as the distance between class means decreases, the SWISS score increases because TSS decreases while TWISS remains constant. The last plot shows the data when the class means are equal, and this results in a SWISS score of one. This makes sense because the overall mean is the same as the class means. Therefore, TWISS is equal to TSS, and hence, the SWISS score equals one.

Figure 2.4 shows how changing the proportion of points in each class, while keeping the class means and standard deviations constant, is quantified by the SWISS score. Each plot has a total of 100 points. The first plot shows the data when there are an equal number of points in the two classes and, for this example, the SWISS score is very close to zero. Then, as the number of points in each class becomes more uneven, the SWISS score increases. However, the SWISS score may not increase all the way to one, as seen in the final plot. This plot shows the data when there are only 2 red points and 98 blue points, and the SWISS score is 0.44. This increase in the SWISS score makes sense because, as the proportion of points in each class becomes more unbalanced, the overall mean will move towards the larger class. This will decrease TSS which results in a larger SWISS score.

Two-dimensional Toy Example

Suppose there is an interest in comparing two different datasets, or one dataset that has been processed using two different methods, such as different normalizations. A smaller SWISS score reflects that the data have either: (1) tighter clusters (smaller TWISS) and/or (2) clusters that are spread further apart (larger TSS). Thus, one processing method, say Method A, is *better* in the sense of SWISS than another processing method, say Method B, if Method A has a smaller SWISS score than Method B (Cabanski *et al.*, 2010). In this sense, Method A gives a more effective clustering of data into the predefined classes than Method B.

Figure 2.5 illustrates a variety of possible SWISS scores on a two-dimensional toy example. These plots confirm that data that are better clustered, in the sense that the two classes have better separation and tighter clusters (plot A, SWISS = 0.25), have a lower SWISS score than data where there is less separation between the classes (plot B, SWISS = 0.91). Notice that the axes are the same for all plots. When comparing the clustering of the data shown in plots A and C, it is unclear by visual inspection which method clusters the data better. Also, TWISS cannot be directly compared because

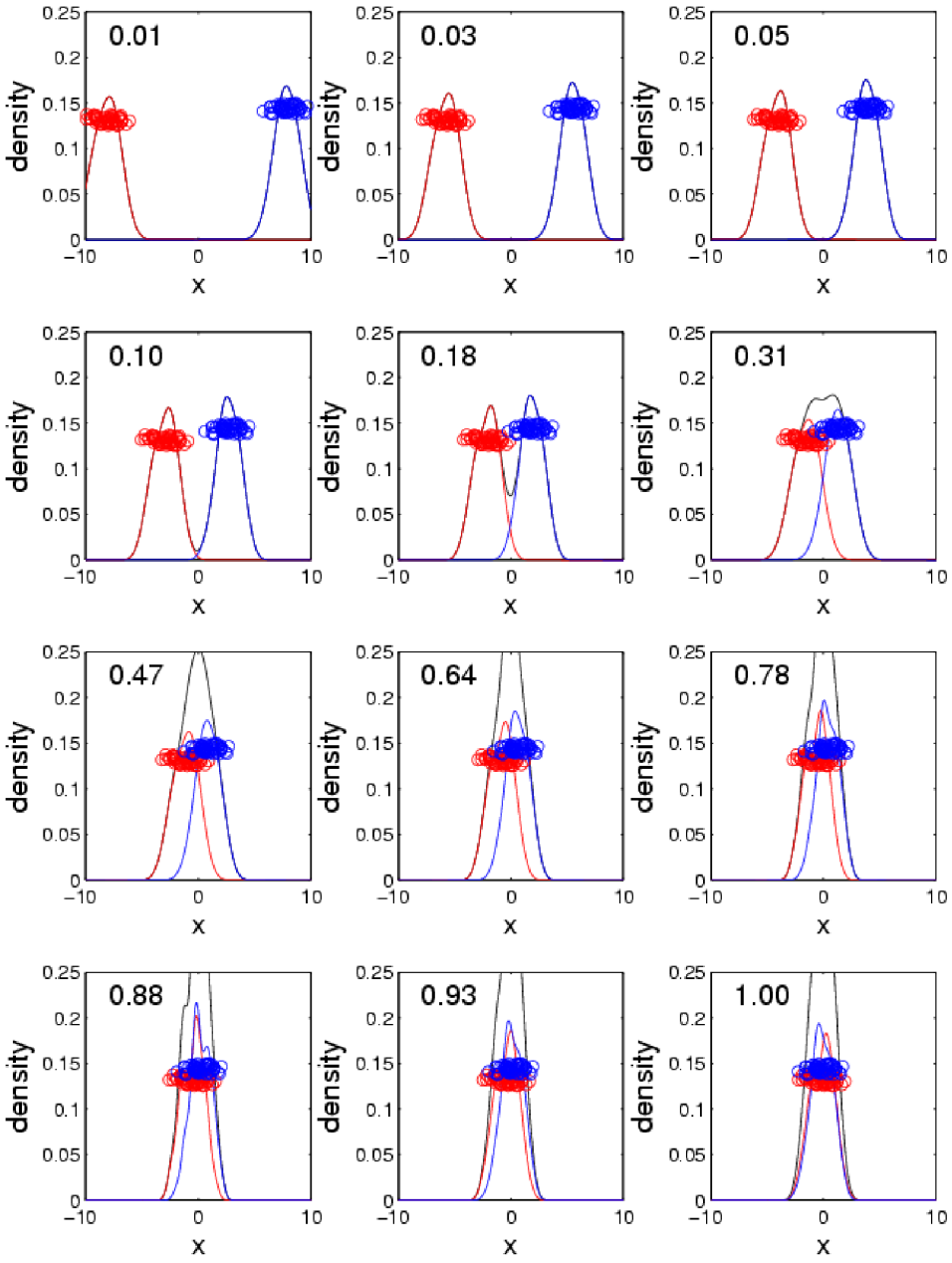

Figure 2.3: Sequence of 1D scatter plots with jittered heights. The data have been split into two classes, denoted by different colors. The red and blue curves are smooth histograms of the red and blue points, respectively. The standard deviations of the two classes are fixed and the distance between the means is varied. The SWISS scores are reported at the top of each plot. As the class means converge, the SWISS score increases.

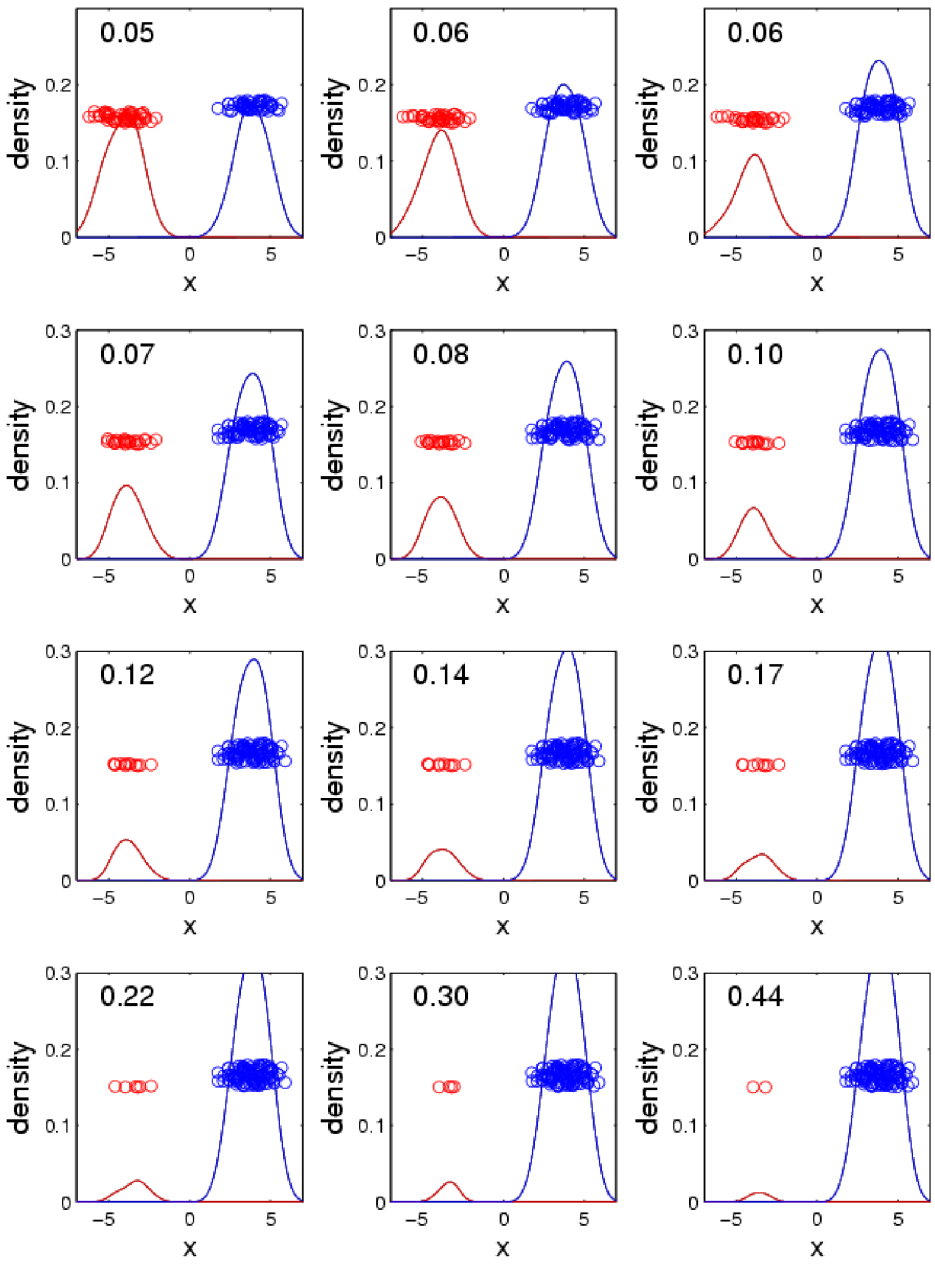

Figure 2.4: Sequence of 1D scatter plots with jittered heights. The data have been split into two classes, denoted by different colors. The red and blue curves are smooth histograms of the red and blue points, respectively. The standard deviations and means of the two classes are fixed and the proportion of points in each class is varied. The SWISS scores are reported at the top of each plot. As the proportion of points in each class becomes more unbalanced, the SWISS score increases.

the datasets appear to be on different scales. However, once TWISS is standardized, the SWISS scores have the same scale and are directly comparable. The data visualized in plots A and C have equivalent clustering performance because both datasets have a SWISS score of 0.25. The data shown in plot D are a simple shift of the data in plot C, with the two classes shifted toward each other. Although TWISS is constant between plots C and D, this small shift still has a large effect on the SWISS score (0.25 vs. 0.46).

One may wonder whether the SWISS score in plot B is significantly lower than one. That is, is the clustering performance better than random chance? Subsection 2.2.2 describes a permutation test based on SWISS, where the null hypothesis is that SWISS equals one. In that subsection, we will show that this permutation test is equivalent to testing for a difference in class means.

Suppose that the data visualized in plots A and D come from the same dataset that has been processed using two different processing methods, which we will refer to as Method A and Method D, respectively. Because the SWISS score of Method A is lower than the SWISS score of Method D (0.25 vs. 0.46), Method A is preferred over Method D. However, suppose there is a preference for using Method D. For example, Method D may be easier to implement or may be more cost effective. To answer the question of whether the difference between the SWISS scores of Methods A and D is statistically significant, a permutation test based on SWISS is developed and described in detail in Subsection 2.2.3.

2.2.2 One-sample SWISS Permutation Test

This section describes a permutation test to determine if the clustering performance based on predefined class labels is better than random chance. For this test, the null and alternative hypotheses are $H_0 : \text{SWISS} = 1$ vs. $H_1 : \text{SWISS} < 1$. To perform this permutation test, first decide on the number of permutations, n_{perm}. Setting n_{perm} to be at least 1000 is suggested. For each permutation, randomly permute the class labels and recalculate the SWISS score. The p-value is the proportion of permuted SWISS scores that are less than the original SWISS score. If the p-value is small, reject the null hypothesis and conclude that the SWISS score is significantly less than 1.

Let us return to the toy example shown in plot B of Figure 2.5. The SWISS score of this data is 0.91. We would like to test whether this SWISS score is significantly less than one. Figure 2.6

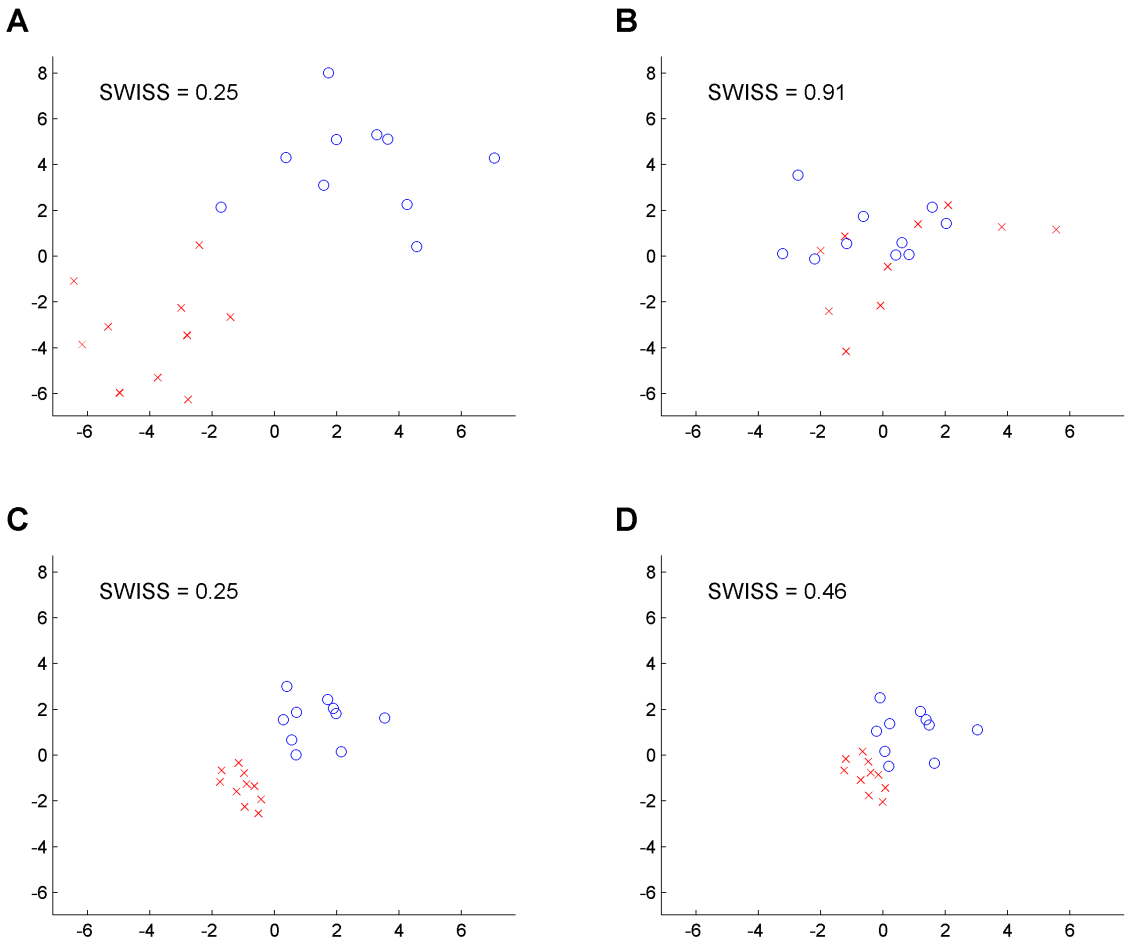

Figure 2.5: Two-dimensional toy examples with the same axes in plots A-D. The two classes are distinguished by different colors and symbols. Data that are clustered better (A) have a lower SWISS score than data where there is less separation between the classes (B; 0.25 vs. 0.91). SWISS scores can be compared even when the data are on different scales (A vs. C). Plot D is a simple shift of the data in C towards the overall mean. This small shift can have a large effect on the SWISS score (0.25 vs. 0.46).

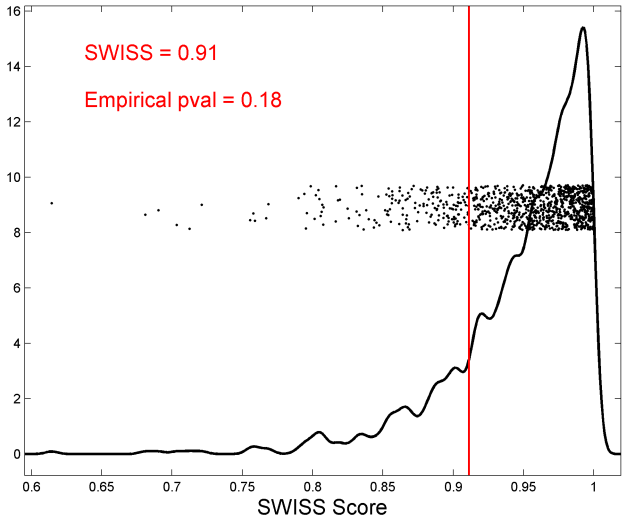

Figure 2.6: SWISS permutation test results for data shown in plot B of Figure 2.5. Each point corresponds to the SWISS score after a random permutation of the class labels. The black curve is a smooth histogram of the permuted SWISS scores. The red line shows the original SWISS score, at 0.91. The p-value is 0.18, corresponding to the proportion of permuted SWISS scores less than 0.91. Because the p-value is greater than 0.05, there is not sufficient evidence to conclude that the SWISS score is different from one.

shows the results from the permutation test, with $n_{perm} = 1000$. The x-axis shows the range of SWISS scores based upon the permuted data. Each point corresponds to the SWISS score after a random relabeling of the classes, with heights randomly jittered for visual separation. The black curve is a smooth histogram of the permuted SWISS scores. The red line at 0.91 corresponds to the original SWISS score with the true class labels. Eighteen percent of the permuted SWISS scores are smaller than the original SWISS score of 0.91, resulting in an empirical p-value of 0.18. This large p-value suggests that the SWISS score is not significantly less than one. Therefore, the clustering performance visualized in plot B of Figure 2.5 may be explained by random chance.

This permutation test is equivalent to testing for a difference between the two class means, as shown below:

$$\begin{aligned} \text{SWISS} = 1 &\Leftrightarrow \frac{\text{TWISS}}{\text{TSS}} = 1 \\ &\Leftrightarrow \text{TWISS} = \text{TSS} \\ &\Leftrightarrow \mu_X = \mu_Y \end{aligned}$$

Thus, the null hypothesis will be rejected if there is sufficient evidence that the class means are not equal. In many standard settings, it is preferable to use an exact difference of means test. For example, using the two sample t-test when $d = 1$ and each class follows a Gaussian distribution, or Hotelling's T^2 when $d > 1$ and each class follows a multivariate Gaussian distribution (Section 6.3 in Muirhead, 2005). However, it is not always practical to use these exact tests. For example, the data components may not follow a Gaussian distribution. Additionally, when dealing with genetic data, it is common that the dimension is much larger than the sample size. In this setting, calculating the inverse sample covariance matrix is challenging. As an alternative, SWISS provides a computationally easier method with no distributional assumptions to test for a difference between class means.

2.2.3 Two-sample SWISS Permutation Test

Suppose two different processing methods, say A and B, are applied to the same N data points. A permutation test for the SWISS method was developed by Cabanski *et al.* (2010) to test whether the difference in SWISS scores between Methods A and B is statistically significant. The null and alternative hypotheses are $H_0 : \text{SWISS}_A = \text{SWISS}_B$ vs. $H_1 : \text{SWISS}_A \neq \text{SWISS}_B$.

We introduce a slight change in notation for this subsection only. Specifically, let A_{ij} be a d_1-dimensional vector of covariates of the j^{th}observation ($j = 1, 2, \ldots n_i$) from the i^{th} class ($i = 1, 2$) from Method A. Relating to earlier notation,

$$A = [A_{11}, \ldots, A_{1n_1}, A_{21}, \ldots, A_{2n_2}] = [X_1(d), \ldots, X_m(d), Y_1(d), \ldots, Y_n(d)],$$

where A is the $d_1 \times N$ matrix of the A_{ij}'s, $A_{ij} = \left(A_i^{(1)}, \ldots, A_i^{(d_1)}\right)^T$, $N = n_1 + n_2 = m + n$, and the X_i's and Y_j's have been processed using Method A. Similarly, let B_{ij} be a d_2-dimensional vector of covariates of the j^{th}observation from the i^{th} class from Method B and B the $d_2 \times N$ matrix of the B_{ij}'s. Let C be the N-dimensional vector of class labels corresponding to the columns of A and B. The first n_1 elements of C will have entry 1 and the remaining n_2 elements will have entry 2. Let $\overline{W}_A$ and $\overline{W}_B$ be the d_1- and d_2-dimensional overall mean vectors, and $\bar{A}_i$ and $\bar{B}_i$ be the mean vectors of class i of A and B, respectively.

The steps of the permutation test are:

1. Form the $2\,\mathrm{x}\,N$ sum of squared deviation matrices D_A and D_B. Each column of the squared deviation matrix is the squared deviation of the corresponding data point to its respective class mean (row 1) and overall mean (row 2). Specifically, for each data point, indexed by $j = 1, 2, \ldots N$, calculate D_A and D_B as

$$D_A(1,j) = \sum_{p=1}^{d_1} \left\{ A_j^{(p)} - \overline{A}_{C(j)}^{(p)} \right\}^2$$

$$D_B(1,j) = \sum_{p=1}^{d_2} \left\{ B_j^{(p)} - \overline{B}_{C(j)}^{(p)} \right\}^2$$

$$D_A(2,j) = \sum_{p=1}^{d_1} \left\{ A_j^{(p)} - \overline{W}_A^{(p)} \right\}^2$$

$$D_B(2,j) = \sum_{p=1}^{d_2} \left\{ B_j^{(p)} - \overline{W}_B^{(p)} \right\}^2$$

The distances that are used to form D_A and D_B are best visualized by Figure 2.1 on page 6. Suppose that the data shown in Figure 2.1 have been processed using Method A. Then $D_A(1,1)$ is the squared distance between the first data point and its respective class mean (i.e., squaring the distance shown by the corresponding colored line in plot A). This squared distance is similarly calculated for the remaining data points, and these values are recorded in the first row of D_A. Note that summing over the first row of D_A gives TWISS. $D_A(2,1)$ is the squared distance between the first data point and the overall mean (i.e., squaring the distance shown by the corresponding black line in plot B). This squared distance is similarly calculated for the remaining data points, and these values are recorded in the second row of D_A. Note that summing over the second row of D_A gives TSS. Similar calculations are performed on the data that have been processed using Method B to obtain D_B.

2. Standardize each element in D_A and D_B by dividing by the corresponding TSS. That is, for $l \in \{A, B\}$,

$$D_l = \left(\frac{1}{\sum_{m=1}^{N} D_l(2,m)} \right) D_l.$$

3. Calculate the SWISS scores, the ratio of TWISS over TSS, for Methods A and B. For $l \in \{A, B\}$,

$$\mathrm{SWISS}_l = \frac{\sum_{m=1}^{N} D_l(1,m)}{\sum_{m=1}^{N} D_l(2,m)} = \sum_{m=1}^{N} D_l(1,m).$$

4. Calculate SWISS scores based upon the permuted data as follows. Let n_{perm} be the number of permutations. Setting n_{perm} to be at least 1000 is suggested.

(a) Generate an $n_{perm} \times N$ matrix R of Bernoulli random variables, assigning 0 or 1 each with probability $\frac{1}{2}$.

(b) Calculate the n_{perm}-dimensional vector P of permuted SWISS scores. For each data point in each permutation, we randomly choose which within class and overall sum of squared deviations to use (Method A or Method B). Specifically, for permutation $i = 1, 2, \ldots n_{perm}$, let the i^{th} entry of P be

$$P(i) = \frac{\sum_{m=1}^{N} [R(i,m) * D_A(1,m) + (1 - R(i,m)) * D_B(1,m)]}{\sum_{m=1}^{N} [R(i,m) * D_A(2,m) + (1 - R(i,m)) * D_B(2,m)]}$$

Note that for each permutation, neither the mean vectors nor the sum of squared deviations are recalculated. Thus, it is possible to have a permutation where TSS is actually smaller than TWISS, and hence, the permuted ratio is larger than 1.

5. Finally, calculate the empirical p-values. Two p-values are reported to indicate the behavior on each side of this naturally two-sided test. Figure 2.7 demonstrates how the empirical p-values are calculated. This figure shows the permutation test output when comparing the data visualized in plots A and D of Figure 2.5. We will assume that this data come from one dataset processed by two different methods, Method A and Method B, respectively. The SWISS scores are 0.25 and 0.46, and are reported in the upper left corner of the plot. The x-axis shows the range of SWISS scores based upon the permuted data. The red and blue lines represent the SWISS scores of the two methods. The dots show the distribution of the permuted SWISS scores (with jittered heights), which range from 0.20 to 0.55. The black line shows a smooth histogram of these dots. To calculate the empirical p-values, take the proportion of dots outside the red and blue lines. That is, for $l \in \{A, B\}$,

$$\text{p-value}_l = \min \left\{ \frac{\#\text{ of } P(i) < \text{SWISS}_l}{n_{perm}}, 1 - \frac{\#\text{ of } P(i) < \text{SWISS}_l}{n_{perm}} \right\}.$$

The lower p-value is 0.03 because 3% of all SWISS scores from the permuted data is less than the lower SWISS score of 0.25. Similarly, the upper p-value is 0.04, corresponding to the 4% of permuted SWISS scores greater than the larger SWISS score of 0.46. These empirical p-values are reported in

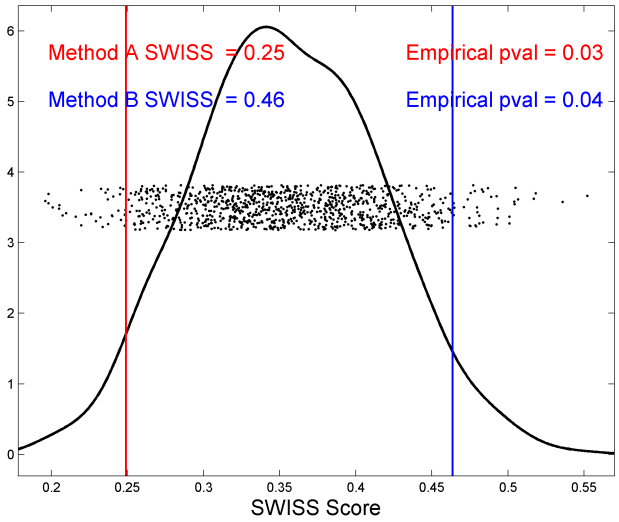

Figure 2.7: SWISS permutation test results testing for a significant difference in SWISS scores between the toy examples shown in plots A and D of Figure 2.5. This plot shows the distribution of the permuted SWISS scores (dots), summarized by a smooth histogram (black curve), along with the SWISS scores of Method A (red vertical line) and Method B (blue vertical line). The SWISS scores and corresponding empirical p-values are also reported. Because the sum of the p-values are greater than 0.05, conclude that there is no significant difference between the SWISS scores of Methods A and B.

the upper right corner of the plot.

Reject the null hypothesis that the two SWISS scores are equal at significance level α if the sum of the upper and lower p-values is less than α. Otherwise, there is not sufficient evidence to conclude a difference between the SWISS scores of Method A and Method B. For the toy example results shown in Figure 2.7, we conclude, at the 5% level of significance, that the difference between the SWISS scores of Methods A and B (0.25 and 0.46, respectively) may be due to random chance because the sum of the p-values is 0.07.

This permutation test can also be performed as a one-sided test. In this case, the hypotheses are $H_0 : \text{SWISS}_A = \text{SWISS}_B$ vs. $H_1 : \text{SWISS}_A < \text{SWISS}_B$. If the proportion of points less than Method A's SWISS score is less than α, we conclude that the SWISS score of Method A is significantly less than the SWISS score of Method B, and, hence, *significantly better* at clustering the data into the predefined classes than Method B. For the toy example results shown in Figure 2.7, the corresponding one-sided p-value is 0.03. Therefore, the SWISS score of Method A is significantly less than the SWISS score of Method B at $\alpha = 0.05$.

2.3 Multiclass SWISS

So far, we have only considered the case where the data consist of two classes. This section will extend the SWISS method to the multiclass setting.

When Cabanski *et al.* (2010) proposed SWISS, they dealt with the multiclass case in the same manner as the two class case. That is, TSS is calculated by summing the squared distance between each point and the overall mean, and TWISS is calculated by summing the squared distances between each point and its class mean. We will refer to the SWISS score calculated in this manner as the "original SWISS score". Figure 2.8 shows a two dimensional toy example that motivates the need for a multiclass correction. Plot A shows the data split into two classes, denoted by red x's and blue o's. The colored lines show the distance between each point and its class mean. The original SWISS score of this dataset (shown at the top of the plot) is 0.27, which reflects the good clustering of this dataset.

Plot B shows the same data as in plot A. However, the blue o's have now been divided into two classes, blue o's and green +'s. The TSS (the sum of squares in the denominator of SWISS) has not changed since the points did not move, only the class labels were changed. The TWISS of plot B is slightly smaller than the TWISS of plot A. The original SWISS score of the data in plot B is also 0.27; however, this clustering is of distinctly lower quality because the blue and green classes overlap.

Plot C shows a modification of the data in plot B, where all of the green +'s have been shifted up and to the left by the same amount. Now there is clear separation between all of the classes. TWISS did not change from plot B to plot C since each point is still the same distance from its class mean. TSS is different in plots B and C, though this difference is very small. Thus, the original SWISS score of this dataset is also 0.27. This is not an appealing property of SWISS because the clusters in plot C are clearly more distinct than in plot B. SWISS feels the amount of variation within clusters and distances from the overall mean, but it does not take into account the distances between classes. Thus, the SWISS score can be substantially improved when dealing with multiclass data.

We have extended the SWISS method discussed in Subsection 2.2.1 to the multiclass setting by calculating the original SWISS score for each pair of classes, then taking the *average of all pairwise SWISS scores*. The plots in Figure 2.8 also report the average pairwise SWISS scores. The original and average pairwise SWISS scores for plot A are equal because there are only two classes. When

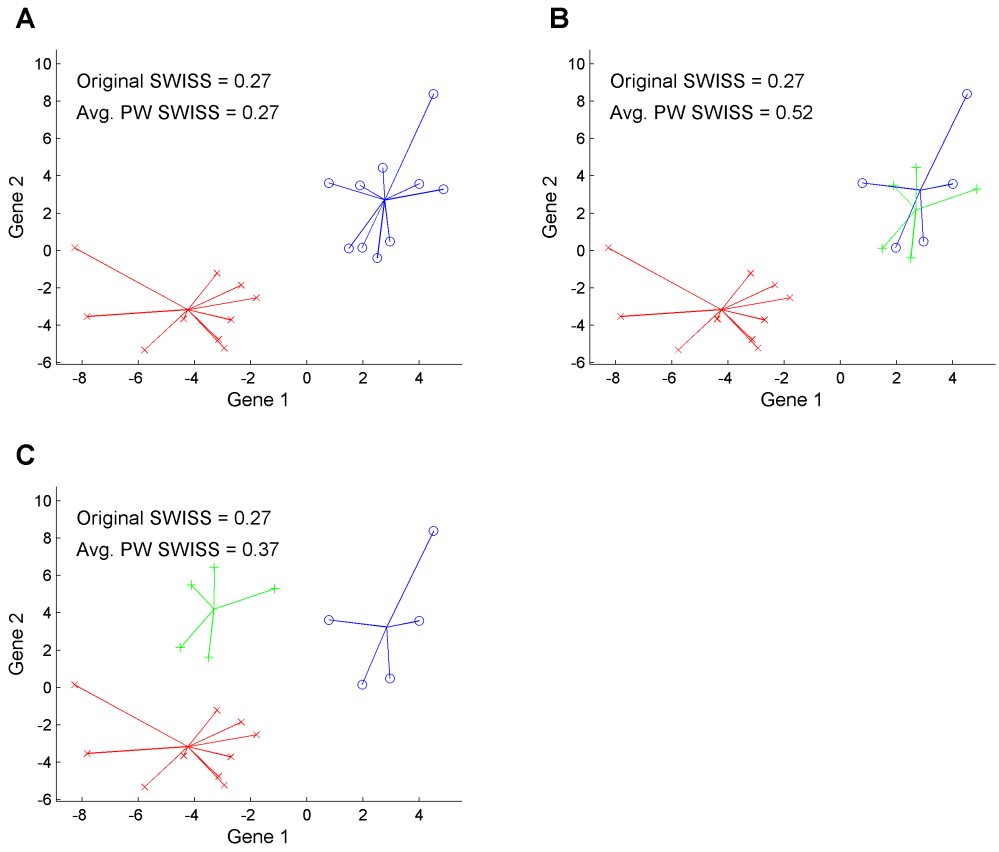

Figure 2.8: Two dimensional toy example showing the need for a multiclass correction of SWISS. Plot A shows the data split into two classes, denoted by different colors and symbols. Plot B shows the same data as in plot A; however, the data have now been split into three classes. Plot C shows a modified version of the data in plot B where the green +'s have been shifted up and to the left. The colored lines show the distance between each point and its class mean. The original SWISS score is reported at the top of each plot, and is the same for all three plots, although plots A and C clearly have more distinct clusters than plot B. The plots also report the average pairwise SWISS scores, which are much lower for plots A and C compared to plot B.

comparing the average pairwise SWISS scores of plots B and C, we see that plot C has a lower SWISS score than plot B (0.37 vs. 0.52). This average pairwise correction seems appropriate because the data with clear separation between the classes (plot C) now have a lower SWISS score than the data with overlapping classes (plot B).

For the remainder of this chapter, when calculating the SWISS score, we will use the original SWISS score described in Subsection 2.2.1 when the data consist of two classes and the average pairwise SWISS score when the data consist of more than two classes.

2.4 Comparison with Competing Methods

As mentioned in Subsection 2.2.1, there are several useful properties of SWISS. Because SWISS is unit free, it can be used to compare methods that are on different scales. For example, different scales can arise from differing normalization methods or when comparing different microarray platforms. SWISS is also shift, scale and rotation invariant. Additionally, because it is based only on Euclidean distance and normalized by TSS, SWISS can be used to compare methods that have different dimensions. This can be useful when comparing the same biological samples, but using two different gene sets. Finally, because the permutation test reports a p-value, investigators are able to decide which processing method is preferred, or learn if there is no statistically significant difference between the two methods, without relying on subjective evaluation.

Using the within class sum of squares to compare how well data are clustered has previously appeared in the literature. For instance, Kaufman and Rousseeuw (1990) use within class sum of squares (which they refer to as WCSS) as a tool to aid in the decision of the number of clusters that should be used for k-means clustering. Giancarlo *et al.* (2008) show WCSS can provide the basis of a reasonable method for choosing k. However, because WCSS is not standardized, it can only be used to compare the effectiveness of clustering methods when the total sum of squares is constant. Thus, WCSS is not able to compare the effectiveness of clustering on two different processing methods when the processed data given by those methods are on different scales (i.e., have different TSS).

Dudoit *et al.* (2002) propose a statistic, the BW ratio, as a gene filtering tool. The BW ratio is the ratio of the between class sum of squares (BCSS) to within class sum of squares (TWISS). Because

$\text{TSS} = \text{TWISS} + \text{BCSS}$, we can rewrite the BW ratio as

$$\text{BW} = \frac{\text{TSS} - \text{TWISS}}{\text{TWISS}} = \frac{\text{TSS}}{\text{TWISS}} - 1 = \frac{1}{\text{SWISS}} - 1.$$

This shows that a large SWISS score corresponds to a small BW ratio, and vice versa. Unlike SWISS, the BW ratio is not bound between zero and one. Dudoit *et al.* filter out genes with near constant expression levels by selecting the genes with the largest BW ratio (i.e., smallest SWISS). The authors did not apply the BW ratio to other settings beyond gene filtering.

Davies and Bouldin (1979) propose a metric, called the Davies-Bouldin index (DBI), for evaluating clustering algorithms that is very similar to SWISS. Using the same notation as SWISS, defined in Subsection 2.2.1, they define $R = \sqrt{\text{TWISS}}/M$, where $M = \sqrt{\sum_{p=1}^{d} \left\{ \overline{X}^{(p)} - \overline{Y}^{(p)} \right\}^2}$ is the Euclidean distance between the class centroids. In the two class setting, $\text{DBI} = R$, which appears similar to the square root of SWISS with a different normalization factor. Instead of normalizing by TSS, Davies and Bouldin choose to normalize TWISS by the squared distance between the class centroids. Unlike SWISS, which is bounded by $[0,1]$, DBI is unbounded above. Similar to SWISS, DBI is unit free and shift, scale and rotation invariant. In the multiclass setting, DBI is calculated in a very different manner than the multiclass SWISS score. We will index R as R_{ij} when R is calculated between a pair of classes, i and j. They define $D_i = \max_{j:i \neq j} R_{ij}$ and $\text{DBI} = K^{-1} \sum_{i=1}^{K} D_i$, where K is the total number of classes. For each class, D_i equals R_{ij} for the most similar class j, which is similar to choosing the "worst case scenario". This is very different from the multiclass SWISS, which is calculated by taking the average of all pairwise SWISS scores. Although calculated differently from SWISS, DBI has the same interpretation that the smaller the value, the better the clustering of the data. DBI has been applied in deciding the value of k in the k-means clustering algorithm when the true value of k is unknown.

Table 2.1 shows both the SWISS score and Davies-Bouldin index for the toy examples considered in Figure 2.8. Notice that while SWISS is bound by zero and one, DBI achieves values larger than one. Both SWISS and DBI assign a large value to the data in plot B, which reflects poor clustering of the data. Both methods assign the lowest value to the data in plot A, with the data in plot C receiving only a slightly larger value. From this comparison, it appears that SWISS and DBI are feeling similar

	SWISS	DBI
Plot A	0.27	1.37
Plot B	0.52	5.97
Plot C	0.37	1.40

Table 2.1: Comparison of the SWISS score and Davies-Bouldin index (DBI) for the three toy example datasets visualized in Figure 2.8.

aspects of the data.

To our knowledge, there are no methods currently in the literature, including WCSS, the BW ratio and DBI, that have been applied to address the variety of problems that SWISS can tackle. However, there are methods that can be compared to SWISS when addressing specific problems. As an example, in Subsection 2.5.1, we perform a SWISS analysis comparing two different Affymetrix microarray platform pre-processing methods. In that subsection, we will also compare our SWISS method with other methods used in evaluating Affymetrix pre-processing and normalization methods.

2.5 Applications

The usefulness of SWISS will be demonstrated by two different applications. The first application, in Subsection 2.5.1, compares two different pre-processing methods on the Affymetrix microarray platform. The second application, discussed in Subsection 2.5.2, compares two different microarray experimental designs using SWISS.

2.5.1 Comparing Affymetrix Pre-processing Methods

An important step in processing microarray data is to produce a single value for the gene expression level of an RNA transcript using one of a growing number of statistical methods. In this application, we use SWISS to compare two different Affymetrix pre-processing methods: RMA (Irizarry *et al.*, 2003) and the Affymetrix Micro Array Suite 5.0 (MAS 5.0; Affymetrix White Paper, 2002). The data contain 5 replicates of 4 different samples, for a total of 20 data points. Each sample will be considered a class, resulting in 4 classes with 5 points in each class. For a further description of the dataset, see see section "Microarray Experiments, Data Collection and Processing: Experimental Application II" of Cabanski *et al.* (2010).

The SWISS scores of RMA and MAS 5.0 are 0.125 and 0.613, respectively. The two-sample permutation test gives a p-value of 0.04. At the 5% level of significance, there is sufficient evidence to conclude that the two SWISS scores are not equal. Therefore, RMA is preferred over MAS 5.0 because its SWISS score is significantly lower. Because each class consists of multiple replicates of the same experimental sample, this SWISS analysis shows that RMA gives the most reproducible results.

Millenaar *et al.* (2006) compare six different pre-processing algorithms used to generate gene expression levels for the Affymetrix microarray platform, including RMA and MAS 5.0. The authors make their comparison using five different analyses:

1. Differential expression comparison. A two-sample t-test was performed to detect genes with differential expression between two classes. The authors analyzed the number of significantly called genes for each method and the amount of overlap between all methods.

2. Spike-in comparison. A commonly used RNA spike-in experiment from Affymetrix was used to test which method produced the most accurate results. For this comparison, spike-in dilution series data are needed. Fit the linear model $\log_2 E = \beta_0 + \beta_1 \log_2 c + \varepsilon$ where E is a vector of expression measures and c is a vector of concentrations. If the method is unbiased, the slope (value of β_1) should be near 1.

3. Reproducibility. The coefficient of variation (CV), defined as $\frac{\text{standard deviation}}{\text{average}} \times 100\%$, is a measure of reproducibility which is independent of the mean. CV should be calculated for all genes across all arrays. Methods that produce many gene expression estimates with low CV give more reproducible results.

4. Biological relevance. Compare the genes called as being differentially expressed to genes known to vary across conditions.

5. Comparison with Real Time RT-PCR data. To further test which method calculates microarray gene expression most accurately, the results of the methods can be compared with Real Time RT-PCR (Provenzano and Mocellin, 2007) on a subset of genes which are predicted by all the methods to be differentially regulated.

Similar to SWISS, Millenaar *et al.* conclude that RMA gives the most reproducible results. However, there are drawbacks to using their above comparisons. First, some of the comparisons, such as the spike-in and Real Time RT-PCR comparisons, require additional experiments to be performed. Second, in order to assess biological relevance, a fair amount of previous knowledge about the dataset is required. Not only does this comparison rely on similar datasets having already been analyzed, but it may also be very time consuming. Third, most of the above comparisons are qualitative rather than quantitative. Additionally, hypothesis tests are not performed on the few analyses that are quantitative. As previously mentioned, SWISS does not require additional experiments to be performed, is quantitative rather than qualitative and a permutation test based on SWISS helps to identify whether one method should be preferred over another.

2.5.2 Comparing Microarray Experimental Designs

Two-color gene expression microarray assays are among the most common genomic profiling tools currently in use (Churchill, 2002; Vinciotti *et al.*, 2005). Two-color array technologies rely on labeling two samples (such as tumor vs. normal or experimental vs. reference) with different fluorochromes followed by co-hybridization to the same chip-based assay (Meyerson and Hayes, 2005). Considering relative fluorescence (such as a log-ratio), particularly to a common reference such as a cell line reference hybridized on the same array, provides a robust normalization technique to control for manufacturing variability (Novoradovskaya *et al.*, 2004). A two-color array with a common reference such as a cell line will be referred to as a *reference design*. A one-color array or a two-color array using only one signal channel will be referred to as a *single-channel design*.

The reference design, while powerful, has its disadvantages (Churchill, 2002); notably, 50% of the measurements in a reference design experiment are solely for normalization purposes, representing both significant financial and opportunity costs. Additionally, there is an effective doubling in measurement error by the reference design because every $\log_2$ ratio of experimental channel intensity over reference channel intensity includes error contributions from both channels (Churchill, 2002; Vinciotti *et al.*, 2005). Furthermore, genes that are biologically absent or expressed at very low levels in the reference sample are sometimes excluded from consideration even if present at high levels in the experimental sample, which likely reduces the information content of the experiment. In con-

trast to the two-color arrays, one-color arrays do not rely on experimental normalization such as that described for the reference design. Instead, computational techniques are used to normalize fluorescence intensities across arrays. Historically, the distinction between the one and two-color platforms has been viewed primarily in terms of the technology underlying the manufacturing and experimental protocols of the array platform (Meyerson and Hayes, 2005).

SWISS is used to compare the reference design to the single channel design using 39 breast cancer samples (Sorlie *et al.*, 2001) assayed on spotted cDNA arrays, a relatively old microarray platform. Data representing both the reference design and single channel design were obtained for these 39 samples, followed by a normalization step (for specifics, see section "Microarray Experiments, Data Collection and Processing: Experimental Application I" of Cabanski *et al*., 2010). The two classes for this analysis are estrogen receptor positive and negative phenotypes.

Rather than only comparing SWISS scores for a fixed number of genes, the number of genes in the analysis is varied and SWISS scores are calculated. For each experimental design, the gene set is filtered by keeping the genes with the largest variances across all arrays. Note that the gene lists of the reference and single channel designs are decided independently, and thus the gene sets may not be identical. SWISS scores are calculated starting with only 16 genes and ending with all genes included (approximately 8,000). Figure 2.9 shows the results, along with the 90% confidence intervals (black bars) from the SWISS permutation test. This shows that the reference design always has a lower SWISS score and, because the red and blue curves always lie outside of the 90% confidence interval, that the reference design is always statistically better than the single channel design.

Figure 2.9 compares the SWISS scores of the reference and single channel designs when we filter by gene variance across all arrays. As previously noted, different gene sets may have been compared because the most variable genes in the reference design did not necessarily coincide with the most variable single channel design genes. Next, the reference and single channel designs are compared using the same gene sets. Figure 2.10 shows this comparison, along with the corresponding 90% confidence intervals from the permutation test. Plot A compares the two designs using the most variable genes from the single channel design, and plot B uses the most variable genes from the reference design. Note that the blue dashed line (single channel design) from plot A and the solid red line (reference design) from plot B are the same lines shown in Figure 2.9.

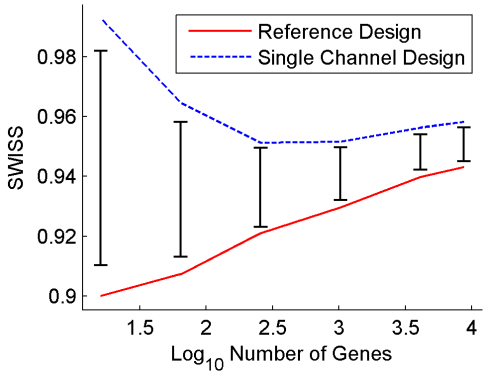

Figure 2.9: SWISS scores for the reference design (solid red) with reference design gene filtering and single channel design (dashed blue) with single channel design gene filtering along with corresponding 90% confidence intervals (black bars) calculated from the SWISS permutation test are shown. The reference design is always significantly better than the single channel design because the black bars are always inside the blue and red curves. Figure reproduced from Cabanski *et al.* (2010).

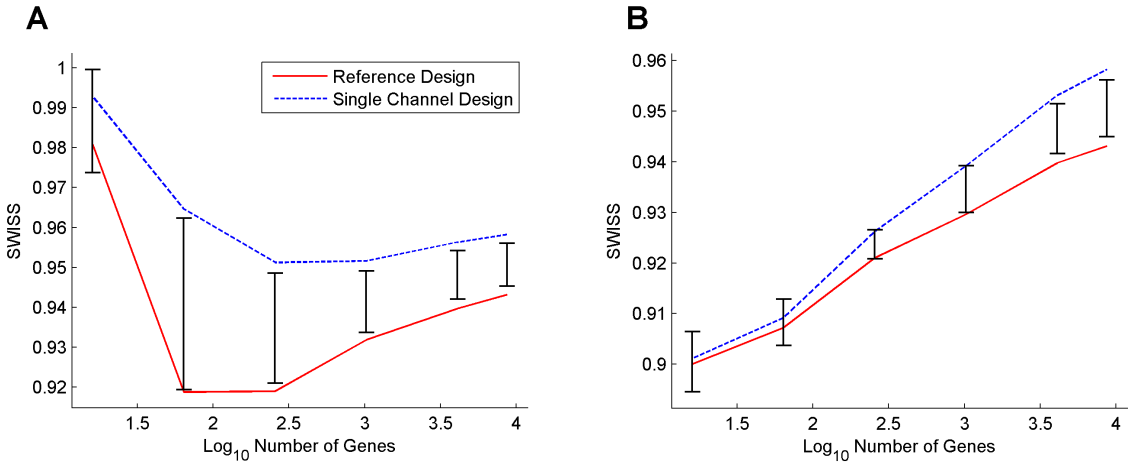

Figure 2.10: The SWISS scores for the reference design (solid red) and single channel design (dashed blue) along with corresponding 90% confidence intervals (black bars) calculated from the SWISS permutation test are shown. The genes for both designs in plot A are filtered according to variance across all arrays in the single channel design, and the genes in plot B are filtered according to variance across all arrays in the reference design. Figure reproduced from Cabanski *et al.* (2010).

Remember that in Figure 2.9, the reference design was always significantly better than the single channel design. However, in plots A and B of Figure 2.10, neither gene filtering exhibits a uniformly significant difference between the two designs. This is especially apparent in plot B, where there is no significant difference between the reference and single channel designs until at least 1000 genes are included. Note that both the red and blue curves in plots A and B have similar shapes, as opposed to Figure 2.9 where the curves had different shapes. This suggests that for two different gene filterings, the SWISS curves may have different shapes (as in Figure 2.9). However, when the same gene set is used on both curves, the SWISS curves seem to have similar shapes (as in Figure 2.10).

When comparing plots A and B of Figure 2.10, notice that the SWISS scores in plot B are always less than or equal to the SWISS scores in plot A. This implies that the filtering method determined by choosing the most variable genes in the reference design is superior to choosing the most variable genes in the single channel design. Furthermore, single channel filtering leads to higher SWISS scores, suggesting that the most variable single channel genes contain more noise and less signal. Therefore, for spotted cDNA arrays, the reference design seems best for filtering genes by variation across arrays. However, once the gene set has been selected, the single channel design is not statistically worse than the reference design in terms of clustering performance for up to 1000 genes.

2.6 Other Considerations

While stressing the broad applicability of SWISS to a range of analytical problems and the ease of its use, it is important to understand the limitations as well (Cabanski *et al.*, 2010). First, high-dimensional data such as microarray experiments are the product of complex protocols and depend on the quality of reagents and samples. Any change in upstream elements, such as a lab protocol or normalization method, might influence the resulting SWISS score dramatically. Additionally, SWISS scores may not represent the only criterion on which one method is preferred. For example, we showed in Subsection 2.5.1 that RMA gives more reproducible results than MAS 5.0. However, some investigators may prefer using MAS 5.0 because it is more conservative, gives positive output values, down-weights outliers and minimizes bias (Affymetrix Technical Note, 2005).

We also acknowledge that while SWISS is convenient to implement across a broad set of analyses,

there are likely cases where more dedicated methods would provide more nuanced insights. For example, SWISS should not be the only tool used when the investigator is interested in performing an in-depth analysis of competing methods or platforms, such as comparing a new normalization method with other established normalization methods.

We also acknowledge that the examples we have provided in Section 2.5 should be viewed with caution in terms of the potential to introduce bias. Decisions such as gene filtering and cross-platform gene annotation may influence the interpretation of the results, as documented in Figures 2.9 and 2.10. Depending on which set of filtered genes is used, an investigator may reach different conclusions about the superiority of a platform. More troubling, the set of assumptions, such as filtering, may be based on the performance of those genes in one platform over the genes that might have been chosen by the other, ultimately biasing the analysis to favor a potentially spurious result. We have shown in Subsection 2.5.2 examples in which one might be tempted to select what appears to be optimal gene sets based on SWISS scores. This is not the intention of the examples but rather to demonstrate the opposite: to document the impact on SWISS scores by varying gene lists. While SWISS does not necessarily account for the full range of potential biases, it does allow for decisions about data transformations such as gene filtering to be made independently for each data source.

When approached with these cautions in mind, we feel that the concerns raised in this section are offset by the broad scope of applicability that the SWISS method offers.

Chapter 3

High-dimensional Asymptotic Behavior of SWISS

As discussed in Chapter 2, Standardized WithIn class Sum of Squares (SWISS, defined in Subsection 2.2.1) has proven itself to be useful in evaluating different processing methods of high-dimensional genetic data in terms of their clustering performance on predefined classes, such as tumor subtypes. This motivates this study of the high-dimensional asymptotic behavior of SWISS. We will explore an asymptotic representation of the SWISS score of two class data as the dimension d grows. Specifically, if we denote m and n as the sample size of each class and $N = m+n$ as the total sample size, we will describe the asymptotic behavior of the SWISS score as $\frac{d}{N} \to \infty$. This asymptotic domain contains two different settings, both of which will receive attention.

The first setting, known as High Dimension Low Sample Size (HDLSS), occurs when $d \to \infty$ and N is fixed. The first paper to consider this asymptotic domain was Casella and Hwang (1982). Hall *et al.* (2005) describe a geometric representation of HDLSS data where d tends to infinity while the sample size N is fixed. In particular, under certain assumptions, the data tend to lie deterministically at the vertices of a regular simplex with all of the randomness appearing in the rotation of this simplex. This geometric representation provides the insight needed to understand the behavior of the SWISS score in that high-dimensional setting.

The second setting, which we will refer to as High Dimension Moderate Sample Size (HDMSS), allows N to also grow, but at a much slower rate than d so that $\frac{d}{N} \to \infty$. This setting has received attention in the literature, but often as specific cases rather than as a general setting. For example, Fan and Lv (2008) study asymptotics where $N \to \infty$ and $d \sim e^N$. In contrast, we choose to let $d \to \infty$ and let N follow, which views the Fan and Lv special case as $d \to \infty$ and $N \sim \log(d)$. While this notation

is a sharp contrast to classical mathematical statistics, it feels more natural and allows for a more transparent relationship to the HDLSS setting. Yata and Aoshima (2012) show that, under certain conditions, the same geometric representation of HDLSS data still remains when $N \to \infty$. Under these same conditions, we are able to extend the results of the behavior of SWISS in the HDLSS setting to the HDMSS setting. In fact, the asymptotic behavior of SWISS is easier to interpret in this setting as the results now depend on two fewer parameters because m and n disappear in the limit.

This chapter is laid out as follows. In Section 3.1, the geometric representation of HDLSS data, first described by Hall *et al.* (2005), is summarized. Section 3.2 gives a precise and detailed description of the asymptotic behavior of the SWISS score in the HDLSS setting. Because the asymptotic representation of the SWISS score is a function of multiple parameters, Section 3.2 will also give insight to the SWISS score by studying important special cases. Section 3.3 will describe and interpret the asymptotic behavior of the SWISS score in the HDMSS setting. Finally, Section 3.4 describes future research problems related to the high-dimensional asymptotic behavior of the SWISS score.

3.1 Geometric Representation of High-dimensional Data

This section starts with a summary of the work of Hall *et al.* (2005) who describe, under some assumptions, a geometric representation of data where the dimension d tends to infinity while the sample size is fixed. They find that there is a tendency for the data to lie deterministically at the vertices of a regular simplex, where essentially all of the randomness in the data appears only as a rotation of this simplex. However, the results in Hall *et al.* require the entries of the data vector to be nearly independent, in the sense that when they are viewed as a time series, these entries must satisfy a ρ-mixing condition (Kolmogorov and Rozanov, 1960). Ahn *et al.* (2007), Jung and Marron (2009), Qiao *et al.* (2010) and Yata and Aoshima (2012) give much milder conditions for the results of Hall *et al.* to represent one HDLSS sample using asymptotic properties of the sample covariance matrix and its eigenvalues. Yata and Aoshima also extend these results and show that, under some additional assumptions, the same geometric representation of the data holds in the HDMSS setting. This section reports the results from the aforementioned papers that will be useful in determining the asymptotic representation of the SWISS score in the HDLSS and HDMSS settings.

The notation that follows will be used throughout the rest of this chapter. Assume that the data consist of two classes, $\mathscr{X}(d) = \{X_1(d), \ldots, X_m(d)\}$ and $\mathscr{Y}(d) = \{Y_1(d), \ldots, Y_n(d)\}$, where the data vectors $X_i(d) = \left(X_i^{(1)}, \ldots, X_i^{(d)}\right)^{\mathrm{T}}$ and $Y_j(d) = \left(Y_j^{(1)}, \ldots, Y_j^{(d)}\right)^{\mathrm{T}}$ are independent and identically distributed from a d-dimensional multivariate distribution with positive definite covariance matrix Σ_d^X and Σ_d^Y, respectively. Without loss of generality, assume that each $X_i(d)$ and $Y_j(d)$ has zero mean.

Denote the $d \times d$ sample covariance matrix of $\mathbf{X}_d = [X_1(d), \ldots, X_m(d)]$ as $\mathbf{S}_d^X = m^{-1}\mathbf{X}_d\mathbf{X}_d^T$ and the $m \times m$ dual sample covariance covariance matrix as $\mathbf{S}_{D,d}^X = m^{-1}\mathbf{X}_d^T\mathbf{X}_d$. An advantage of considering the dual covariance matrix is that $\mathbf{S}_d^X$ and $\mathbf{S}_{D,d}^X$ share non-zero eigenvalues. The eigenvalue decomposition of Σ_d^X is $\Sigma_d^X = V_d\Lambda_d^X V_d^T$, where $\Lambda_d^X = diag\left\{\lambda_1^X, \ldots, \lambda_d^X\right\}$ is the eigenvalue diagonal matrix. Define the average of the eigenvalues as $\sigma_d^2 = d^{-1}\sum_{i=1}^d \lambda_i^X$.

Write $\mathbf{X}_d = V_d\left(\Lambda_d^X\right)^{1/2}\mathbf{Z}_d$, where $\mathbf{Z}_d = \left(\Lambda_d^X\right)^{-1/2}V_d^T\mathbf{X}_d$ is a $d \times m$ sphered data matrix from a distribution with zero mean and identity covariance matrix. We will represent the elements of $\mathbf{Z}_d = [\mathbf{z}_1, \ldots, \mathbf{z}_d]^T$ as $\mathbf{z}_i = (z_{i1}, \ldots, z_{im})^T$, $i = 1, \ldots, d$. Define $\phi_{ij} = \mathrm{Cov}\left(z_i^2 - 1, z_j^2 - 1\right)$.

Let us also write $D_k = \left(\sum_{i=1}^d \lambda_i^X\right)^{-1}\sum_{i=1}^d \lambda_i^X z_{ik}$ as a diagonal element of $m\left(\sum_{i=1}^d \lambda_i^X\right)^{-1}\mathbf{S}_{D,d}$. Note that $D_k = \|X_k(d)\|^2 / \mathrm{tr}\left(\Sigma_d^X\right)$ and $E(D_k) = 1$ (Yata and Aoshima, 2012).

We assume the following:

A1. Each column of $\mathbf{X}_d$ has zero mean and covariance matrix Σ_d^X.

A2. The fourth moments of each column of $\mathbf{Z}$ are uniformly bounded.

A3. The eigenvalues of Σ_d^X are sufficiently diffused, in the sense that

$$\varepsilon_d^X = \frac{\sum_{i=1}^d \left(\lambda_i^X\right)^2}{\left(\sum_{i=1}^d \lambda_i^X\right)^2} \to 0 \text{ as } d \to \infty.$$

The statistic ε_d^X can be viewed as a measure of the sphericity of the data matrix. This is related to the statistic $\varepsilon = \left(d\varepsilon_d^X\right)^{-1}$, which was proposed by John (1972) as the basis of a hypothesis test for equality of eigenvalues.

A4. The sum of the eigenvalues of Σ_d^X is the same order as d, i.e.,

$$\sigma_d^2 = d^{-1}\sum_{i=1}^d \lambda_i^X = O(1).$$

A5. The dependence between components of $\mathbf{Z}_d$ must be small, as regulated by a covariance condition

$$\mathrm{Var}\left(D_k\right) = \frac{E\left\{\left(\sum_{i=1}^{d}\lambda_i^X\left(z_{ik}^2-1\right)\right)^2\right\}}{\left(\sum_{i=1}^{d}\lambda_i^X\right)^2} = \frac{\sum_{i,j}\lambda_i^X\lambda_j^X\phi_{ij}}{\left(\sum_{i=1}^{d}\lambda_i^X\right)^2} \to 0 \text{ as } d\to\infty.$$

This condition says that the variance of D_k goes to zero as the dimension grows.

Remark 1. Suppose that assumption A3 holds. Then assumption A5 also holds if $\mathbf{X}_d$ is Gaussian (Yata and Aoshima, 2012). A5 fails to hold in cases when the components of $\mathbf{X}_d$ are uncorrelated but dependent. Examples when A5 does not hold are when $\mathbf{X}_d$ follows a scale mixture of Gaussian distributions (details shown in Example 1, below) or when $\mathbf{X}_d$ follows a multivariate t-distribution with mean zero, covariance matrix $\mathbf{I}_d$ and degrees of freedom ν (Yata and Aoshima, 2012).

A1-A4 are from Ahn *et al.* (2007) and A5 is from Yata and Aoshima (2012). One main result of Ahn *et al.* is that, under A1-A4, the sample eigenvalues behave as if they follow an identity covariance matrix in the sense that

$$\frac{m}{\sum_{i=1}^{d}\lambda_i^X}\mathbf{S}_{D,d}^X \to \mathbf{I}_m$$

in probability as $d\to\infty$ for a fixed m. Yata and Aoshima (2012) show that the above result still holds under A1-A5. Later in this section, Example 1 describes why the additional assumption A5 is needed.

Now assume that the above conditions also hold for the n-point sample $\mathscr{Y}(d)$. In particular, let the average of the eigenvalues be defined as $\tau_d^2 = d^{-1}\sum_{i=1}^{d}\lambda_i^Y$. When the eigenvalues for both classes are sufficiently diffused, i.e., $\varepsilon_d^X\to 0$ and $\varepsilon_d^Y\to 0$ as $d\to\infty$, then the pairwise squared distances between any two points in the same class are approximately equal (Qiao *et al.*, 2010). Specifically,

$$\frac{1}{d\sigma_d^2}\sum_{k=1}^{d}\left(X_i^{(k)}-X_j^{(k)}\right)^2 \to 2 \text{ as } d\to\infty$$

$$\frac{1}{d\tau_d^2}\sum_{k=1}^{d}\left(Y_i^{(k)}-Y_j^{(k)}\right)^2 \to 2 \text{ as } d\to\infty.$$

Assume that, after scaling by d^{-1}, the squared distance between the means is a constant μ^2,

$$d^{-1}\sum_{k=1}^{d}\left\{E\left(X^{(k)}\right)-E\left(Y^{(k)}\right)\right\}^2 \to \mu^2.$$

For convenience, assume that the limiting average eigenvalues exist; $\sigma_d^2 \to \sigma^2$ and $\tau_d^2 \to \tau^2$ as $d \to \infty$. Qiao *et al.* also show, using a weak law of large numbers, that the distance between $X_i(d)$ and $Y_j(d)$, divided by $d^{1/2}$, converges in probability to $\left(\sigma^2 + \tau^2 + \mu^2\right)^{1/2}$ as $d \to \infty$:

$$\frac{1}{d^{1/2}} \left\{ \sum_{k=1}^{d} \left(X_i^{(k)} - Y_j^{(k)} \right)^2 \right\}^{1/2} \to c \equiv \left(\sigma^2 + \tau^2 + \mu^2\right)^{1/2}.$$

The above statement says that if each of the two samples is sufficiently spherical as $d \to \infty$, the pairwise average distance between any two data vectors from each sample is approximately constant.

Letting $N = m + n$, the following geometric picture of the two classes, $\mathscr{X}(d)$ and $\mathscr{Y}(d)$, for large d and fixed m and n, is obtained. The N points are asymptotically located at the vertices of a N-polyhedron in $(N-1)$-dimensional space. The m points of $\mathscr{X}(d)$ are the vertices of an m-simplex with edge length $\left(2d\sigma_d^2\right)^{1/2}$. The other n points of $\mathscr{Y}(d)$ are the vertices of an n-simplex with edge length $\left(2d\tau_d^2\right)^{1/2}$. The lengths of the edges between a vertex from a point in $\mathscr{X}(d)$ and one from a point in $\mathscr{Y}(d)$ are all of length $cd^{1/2}$. Almost all of the stochastic variability in the data goes into random rotation, although some goes into random perturbations of the vertices that disappear as $d \to \infty$.

Figure 3.1 shows the geometric representation of an HDLSS toy example. There are three points in the $\mathscr{X}$ class ($m = 3$) and one point in the $\mathscr{Y}$ class ($n = 1$). For large d, the three points from the $\mathscr{X}$ class will form a simplex with edge length $\left(2d\sigma_d^2\right)^{1/2}$. The lengths of edges between a point in $\mathscr{X}$ and the $\mathscr{Y}$ point are all of length $cd^{1/2}$, where $c = \left(\sigma_d^2 + \tau_d^2 + \mu^2\right)^{1/2}$.

It was pointed out by John Kent, through a counter example, that an additional assumption beyond A1-A4 is needed for the geometric representation to hold. This example was first discussed in the literature by Jung and Marron (2009).

Example 1. (Strong dependency via a scale mixture of Gaussians). Suppose that $\mathbf{X} = [X_1, \ldots, X_n]$ follows a mixture of Gaussian distributions. Specifically, let $X_i \sim V_1 U + \sigma V_2 (1 - U)$, where V_1, V_2 are two independent $N_d(0, I_d)$ random variables, $U \sim$ Bernoulli $\left(\frac{1}{2}\right)$ and independent of V_1 and V_2, and $\sigma > 1$.

Note that $\mathrm{Cov}(\mathbf{X}) = \left(\frac{1+\sigma^2}{2}\right) I_d$, so condition A3 holds and the variables are uncorrelated. Therefore, A1-A4 are satisfied. However, even though the X_i's are uncorrelated, they are not independent.

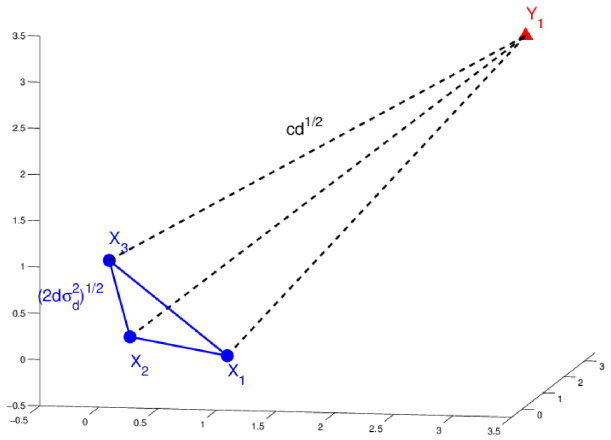

Figure 3.1: Two class toy example showing the geometric representation in the HDLSS setting. There are three points in the $\mathscr{X}$ class ($m = 3$), denoted by solid circles, and one point in the $\mathscr{Y}$ class ($n = 1$), denoted by a solid triangle. Each point in the $\mathscr{X}$ class is a fixed distance apart from each other (solid lines) and a fixed distance from the $\mathscr{Y}$ point (dashed lines).

Due to this strong dependency, the pairwise distances have a non-degenerate discrete limiting distribution:

$$\sum_{k=1}^{d}\left(X_i^{(k)}-X_j^{(k)}\right)^2=\begin{cases} d & +O_p(1) \quad \text{w.p. } \frac{1}{4} \\ \sigma^2 d & +O_p(1) \quad \text{w.p. } \frac{1}{4} \\ \left(1+\sigma^2\right)d & +O_p(1) \quad \text{w.p. } \frac{1}{2} \end{cases}$$

for all $i \neq j$. The additional assumption A5 elegantly handles this case. Note that

$$\phi_{ij}=\text{Cov}\left(z_i^2-1,z_j^2-1\right)=\left(\frac{1+\sigma^2}{2}\right)^{-2}\text{Cov}\left(x_i^2,x_j^2\right)=\left(\frac{1+\sigma^2}{2}\right)^{-2}\left(\frac{1-\sigma^2}{2}\right)^2=\left(\frac{1-\sigma^2}{1+\sigma^2}\right)^2$$

for all $i \neq j$. Note that $\phi_{ii}=\text{Var}\left(z_i^2-1\right)\geq 0$. Thus, A5 does not hold because

$$
\begin{aligned}
\frac{\sum_{i,j}\lambda_i^X\lambda_j^X\phi_{ij}}{\left(\sum_{i=1}^d\lambda_i^X\right)^2} &= \frac{\left(\frac{1+\sigma^2}{2}\right)^2\sum_{i,j}\phi_{ij}}{d^2\left(\frac{1+\sigma^2}{2}\right)^2} \\
&= d^{-2}\left\{\sum_{i\neq j}\phi_{ij}+\sum_{i=1}^d\phi_{ii}\right\} \\
&= d^{-2}\left\{\left(d^2-d\right)\left(\frac{1-\sigma^2}{1+\sigma^2}\right)^2+\sum_{i=1}^d\phi_{ii}\right\} \\
&= \left(\frac{1-\sigma^2}{1+\sigma^2}\right)^2-d^{-1}\left(\frac{1-\sigma^2}{1+\sigma^2}\right)^2+d^{-2}\sum_{i=1}^d\phi_{ii} \\
&\geq \left(\frac{1-\sigma^2}{1+\sigma^2}\right) \text{ as } d\to\infty.
\end{aligned}
$$

Jung and Marron (2009) and Qiao *et al.* (2010) require a different assumption than A5 presented here from Yata and Aoshima (2012). Jung and Marron assume that the components of $\mathbf{Z}_d$ must satisfy a ρ-mixing condition (Kolmogorov and Rozanov, 1960). Qiao *et al.* instead require that the components of $\mathbf{Z}_d$ are independent. A5, proposed in Yata and Aoshima (2012), is a slightly more general dependency condition that holds when the assumptions of Jung and Marron and Qiao *et al.* are satisfied, as follows.

If the components of $\mathbf{Z}_d$ are independent, as required by Qiao *et al.*, then $\phi_{ij}=0$ for all $i\neq j$. Let $M=\sup_i(\phi_{ii})<\infty$. Then assumption A5 simplifies to

$$
\frac{\sum_{i,j}\lambda_i^X\lambda_j^X\phi_{ij}}{\left(\sum_{i=1}^d\lambda_i^X\right)^2}=\frac{\sum_{i\neq j}\lambda_i^X\lambda_j^X\phi_{ij}}{\left(\sum_{i=1}^d\lambda_i^X\right)^2}+\frac{\sum_{i=1}^d\left(\lambda_i^X\right)^2\phi_{ii}}{\left(\sum_{i=1}^d\lambda_i^X\right)^2}\leq M\frac{\sum_{i=1}^d\left(\lambda_i^X\right)^2}{\left(\sum_{i=1}^d\lambda_i^X\right)^2},
$$

which converges to zero as $d\to\infty$ under assumption A3. Thus, A5 holds if the components of $\mathbf{Z}_d$ are independent.

Instead, if the ρ-mixing condition of Jung and Marron holds, Yata and Aoshima claim that $|E\left\{\left(z_i^2-1\right)\left(z_j^2-1\right)\right\}|\to 0$ as $|i-j|\to\infty$. This implies that $\phi_{ij}\to 0$ in the same limit. There exists a variable $k\in\mathbb{N}$ such that $k\to\infty$ and $k\frac{\Sigma\left(\lambda_i^X\right)^2}{\left(\sum_{i=1}^d\lambda_i^X\right)^2}\to 0$ as $d\to\infty$ under A3. Write A5 as

$$
\frac{\sum_{i,j}\lambda_i^X\lambda_j^X\phi_{ij}}{\left(\sum_{i=1}^d\lambda_i^X\right)^2}=\frac{\sum_{|i-j|\leq k}\lambda_i^X\lambda_j^X\phi_{ij}}{\left(\sum_{i=1}^d\lambda_i^X\right)^2}+\frac{\sum_{|i-j|>k}\lambda_i^X\lambda_j^X\phi_{ij}}{\left(\sum_{i=1}^d\lambda_i^X\right)^2}. \tag{3.1}
$$

Let $M = \sup_{i,j} |\phi_{ij}| < \infty$. Then, for the first term of 3.1, it holds under A3 that

$$\frac{\sum_{|i-j|\leq k} \lambda_i^X \lambda_j^X \phi_{ij}}{\left(\sum_{i=1}^d \lambda_i^X\right)^2} \leq M\,(2k+1)\,\frac{\sum_{i=1}^d \left(\lambda_i^X\right)^2}{\left(\sum_{i=1}^d \lambda_i^X\right)^2},$$

which converges to zero as $d \to \infty$. For the second term of 3.1, it holds as $d \to \infty$ that

$$\frac{\sum_{|i-j|>k} \lambda_i^X \lambda_j^X \phi_{ij}}{\left(\sum_{i=1}^d \lambda_i^X\right)^2} \to 0$$

from the fact that $\phi_{ij} \to 0$ as $k \to \infty$ for $|i-j| > k$. Thus, the ρ-mixing condition of Jung and Marron implies A5.

We choose to use the assumption of Yata and Aoshima because it allows for a more transparent relationship between the additional assumptions needed for the geometric representation of data to hold in the HDMSS setting, as described next.

Yata and Aoshima (2012) show that the same geometrical representation of the data holds in the HDMSS setting if the following conditions also hold when $d \to \infty$ and $m \to \infty$:

$$m^2 \frac{\sum_{i=1}^d \left(\lambda_i^X\right)^2}{\left(\sum_{i=1}^d \lambda_i^X\right)^2} \to 0 \tag{B1}$$

$$m \frac{\sum_{i,j} \lambda_i^X \lambda_j^X \phi_{ij}}{\left(\sum_{i=1}^d \lambda_i^X\right)^2} \to 0 \tag{B2}$$

Notice that B1 and B2 are similar to the conditions A3 and A5, respectively, but multiplied by an additional factor of m. Similarly, we need the above conditions to also hold for the n-point sample $\mathscr{Y}(d)$. If B1 and B2 are satisfied, it follows that $\frac{d}{N} \to \infty$.

As a simple example, assume that all of the eigenvalues λ_i^X are equal and the components of $\mathbf{Z}_d$ are independent (i.e., $\phi_{ij} = 0$ for all $i \neq j$). Then, B1 and B2 require $d^{-1}m^2 \to 0$ as $d, m \to \infty$.

3.2 SWISS Score of HDLSS Data

Utilizing the geometric structure of HDLSS data presented in Section 3.1, we can derive an asymptotic representation of the SWISS score as $d \to \infty$. Theorem 1 describes the SWISS score in the HDLSS setting, where the sample size N is fixed. Because the result is a complicated function of multiple parameters, some corollaries give insight by studying important special cases. The final result of this section describes how Theorem 1 and its corollaries can be applied. Specifically, we show the range of attainable SWISS scores in the HDLSS setting for fixed $m = n$. Additionally, we describe and give insight into how to choose σ^2, τ^2, and μ^2 to obtain any possible SWISS score.

Theorem 1. *Assume that conditions A1-A5 described in Section 3.1 hold. Then, as $d \to \infty$, the SWISS score converges in probability to*

$$\frac{(m-1)\,\sigma^2 + (n-1)\,\tau^2}{\left(\frac{mn}{m+n}\right)\left\{\mu^2 + \sigma^2/m + \tau^2/n\right\} + (m-1)\,\sigma^2 + (n-1)\,\tau^2}.$$

Proof. SWISS is defined in Subsection 2.2.1 as the Total WithIn class Sum of Squares (TWISS) normalized by the Total Sum of Squares (TSS).

First, we will calculate TWISS. The squared distance from any point in $\mathscr{X}(d)$ to its centroid C_X is asymptotically $\sigma^2 d\left(1 - m^{-1}\right)$ (Hall *et al.*, 2005). Similarly, the squared distance from any point in $\mathscr{Y}(d)$ to its centroid C_Y is asymptotically $\tau^2 d\left(1 - n^{-1}\right)$. Therefore, TWISS converges to

$$m\left\{\sigma^2 d\left(1 - m^{-1}\right)\right\} + n\left\{\tau^2 d\left(1 - n^{-1}\right)\right\} = d\left\{(m-1)\,\sigma^2 + (n-1)\,\tau^2\right\}.$$

The first step in calculating TSS is to determine the squared distances between the class centroids and the overall mean. Let $Y \in \mathscr{Y}(d)$. It was shown in Section 3.1 that Y is approximately a distance of $cd^{1/2}$ from each point in $\mathscr{X}(d)$. Then Y, any point $X \in \mathscr{X}(d)$ and C_X are the vertices of a right triangle where the line joining Y to X is the hypotenuse. This geometry is visualized in plot A of Figure 3.2. Therefore, by the Pythagorean theorem, the squared distance from a point in $\mathscr{Y}(d)$ to C_X

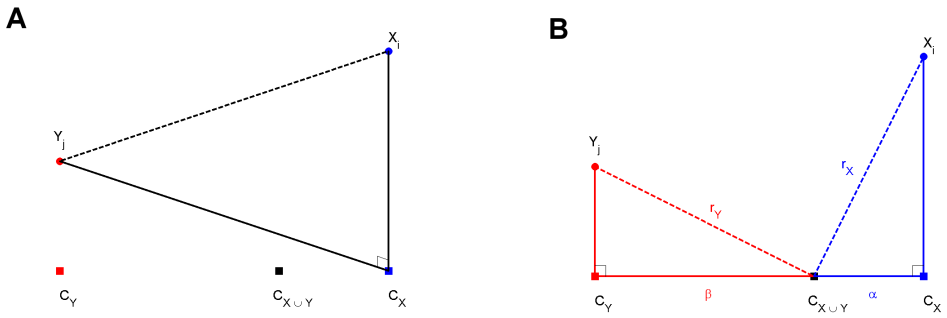

Figure 3.2: Geometrical structure of HDLSS data. (A) X_i, Y_j and C_X are the vertices of a right triangle, where the hypotenuse is designated by a dashed line. (B) X_i, C_X and $C_{X\cup Y}$ are the vertices of one right triangle and Y_j, C_Y and $C_{X\cup Y}$ are the vertices of another right triangle, where the hypotenuses are designated by dashed lines.

is asymptotically

$$\begin{aligned} c^2 d - \sigma^2 d\left(1-m^{-1}\right) &= d\left(\mu^2+\sigma^2+\tau^2\right) - \sigma^2 d\left(1-m^{-1}\right) \\ &= d\left\{\mu^2+\sigma^2/m+\tau^2\right\}. \end{aligned}$$

The same analysis yields that the squared distance from C_X to C_Y converges to

$$d\left\{\mu^2+\sigma^2/m+\tau^2\right\} - \tau^2 d\left(1-n^{-1}\right) = d\left\{\mu^2+\sigma^2/m+\tau^2/n\right\}.$$

Denote the distance between C_X and the overall mean, $C_{X\cup Y}$, as α and the distance between C_Y and $C_{X\cup Y}$ as β (as shown in plot B of Figure 3.2). We have just shown that

$$(\alpha+\beta)^2 = d\left\{\mu^2+\sigma^2/m+\tau^2/n\right\}. \tag{3.2}$$

It is also known that $C_{X\cup Y}$ is simply a weighted average of C_X and C_Y, giving

$$\frac{\beta}{\alpha} = \frac{m}{n}. \tag{3.3}$$

Solving the simultaneous equations (3.2) and (3.3) for α and β, we obtain

$$\begin{aligned}\alpha &= \left(\frac{n}{m+n}\right) d^{1/2}\left\{\mu^2+\sigma^2/m+\tau^2/n\right\} \\ \beta &= \left(\frac{m}{m+n}\right) d^{1/2}\left\{\mu^2+\sigma^2/m+\tau^2/n\right\}.\end{aligned}$$

Now that we have the distances between the class centroids and $C_{X\cup Y}$, the next step in calculating TSS is determining the squared distance, r_X^2, between a point $X \in \mathscr{X}(d)$ and $C_{X\cup Y}$. Once again, using the Pythagorean theorem, we obtain

$$\begin{aligned}r_X^2 &= \alpha^2 + d\left(1-m^{-1}\right)\sigma^2 \\ &= \left(\frac{n}{m+n}\right)^2 d\left\{\mu^2+\sigma^2/m+\tau^2/n\right\} + d\left(1-m^{-1}\right)\sigma^2.\end{aligned}$$

Similarly, the squared distance, r_Y^2, between a point in $\mathscr{Y}(d)$ and $C_{X\cup Y}$ is

$$\begin{aligned}r_Y^2 &= \beta^2 + d\left(1-n^{-1}\right)\tau^2 \\ &= \left(\frac{m}{m+n}\right)^2 d\left\{\mu^2+\sigma^2/m+\tau^2/n\right\} + d\left(1-n^{-1}\right)\tau^2.\end{aligned}$$

We now obtain TSS by summing the squared distances between each point and $C_{X\cup Y}$.

$$\begin{aligned}\text{SST} &= mr_X^2 + nr_Y^2 \\ &= d\left[\left(\frac{mn}{m+n}\right)\left\{\mu^2+\sigma^2/m+\tau^2/n\right\} + (m-1)\sigma^2 + (n-1)\tau^2\right].\end{aligned}$$

Finally, by Slutsky's Theorem (p. 60 in Shao, 2003), SWISS is calculated as

$$\begin{aligned}\text{SWISS} &= \frac{\text{TWISS}}{\text{TSS}} \\ &= \frac{d\left\{(m-1)\sigma^2+(n-1)\tau^2\right\}}{d\left[\left(\frac{mn}{m+n}\right)\left\{\mu^2+\sigma^2/m+\tau^2/n\right\} + (m-1)\sigma^2+(n-1)\tau^2\right]} \\ &= \frac{(m-1)\sigma^2+(n-1)\tau^2}{\left(\frac{mn}{m+n}\right)\left\{\mu^2+\sigma^2/m+\tau^2/n\right\} + (m-1)\sigma^2+(n-1)\tau^2}.\end{aligned}$$

□

The result of Theorem 1 is difficult to interpret because it is a function of five different parameters. The following corollaries describe the behavior of the SWISS score for important special situations that may arise under the HDLSS setting.

Corollary 1. *Assume that σ, τ, m and n are finite. Then SWISS$\to 0$ as $d \to \infty$ and $\mu \to \infty$.*

Corollary 1 says that, in the HDLSS setting, as the distance between the class means (μ) increases to infinity, all of the variation in the data is due to variation between the class means, resulting in a SWISS score of zero.

It may appear surprising that there is no restriction on the minimum growth rate of μ. For example, write $\mu = d^{\alpha}$ for $\alpha > 0$. We saw in the proof of Theorem 1 that the distance from any point in $\mathscr{X}(d)$ to its centroid C_X is asymptotically $\sigma d^{1/2}\left(1-m^{-1}\right)^{1/2}$ and from any point in $\mathscr{Y}(d)$ to its centroid C_Y is asymptotically $\tau d^{1/2}\left(1-n^{-1}\right)^{1/2}$. It seems reasonable that the restriction $\alpha > \frac{1}{2}$ is needed to ensure that the variation between the class means outgrows the variation within the classes, resulting in a SWISS score of zero. However, this restriction is not needed because μ is already multiplied by $d^{1/2}$ when it is considered in our calculations (appearing as the distance from a point in $\mathscr{X}(d)$ to a point in $\mathscr{Y}(d)$, which equals $cd^{1/2} = d^{1/2}\left(\mu^2+\sigma^2+\tau^2\right)^{1/2}$). This allows us to factor out $d^{1/2}$ in our asymptotic representation of the SWISS score. Therefore, Corollary 1 holds for any $\mu \to \infty$.

Corollary 1 shows that allowing μ to grow to infinity results in a SWISS score of zero in the limit. It seems natural that to obtain a limiting SWISS score of one (the opposite extreme), we should set $\mu = 0$. However, as seen in Corollary 2, this is not the case.

Corollary 2. *Assume that σ, τ, m and n are finite and $\mu = 0$. Then, as $d \to \infty$,*

$$SWISS \quad \to \quad 1 - \frac{n\sigma^2 + m\tau^2}{(m+n-1)\left(m\sigma^2 + n\tau^2\right)}.$$

(a) If $\sigma = \tau$ and $\mu = 0$, then SWISS $\to 1 - \frac{1}{m+n-1}$ as $d \to \infty$.

(b) If $m = n$ and $\mu = 0$, then SWISS $\to 1 - \frac{1}{2m-1}$ as $d \to \infty$.

Corollary 2 says that if the population means of the two classes are equal, then the SWISS score is near one for large $N = m+n$. The fact that SWISS is not exactly equal to one when there is no difference between the class means may at first appear counter-intuitive. This disconnect is simply due

to sampling variability: the distance between the class centroids is not zero even though the distance between the population means (μ) is zero. We will show that under the HDMSS setting (described in the next section) that if we also allow N to grow to infinity after letting $d \to \infty$, resulting in the sample class centroids converging to the population means, then the SWISS score converges to exactly one.

Next, we investigate the general behavior of the SWISS score when the two classes have equal sample sizes $(m = n)$.

Corollary 3. *Assume that μ, σ, τ, and $m = n$ are finite. Then, as $d \to \infty$,*

$$SWISS \to \frac{2(m-1)\left(\sigma^2+\tau^2\right)}{m\mu^2+(2m-1)\left(\sigma^2+\tau^2\right)}.$$

Corollaries 1 and 3 show that in the HDLSS setting when $m = n$, we can obtain a SWISS score of zero by letting $\mu \to \infty$. At the other extreme, when $m = n$, Corollaries 2 and 3 show that the largest possible SWISS score is $1 - \frac{1}{2m-1}$, which occurs when $\mu = 0$. In fact, for fixed σ^2 and τ^2, we can obtain any SWISS score between zero and $1 - \frac{1}{2m-1}$ by simply changing μ^2. The details are given in the following corollary.

Corollary 4. *If $m = n$ and σ^2, τ^2 and $\mu^2$are chosen such that*

$$\frac{\mu^2}{(\sigma^2+\tau^2)} = \frac{2}{S}\left(1-\frac{1}{m}\right)+\frac{1}{m}-2$$

where $0 < S < 1 - \frac{1}{2m-1}$, then SWISS $\to S$ as $d \to \infty$.

The ratio $\mu^2/\left(\sigma^2+\tau^2\right)$ can be thought of as a signal-to-noise ratio. Corollary 4 says that to achieve a smaller SWISS score and, thus, a better clustering of the data, the signal-to-noise ratio needs to increase. The relationship between the signal-to-noise ratio and the SWISS score in the HDLSS setting is visualized in Figure 3.3. Each line represents a different pair of sample sizes (shown in the legend).

Figure 3.3 also shows that for a fixed signal-to-noise ratio, a smaller SWISS score is achieved by a smaller sample size. It was shown in the proof of Theorem 1 that the squared distance between class centroids, scaled by a factor of d, equals $\mu^2 + \sigma^2/m + \tau^2/n$. Therefore, the distance between class centroids is larger for smaller sample sizes. Additionally, the squared distance between a point and its

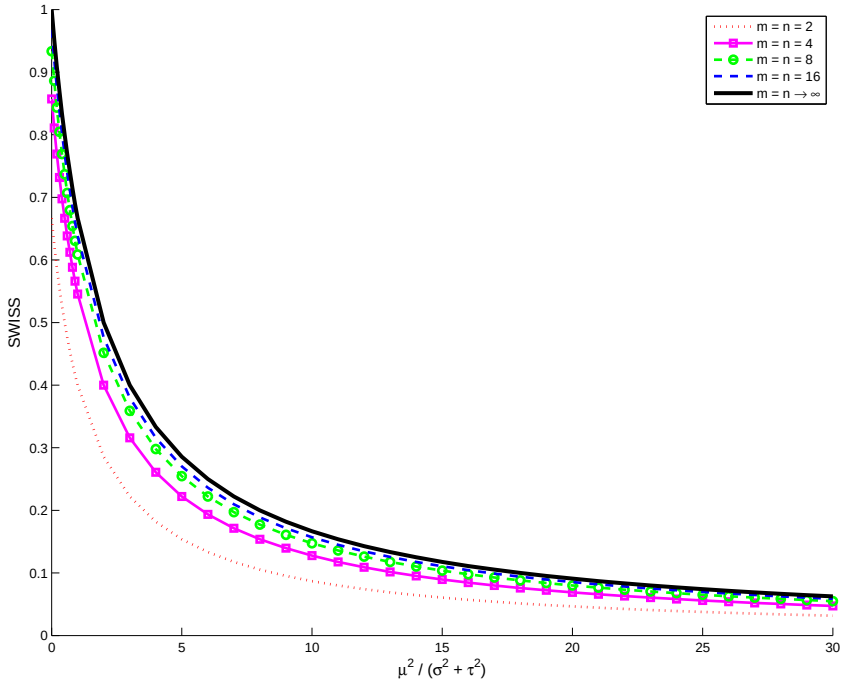

Figure 3.3: The relationship between the signal-to-noise ratio $\mu^2/\left(\sigma^2+\tau^2\right)$ and the SWISS score in the HDLSS setting for a variety of sample sizes. The solid line shows the relationship in the HDMSS setting ($m = n \to \infty$). As the signal-to-noise ratio increases, the SWISS score decreases. Additionally, for a fixed signal-to-noise ratio, a smaller sample size achieves a smaller SWISS score than a larger sample size.

class centroid, scaled by a factor of d, is either $\sigma^2\left(1-m^{-1}\right)$ or $\tau^2\left(1-n^{-1}\right)$, which decreases as the sample size decreases. Therefore, the SWISS score is smaller for smaller sample sizes because the distance between centroids increases (larger TSS) and the distance between each point and its class centroid decreases (smaller TWISS).

The solid line in Figure 3.3 (at the top edge of the bundle of curves) shows the limiting case of this relationship, when m and n both grow to infinity (although at a slower rate than d, such that assumptions B1 and B2 from Section 3.1 hold). This High Dimension Moderate Sample Size setting will be discussed in more detail in the next section.

3.3 SWISS Score of HDMSS Data

In this section, we will describe the behavior of the SWISS score in the High Dimension Moderate Sample Size (HDMSS) setting, where d and N both grow such that $\frac{d}{N} \to \infty$. HDMSS data fall squarely between two popular situations: HDLSS (N fixed with $d \to \infty$) and random matrices ($\frac{d(N)}{N} \to c$ as $N \to \infty$). We require that assumptions A1-A5 and B1-B2 from Section 3.1 hold to guarantee that the same geometric representation of HDLSS data described in that section still holds. The following corollaries follow directly from taking limits of the results presented in Section 3.2. Since these results depend on fewer parameters than the results presented in the previous section, we are able to more easily interpret these results.

Corollary 5. *Assume that* $\mu, \sigma, \tau < \infty$ *and either* $m \to \infty$ *or* $n \to \infty$. *Then SWISS* $\to 1$ *as* $d \to \infty$.

Corollary 5 says that if the class variances and distance between classes are finite, then the SWISS score converges to one as the sample size of one population grows. Notice that Corollary 5 needs only one of m or n to tend to infinity, which is more general than requiring both sample sizes to grow. Additionally, this theorem includes the case when $\mu = 0$. In the HDLSS setting, Corollary 2 tells us that the SWISS score does not converge to one when $\mu = 0$ due to sampling variation. However, Corollary 5 says that the SWISS score equals one when $\mu = 0$ (or any finite value) in the HDMSS setting.

Corollary 5 describes the asymptotic behavior of the SWISS score when only one of the class sample sizes grows. The final corollaries describe the behavior of the SWISS score when both m and n grow at the same rate.

Corollary 6. *Assume that* $\mu, \sigma, \tau < \infty$ *and* $m, n \to \infty$ *such that* $\frac{m}{n} \to 1$. *Then*

$$SWISS \to 1 - \frac{\mu^2}{\mu^2 + 2\left(\sigma^2 + \tau^2\right)} \qquad as\ d \to \infty.$$

In the HDLSS setting, Corollary 4 shows how to change the signal-to-noise ratio $\mu^2 / \left(\sigma^2 + \tau^2\right)$ to obtain any SWISS score between zero and $1 - \frac{1}{2m-1}$. The following corollary uses the result of

Corollary 6 to show how any SWISS score between zero and one can be achieved in the HDMSS setting by changing the signal-to-noise ratio.

Corollary 7. *If $m, n \to \infty$ such that $\frac{m}{n} \to 1$ and σ^2, τ^2 and μ^2 are chosen such that*

$$\frac{\mu^2}{(\sigma^2+\tau^2)} = 2\left(\frac{1}{S}-1\right)$$

where $0 < S < 1$, then SWISS $\to S$ as $d \to \infty$.

The relationship between the signal-to-noise ratio and the SWISS score in the HDMSS setting (along with special cases in the HDLSS setting) is visualized in Figure 3.3 on page 43. This figure shows that, for a fixed signal-to-noise ratio, the SWISS score achieved in the HDMSS setting is larger than any SWISS score achieved in the HDLSS setting. Similar to the explanation given at the end of Section 3.2, this is because the distance between class centroids decreases and the distance between each point and its class centroid increases as the sample sizes of both classes increase, resulting in a larger SWISS score.

3.4 Future Research Directions

In this chapter, we have presented the asymptotic behavior of the SWISS score under both the HDLSS and HDMSS settings. An interesting future research direction is to study the behavior of SWISS in the random matrices setting. In that setting, $\frac{d(N)}{N} \to c$ as $N \to \infty$. It may be much more difficult to calculate an asymptotic representation of the SWISS score in this situation as it is not known if data under this setting have a fixed geometric representation.

Chapter 4

Recalibrating Quality Scores from Sequencing Data

Second-generation sequencing, also known as next-generation sequencing, is rapidly becoming the technology of choice for genomic studies, including cancer genetics (Meyerson *et al*., 2010). This is because a single experiment can produce data for gene expression, copy number and variant detection studies (Lamlertthon *et al*., 2011). These multiple applications paired with the decreasing cost of sequencing has lead to a current rush for novel biological findings. Unfortunately, the sequencing technology is not error-free. A sequencer will occasionally call an incorrect base, or nucleotide. Depending on the specific machine used, these sequencing errors occur at a rate of approximately one error per every 1,000 bases sequenced (Zhang *et al.*, 2011). When each base is called, a quality score is concurrently given that corresponds to the probability that the base is a sequencing error. Incorporating these quality scores into down-stream analyses, such as mutation calling, can help produce more accurate and confident results. However, it has been shown that these quality scores do not accurately reflect the probability of a sequencing error (Li *et al.*, 2009b; Bravo and Irizarry, 2010). This chapter describes a novel method, ReQON, which recalibrates the quality scores to produce more accurate quality scores.

The rest of the chapter is laid out as follows. Section 4.1 gives a brief overview of second-generation sequencing and specific terminology that will be used throughout the chapter. Section 4.2 describes ReQON, a computational tool which recalibrates base quality scores. Section 4.3 evaluates the performance of ReQON and Section 4.4 compares ReQON with other quality score recalibration methods.

4.1 Second-generation Sequencing

This section will give a brief overview of the technology behind second-generation sequencing. Much of the information presented in this section comes from the thorough and insightful reviews by Meyerson *et al.* (2010) and Zhang *et al.* (2011).

The first-generation sequencing methodology, known as Sanger chemistry, uses specifically labeled nucleotides to read through a DNA template. This sequencing technology starts the read at a specific position along the DNA template and records the different labels for each nucleotide within the sequence. Currently, the Sanger method has the capacity to read through 1000-1200 basepair (Zhang *et al.*, 2011).

In order to sequence longer sections of DNA, a new approach called shotgun sequencing was developed during the Human Genome Project. In this approach, DNA is broken down into smaller fragments and cloned into sequencing vectors in which cloned DNA fragments can be sequenced individually. The complete sequence of a long DNA fragment can be eventually generated by these methods by aligning and reassembling sequence fragments based on partial sequence overlaps. Shotgun sequencing made sequencing the entire human genome possible (Venter *et al.*, 2003).

The core philosophy of massively parallel sequencing used in *second-generation sequencing*, or *next-generation sequencing* (*NGS*), is adapted from shotgun sequencing. NGS technologies read the DNA templates randomly along the entire genome. A *read* is a small fragment of DNA that is sequenced. We will refer to a single position of a read as a *base*, and the specific molecule at that position as a *nucleotide*. Each base is one of four possible nucleotides, represented by characters A, C, G and T. The number of bases that can be sequenced for a given cost has increased one million-fold since 1990, more than doubling every year (Meyerson *et al.*, 2010). Current technology allows for over 100 million reads to be sequenced in a single experiment.

For every base sequenced, the sequencer machine also reports a *quality score*, which corresponds to the probability that an incorrect nucleotide was called. These quality scores are often represented on the Phred scale (Ewing and Green, 1998). The Phred scale defines a quality score Q as a log-transformed error probability P:

$$Q = -10\log_{10}(P).$$

For example, a quality score of 30 corresponds to an error probability of $1/1000$.

The *read length* is the actual number of continuous bases sequenced. The read lengths for NGS is much shorter than that attained by Sanger sequencing. At present, NGS only provides 50-500 continuous basepair reads, which is why sequencing results are defined as *short reads* (Zhang *et al.*, 2011). These short reads are a major limitation and, coupled with large data volume, present difficulties for data analysis. These challenges have motivated the development of new computational tools for every NGS data analysis task from variant detection and sequence mapping to downstream biological and functional analyses (Ding *et al.*, 2010).

NGS can be further divided based on the type of input material. Complete sequencing of the genome using DNA is known as *whole-genome sequencing*. Whole-genome sequencing provides the most comprehensive characterization of the genome, but it requires the greatest amount of sequencing. Sequencing of RNA is known as *RNA-Seq,* or transcriptome sequencing. RNA-Seq allows analysis of gene expression profiles and is particularly powerful for identifying genes with low expression. Additionally, RNA-Seq provides the advantage of not being limited to known genes but can also include the detection of novel transcripts, alternative splice forms and non-human transcripts (Meyerson *et al.*, 2010)

Before the data can be analyzed, reads must be aligned to the specific chromosome, position and DNA strand from which they are most likely to have originated. It is a challenge to efficiently align short reads to a reference genome (Li and Homer, 2010). There are many alignment algorithms available. The most popular whole-genome sequencing aligners include Bowtie (Langmead *et al.*, 2009), MAQ (Li and Durbin, 2009) and BWA (Li and Durbin, 2010). The most popular RNA-Seq aligners include TopHat (Trapnel *et al*., 2009), MapSplice (Wang *et al*., 2010) and GSNAP (Wu and Nacu, 2010).

Because NGS currently produces short reads, coverage is a very important issue (Zhang *et al.*, 2011). The *coverage* at a genomic position is defined as the number of bases that map to that position. For example, if 30 short reads align to a position, even if all nucleotides are not the same, then we say this position has 30x coverage. The *allele frequency* is the fraction of bases at a position of a given nucleotide. Suppose that, of 30 bases covering a position, 28 bases are A's and the other 2 are T's. The A allele frequency is $\frac{28}{30} = 0.933$, the T allele frequency is $\frac{2}{30} = 0.067$ and the C and G allele

Platform	Accuracy
Roche GS-FLX 454 Genome Sequencer	$> 99\%$
Helicos HeliScope	$> 99\%$
Illumina Genome Analyzer	$> 99.5\%$
ABI SOLiD Platform	99.94%

Table 4.1: Raw base calling accuracies of four different second-generation sequencing platforms (Zhang *et al.*, 2011).

frequencies are both 0. Often, it is more useful to describe the allele frequencies at a position without referring to a specific nucleotide. If we know what the reference nucleotide is for a position, we will refer to the allele frequency of the nucleotide matching the reference as the *reference allele frequency*, and the sum of the other non-reference nucleotide frequencies as the *non-reference allele frequency*. Referring to the earlier example, if the reference nucleotide is A, then the reference allele frequency is 0.933 and the non-reference allele frequency is 0.067.

NGS holds enormous promise for the study of mutations, copy number alterations and structural variants, including fusion genes (Ding *et al.*, 2010). The discovery of mutations that determine phenotypes is a fundamental premise of genetic research (Mardis, 2008). Many NGS analyses can be complicated by the presence of sequencing errors, especially variant detection. Systematic errors in NGS are not yet well defined, making it difficult to distinguish true genetic variants from sequencing errors (Shen *et al.*, 2010). Table 4.1 reports the base calling accuracies for four different NGS platforms. For each platform, over 99% of the bases will be called correctly. However, even with a sequencing error rate of 1/1000, this results in 1 million sequencing errors per 1 billion bases sequenced (very typical with current throughput). Because it is not known exactly which bases are sequencing errors and it is impractical to throw out such a large amount of data from the analysis, computational tools are beginning to leverage the information contained in base quality scores. These tools down-weight bases with low quality scores (i.e., high probability of being a sequencing error) and give stronger consideration to bases with high quality scores. Examples of popular variant calling tools that incorporate base quality scores are VarScan (Koboldt *et al.*, 2009), SNVMix (Goya *et al.*, 2010) and Atlas-SNP2 (Shen *et al.*, 2010).

Unfortunately, the reported base quality scores are inaccurate (Li *et al.*, 2009b; Bravo and Irizarry, 2010; our Figure 4.1 on page 54). If the aforementioned variant calling tools are to give the best

results, the quality scores must be accurate. The quality scores from current platforms are often not accurate enough to differentiate true variants from sequencing errors (Shen *et al.*, 2010). Therefore, there is a need for a computational tool that recalibrates these quality scores so that they:

1. more accurately represent the probability of a sequencing error and
2. do a better job of discriminating between sequencing errors and non-errors, making it easier to detect true variants.

We developed an algorithm, called ReQON (Recalibrating Quality Of Nucleotides), that recalibrates the quality scores and produces diagnostic plots, which show the improvement in accuracy of the recalibrated quality scores. ReQON is described in detail in the next section.

4.2 Method

This section describes the ReQON algorithm developed to recalibrate base quality scores.

4.2.1 Input

The input to ReQON is an indexed and sorted BAM file, a commonly used binary text file that contains aligned sequence data (Li *et al.*, 2009a). The reads can be aligned to any genome using any alignment algorithm. The recalibration results will be dependent on the accuracy of these alignments.

4.2.2 Algorithm

ReQON uses logistic regression to recalibrate the quality scores. The following parameters must be specified:

- *Region*: Genomic region that the regression model is trained on. The training region must be large enough to obtain accurate coefficients for the regression model. On the other hand, specifying a larger region than necessary will increase run time and may overfit the model to the training set. We recommend training on one of the smaller chromosomes or specifying *MaxTrain*, described next.

- *MaxTrain* (optional): This option allows the user to train on a fixed number of bases from a large training region. For example, training on the first 10 million bases of chromosome 1 is achieved by setting Region = "chr1" and MaxTrain = $10,000,000$. Typically, results do not significantly improve when the training size is larger than 25 million bases.

- *RefSeq* (optional): File containing the reference sequence corresponding to the training set. This allows ReQON to easily identify sequencing errors (i.e., base is an error if it does not match *RefSeq*). Default: after removing all bases at positions with coverage ≤ 2, bases not matching the most common nucleotide at a position are identified as sequencing errors.

- *SNP* (optional): File of known variant locations to remove from the training set before recalibration. Due to natural genetic variation in the population, these positions are known to have more than one possible nucleotide. This makes it difficult to determine which of the bases not matching *RefSeq* are sequencing errors, and which are attributable to true biological variation.

- *nerr*: The maximum number of errors tolerated at a genomic position (default = 2). Positions with more than *nerr* errors may likely be true variants, so bases from these positions are removed from the training region.

- *nraf*: The maximum non-reference allele frequency at a genomic position that is allowed (default = 0.05). Positions with non-reference allele frequency greater than $nraf$ are removed from the training set for the same reason as above.

The first step is to read the training region from the input BAM file. Positions that are listed in *SNP* are removed from the training set. Next, sequencing errors are identified as bases that do not match *RefSeq*. In reality, some of these identified bases will not be errors but instead correct calls, such as novel variants. In an attempt to remedy this issue, we set two different thresholds to remove positions most likely to contain incorrect error calls. At each position with at least one called error, we calculate $thresh = \max\{nerr, nraf \times \text{coverage}\}$. The threshold sets a maximum number of allowable called errors for positions with low coverage, and a maximum frequency of non-reference bases for positions with high coverage. The default settings allow up to 2 errors per position if the coverage is less than 60x, and more errors ($0.05 \times$ coverage) for positions with at least 60x coverage. We then

identify positions with more called errors than *thresh,* and remove these positions from the training set. Note that when we say that "a position is removed from the training set," this is different than keeping the bases at this position in the training set but switching the call from error to non-error. Instead, we remove all bases at this position from the training set because we do not have high confidence in our call of error/non-error.

We now have a list of all bases in the training set, each classified as either error or non-error. There is a strong relationship between errors and their position in the read. Most errors occur at the end of the read, due to technical aspects of the sequencing process (Dohm *et al.*, 2008). This same pattern is present in plot A of Figure 4.1, which shows the distribution of errors in the training set by their read position. The data used in this figure will be described in Section 4.3. To account for these high-error read positions in our regression model, the algorithm flags read positions that have more errors than a specified threshold, set at 1.5 times the average number of errors per read position. This threshold is visualized as the dashed cyan line in plot A of Figure 4.1. Each flagged read position will be included as an indicator variable in the regression model.

Logistic regression is performed on the training set, with the following covariates:

- Original quality score.
- Indicator variable that original quality score equals zero. Many base calling algorithms assign a quality of zero to indicate bad or randomly called bases.
- Average quality score across all bases in a read. This helps identify reads where all bases are assigned low quality scores.
- Read position. This fits a linear trend across all read positions.
- Indicator variables for any flagged read positions, described in the previous paragraph.
- Nucleotide called. Indicator variables for A, C and G (T is incorporated into the intercept).

Due to memory constraints, it is not practical to fit the model when the training set contains tens of millions of bases. Instead, we split the training set into smaller subsets of no more than 10 million bases. We run the regression model on all subsets, then, for each coefficient, take the median over all

regression models. Once these median regression coefficients are obtained, the model is applied to each base in the input BAM file to obtain predicted error probability values. These probabilities are transformed to the Phred scale (Ewing and Green, 1998) and rounded down to the nearest integer.

4.2.3 Output and Visualizations

ReQON outputs a BAM file with original quality scores replaced by the recalibrated scores. Additionally, ReQON produces diagnostic plots (Figure 4.1) which show the effectiveness of the quality score recalibration on the training set. Plot A shows the distribution of errors in the training set by their read position. Any read position above the threshold (dashed cyan line) is given an additional indicator variable in the regression model, discussed in Subsection 4.2.2. Plot B shows the relative frequency distribution of quality scores both before (solid blue line) and after (dashed red line) recalibration.

The bottom two plots show the reported quality score versus the empirical quality score before (plot C) and after (plot D) recalibration. The *empirical quality score* is calculated by computing the observed error rate for all bases in the training set that are assigned a specific quality score. This error rate is then converted to the log-transformed Phred scale (Ewing and Green, 1998). If the quality scores are accurate, which occurs when the observed and reported sequencing error rates match, the points will fall on the 45-degree line.

The bottom plots also report the *Frequency-Weighted Squared Error (FWSE)*, a measure of how close the points lie to the 45-degree line. FWSE is calculated by squaring the error (vertical distance between the point and 45-degree line), weighting this squared error by its relative frequency (shown in plot B) and summing across all quality scores. To help visualize which quality scores have small relative frequencies, and thus do not have a large contribution to FWSE, any point with relative frequency less than 0.1% is denoted by a solid dot. FWSE will be close to zero if the quality scores accurately reflect the probability of a sequencing error.

4.2.4 Availability

ReQON is available as an R package through the Bioconductor project (http://www.bioconductor.org/).

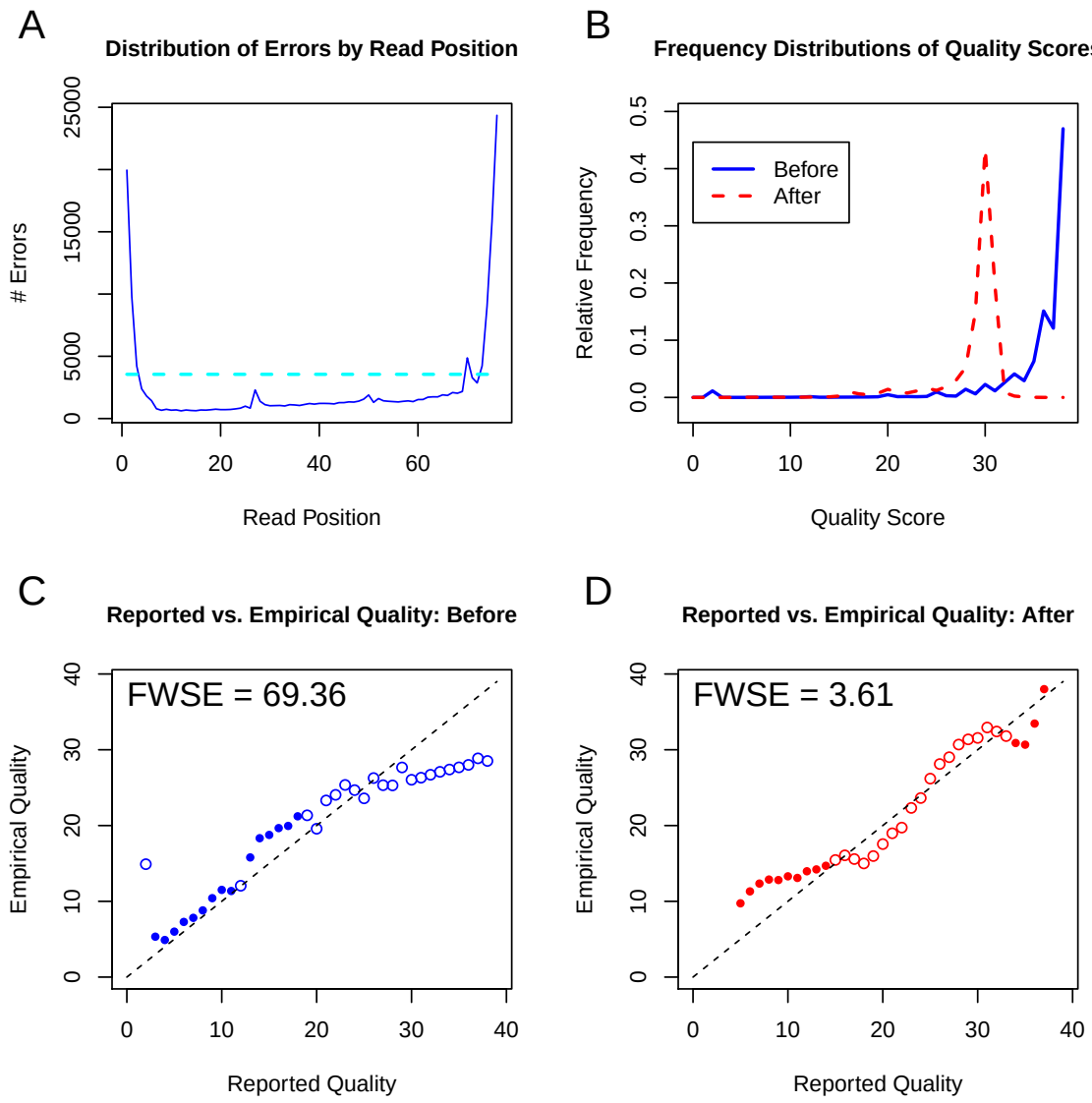

Figure 4.1: Recalibration of U87 cell line replicate 1 with ReQON. Plot A shows the distribution of errors by read position. Plot B shows frequency distributions of quality scores before (solid blue) and after (dashed red) recalibration. Reported quality scores versus empirical quality scores are shown before recalibration (plot C) and after ReQON (plot D). Plots C and D also report FWSE, a measure of quality score accuracy.

4.3 Results

As an example, multiple replicates of RNA from the U87 glioblastoma cell line (Clark *et al.*, 2010) were sequenced using Illumina's Genome Analyzer II, representing identical sequence runs but of differing quality. Each run produced 76 basepair single-end reads which were aligned to the human genome reference version 19 (hg19) using MapSplice (Wang *et al.*, 2010). The aligned reads were sorted and indexed using SAMtools (Li *et al.*, 2009a).

Figure 4.1 shows the diagnostic output of ReQON after recalibrating cell line replicate 1. The model was trained on chromosome 10 after removing positions from dbSNP version 132 (Sherry *et al.*, 1999). Plot A shows that a majority of the sequencing errors occur at the ends of the read. Plot B shows that the majority of quality scores before recalibration were larger than 35, with almost 50% of the bases receiving a quality score near 40. After recalibration, the quality scores are assigned smaller values, with most scores falling between 25 and 35. Plots C confirms that the original quality scores were not very accurate because the quality scores with the largest frequencies (> 35) are far from the 45-degree line and, thus, FWSE is large. For example, we see in plot B that approximately 50% of the bases are assigned a quality score of 40. In plot C, we see that the empirical quality of this reported score is around 30. Thus, this single quality score contributes approximately $0.5\,(40-30)^2 = 50$ to the total FWSE of 69.36. Plot D shows that, after recalibration, the quality scores do a much better job of representing true sequencing error rates, reflected by the 95% decrease in FWSE. Some of the lower quality scores lie away from the 45-degree line, but because few bases are assigned these scores (represented by the solid dots), they contribute little to FWSE.

Table 4.2 shows the before and after recalibration FWSE values of the training set (chromosome 10) for the three cell line replicates. ReQON decreases FWSE by over 90% for all three replicates. Table 4.2 also reports FWSE before and after applying ReQON to an independent testing set (chromosome 20). In each case, FWSE of the recalibrated quality scores is approximately the same for both the training and testing sets. This demonstrates that ReQON does not overfit the model to the training set.

The previous FWSE analysis shows that the recalibrated quality scores accurately represent the probability of a sequencing error. Another desirable property of quality scores is the ability to separate

	Training set (chr 10)		Testing set (chr 20)	
	Before	After	Before	After
Replicate 1	69.36	3.61	71.09	3.04
Replicate 2	69.17	4.61	74.06	4.82
Replicate 3	62.89	5.31	64.34	5.82

Table 4.2: Comparison of Frequency-Weighted Squared Error (FWSE) for three cell line replicates before and after recalibrating quality scores with ReQON. FWSE is calculated for the training set (chr 10) and an independent testing set (chr 20). ReQON does not overfit the model to the training set, shown by the roughly equivalent FWSE values for both the training and testing sets after recalibration.

sequencing errors from true variants. To perform this analysis, we look at all bases that do not match *RefSeq*. We further divide these non-reference bases into true variants (reported in dbSNP version 132) and sequencing errors. Similar to our training set, we remove bases identified as sequencing errors if there are more than two errors at a position (or allele frequency greater than 0.05 for high coverage positions) as these may represent novel variants or systematic alignment errors. To increase confidence in our true variant calls, we also remove positions identified as true variants with *less* than 3 non-reference bases (or allele frequency less than 0.05 for high coverage positions) as these may actually be sequencing errors.

We measure classification performance by the area under the corresponding receiver operating characteristic (ROC) curve, or AUC. Hanley and McNeil (1982) show that the AUC statistic is equivalent to the probability that a randomly chosen point (base) is correctly classified. An AUC of 1 represents perfect classification and an AUC of 0.5 suggests classification by random chance.

Figure 4.2 plots the relative frequency distributions of quality scores for non-reference bases in chromosome 20 of cell line replicate 3 (after training on chromosome 10). The red curve plots the distribution of bases belonging to positions in dbSNP (13,491 bases) and the blue curve plots the distribution of bases identified as sequencing errors (83,180 bases). Plot A shows the distribution of quality scores before recalibration. Note that the blue curve has a larger peak around 2. The AUC for the original quality scores is 0.764. Plot B shows the distribution of quality scores after recalibration with ReQON. The two curves now have very different distributions. The quality scores of most bases at positions in dbSNP (red) have high quality scores (above 25). It is possible that many of the lower quality nucleotides at positions in dbSNP are sequencing errors. In contrast, the quality scores of sequencing errors (blue) are mostly below 25. This better separation between the two classes is

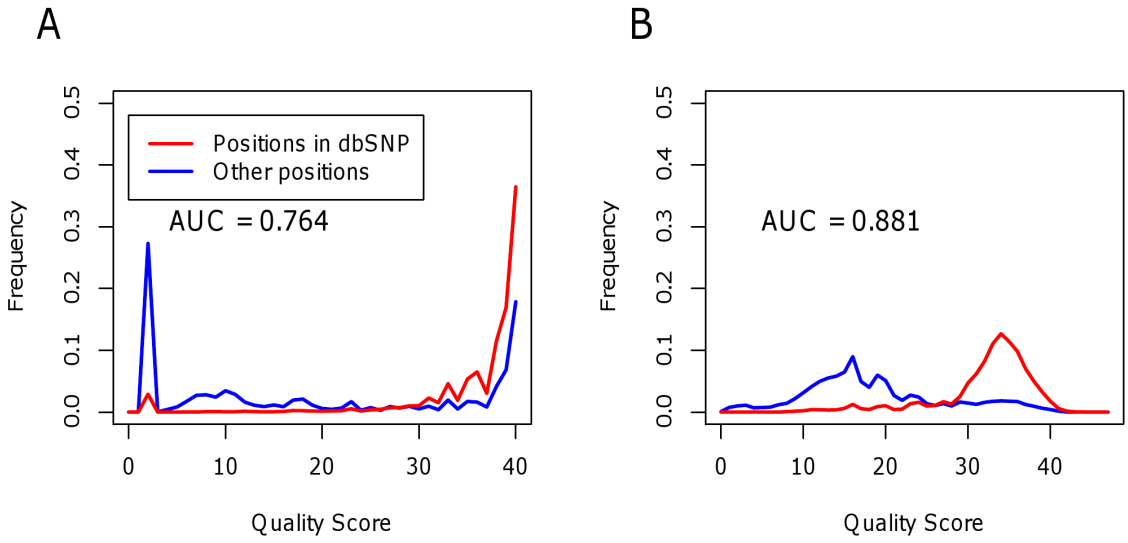

Figure 4.2: Relative frequency distributions of quality scores for bases not matching the reference sequence in chromosome 20 of cell line replicate 3 (trained on chr 10). The non-reference bases are separated as belonging to positions in dbSNP132 (red curve) vs. other positions (blue curve). Plot A shows the distribution of quality scores before ReQON and plot B shows the distribution after ReQON. The area under the ROC curve (AUC) is also reported, which increases after recalibration. This demonstrates that the recalibrated quality scores do a better job of distinguishing sequencing errors from non-errors.

supported by the increased AUC (0.881 vs. 0.764). This analysis shows that, in addition to providing quality scores that more accurately reflect the probability of a sequencing error, the recalibrated quality scores also do a better job of distinguishing sequencing errors from true variants. The AUC statistics for the other two cell line replicates are shown in Table 4.3.

	Original	ReQON	GATK
Replicate 1	0.673	0.806	0.824
Replicate 2	0.746	0.767	0.822
Replicate 3	0.764	0.881	0.874

Table 4.3: Comparison of the area under the ROC curve (AUC) for three cell line replicates recalibrated with ReQON and GATK. Bases from chromosome 20 that do not match the reference sequence are separated as belonging to positions in dbSNP132 or not. Overall, GATK does a slightly better job than ReQON of distinguishing sequencing errors from non-errors, and both recalibration methods outperform the original quality scores.

4.4 Comparison with Competing Methods

There are other available computational tools that recalibrate base quality scores. For example, some variant calling tools, such as Atlas-SNP2 (Shen *et al.*, 2010) and SOAPsnp (Li *et al.*, 2009b), recalibrate the quality scores as part of their algorithm. These tools incorporate the recalibrated qualities into their method, but do not output a file with the recalibrated scores. Therefore, we are not able to compare their performance to ReQON.

The most popular recalibration tool is embedded in the Genome Analysis Toolkit (GATK; DePristo *et al.*, 2011). There are three main differences between the recalibration algorithms of ReQON and GATK. First, GATK chooses not to recalibrate bases with very low quality scores, with the default quality threshold set at 5. Their reasoning is that these quality scores indicate bad or randomly called bases by the sequencer, so these original qualities should be kept as is. This makes sense if these low-quality bases are filtered out before later analyses. However, several NGS analysis tools do not filter out low quality bases and instead weigh all bases based on their quality score. If this is the case, then it makes more sense to recalibrate all bases, regardless of original quality score, which is how ReQON operates. Due to this difference, in general, the low quality bases of GATK have poor accuracy because they are not recalibrated. In contrast, the recalibrated low-quality scores from ReQON are much more accurate. Because computational tools that incorporate base quality scores do so in different ways, it is not clear which of the recalibration methods should be preferred in this regard.

Second, GATK identifies sequencing errors by filtering out known variant positions, then calling all bases that do not match the reference sequence as errors. In reality, some of these bases will not be errors but will be correct calls, such as novel variants. These miscalls disproportionately affect the higher quality scores. Because GATK's observed error rate is approximately the sum of the sequencing error rate, the alignment error rate and the rate of novel variants, GATK will be underestimating the true quality. In contrast, ReQON goes a step further and utilizes information from multiple reads by removing positions from the training set in which we do not have high confidence in our error calls. These removed positions are likely to be novel variants or mismatches due to systematic alignment errors (which we feel should be reflected in the mapping quality score and not the base quality score).

Third, GATK does not specify a training set. Instead, it takes a purely empirical approach, incor-

porating every base in the dataset, except for positions listed in an input file of known variant positions. It classifies each base as an error or non-error, then bins the data according to several covariates, such as original quality score, read position and nucleotide called. GATK calculates empirical error rates for each bin, then assigns each base in that bin the quality score corresponding to that empirical error rate. Assigning quality scores in this manner after classifying each base as an error or non-error raises a major question: why doesn't GATK just assign all error bases a quality of zero, all non-error bases a very high quality, such as 100, and only adjust the quality scores of bases belonging to positions of known variants accordingly? Assigning a full range of quality scores seems much more appropriate when estimating error probabilities based on a model that has been trained on a smaller portion of the data, as ReQON does.

Table 4.4 compares the Frequency-Weighted Squared Error (FWSE) statistic between ReQON and GATK for two separate chromosomes, 10 and 20, on the three cell line replicates described in Section 4.3. ReQON was trained on chromosome 10 and both methods removed positions from dbSNP version 132 during recalibration. The following covariates were used in the GATK model: ReadGroupCovariate, QualityScoreCovariate, DinucCovariate and CycleCovariate. Error calls were made in the same manner as the ReQON algorithm, which we feel more accurately identifies true sequencing errors, as described previously. In every case, FWSE is much lower for quality scores recalibrated with ReQON than GATK. From this, we can conclude that the recalibrated quality scores from ReQON more accurately represent the probability that a base is a sequencing error.

The low-quality bases that GATK chooses not to recalibrate have a large contribution to their FWSE. If we remove these low qualities from the analysis, then FWSE is approximately equal between both methods. However, we feel that all quality scores should be considered because the cutoff below which quality scores are considered low is chosen arbitrarily, and the default cutoff of 5 is not explained by DePristo *et al.* (2011). Additionally, for a more accurate comparison, we could have changed the threshold for GATK and recalibrated all bases regardless of its original quality score. But, as most users will use the default settings when running either recalibration algorithm, we choose to only compare the output using these default settings.

We also investigate how well GATK distinguishes between non-reference bases classified as sequencing errors and non-reference bases belonging to positions of known biological variants. The

	Chromosome 10		Chromosome 20	
	ReQON	GATK	ReQON	GATK
Replicate 1	3.61	11.55	3.04	12.28
Replicate 2	4.61	15.60	4.82	12.91
Replicate 3	5.31	14.91	5.82	17.43

Table 4.4: Comparison of Frequency-Weighted Squared Error (FWSE) for three cell line replicates recalibrated with ReQON and GATK. FWSE is calculated for chromosomes 10 and 20. ReQON outperforms GATK in all cases.

details of this analysis are described at the end of Section 4.3. Table 4.3 on page 57 shows the AUC for the original quality scores and the recalibrated quality scores from ReQON and GATK. Both of the recalibration methods outperform the original quality scores for all three cell line replicates. In general, GATK does a slightly better job than ReQON at distinguishing between errors and non-errors for non-reference bases.

The results in this section and the previous section demonstrate the need for quality score recalibration, especially if these quality scores are to be used in downstream analyses. Recalibrated scores are both more accurate, in the sense that they more closely correspond to the true probability of a sequencing error, and do a better job of discriminating between sequencing errors and non-errors. While GATK is slightly better at discrimination, ReQON produces quality scores that are more accurate. The main differences between ReQON and GATK lie in the underlying assumptions and model used to recalibrate the qualities. We believe that the assumptions of ReQON are more reasonable, and thus, should be preferred over GATK.

4.5 Acknowledgments

I would like to thank Keary Cavin and Chris Bizon from the Renaissance Computing Center at UNC for their help in programming part of the ReQON software in Java. I would also like to thank Joel Parker for recalibrating the data with GATK, which was used in the comparison analysis of Section 4.4. Finally, I would like to thank Matthew Wilkerson for giving me a crash course in processing NGS data. Matt was always more than willing to answer my numerous questions about analyzing and interpreting the results from NGS data.

Bibliography

Ahn J., Marron J.S., Muller K.M. and Chi Y. (2007) The high-dimension, low-sample size geometric representation holds under mild conditions. *Biometrika* **94**, 760-766.

Affymetrix Technical Note (2005) Guide to probe logarithmic intensity error (PLIER) estimation. http://www.affymetrix.com/support/technical/technotes/plier_technote.pdf.

Affymetrix White Paper (2002) Statistical Algorithms Description Document. http://media.affymetrix.com/support/technical/whitepapers/sadd_whitepaper.pdf.

Bravo H.C. and Irizarry R.A. (2010) Model-based quality assessment and base-calling for second-generation sequencing data. *Biometrics* **66**, 665-674.

Cabanski C.R., Qi Y., Yin X., Bair E., Hayward M.C., Fan C., Li J., Wilkerson M.D., Marron J.S., Perou C.M. and Hayes D.N. (2010) SWISS MADE: Standardized WithIn class Sum of Squares to evaluate Methodologies And Dataset Elements. *PLoS ONE* **5**, e9905.

Casella G. and Hwang J.T. (1982) Limit expressions for the risk of James-Stein estimators. *Canad. J. Statist.* **10**, 305-309.

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

Clark M.J., Homer N., O'Connor B.D., Chen Z., Eskin A., Lee H., Marriman B. and Nelson S.F. (2010) U87MG decoded: the genomic sequence of a cytogenetically aberrant human cancer cell line. *PLoS Genet.* **6**, e1000832.

Davies D.L. and Bouldin D.W. (1979) A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **2**, 224-227.

DePristo M.A., Banks E., Poplin R., Garimella K.V., Maguire J.R., Hartl C., Phillippakis A.A., del Angel G., Rivas M.A., Hanna M., McKenna A., Fennell T.J., Kernytsky A.M., Sivachenko A.Y., Cibulskis K., Gabriel S.B., Altshuler D. and Daly M.J. (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **43**, 491-498.

Ding L., Wendl M.C., Koboldt D.C. and Mardis E.R. (2010) Analysis of next-generation genomic data in cancer: accomplishments and challenges. *Hum. Mol. Genet.* **19**, R188-196.

Dohm J.C., Lottaz C., Borodina T. and Himmelbauer H. (2008) Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Res.* **36**, e105.

Dudoit S., Fridlyand J. and Speed T.P. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Amer. Statist. Assoc.* **97**, 77-87.

Ewing B. and Green P. (1998) Basecalling of automated sequencer traces using phred. II. Error probabilities. *Genome Res.* **8**, 186-194.

Fan J. and Lv J. (2008) Sure independence screening for ultrahigh dimensional feature space. *J. R. Stat. Soc. Ser. B* **70**, 849-911.

Fan J. and Ren Y. (2006) Statistical analysis of DNA microarray data in cancer research. *Clin. Cancer Res.* **12**, 4469-4473.

Giancarlo R., Scaturro D. and Utro F. (2008) Computational cluster validation for microarray data analysis: experimental assessment of Clest, Consensus Clustering, Figure of Merit, Gap Statistics and Model Explorer. *BMC Bioinformatics* **9**, 462.

Goya R., Sun M.G.F., Morin R.D., Leung G., Ha G., Wiegand K.C., Senz J., Crisan A., Marra M.A., Hirst M., Huntsman D., Murphy K.P., Aparicio S. and Shah S.P. (2010) SNVMix: predicting single nucleotide variants from next-generation sequencing of tumors. *Bioinformatics* **26**, 730-736.

Hall P., Marron J.S. and Neeman A. (2005) Geometric representation of high dimension, low sample size data. *J. R. Stat. Soc. Ser. B* **67**, 427-444.

Hanley J.A. and McNeil B.J. (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **143**, 29-36.

Irizarry R.A., Hobbs B., Collin F., Beazer-Barclay Y.D., Antonellis K.J., Scherf U. and Speed T.P. (2003) Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **4**, 249-264.

John S. (1972) The distribution of a statistic used for testing sphericity of normal distributions. *Biometrika* **59**, 169-173.

Jung S. and Marron J.S. (2009) PCA consistency in high dimension, low sample size context. *Ann. Statist.* **37**, 4104-4130.

Kaufman L. and Rousseeuw P.J. (1990) *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, New York.

Koboldt D.C., Chen K., Wylie T., Larson D.E., McLellan M.D., Mardis E.R., Weinstock G.M., Wilson R.K. and Ding L. (2009) VarScan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* **25**, 2283-2285.

Kolmogorov A.N. and Rozanov Y.A. (1960) On strong mixing conditions for stationary gaussian processes. *Theory Probab. Appl.* **5**, 204-208.

Lamlertthon W., Hayward M.C. and Hayes D.N. (2011) Emerging technologies for improved stratification of cancer patients: a review of opportunities, challenges, and tools. *Cancer J.* **17**, 451-464.

Langmead B., Trapnell C., Pop M. and Salzberg S.L. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* **10**, R25.

Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**, 1754-1760.

Li H. and Durbin R. (2010) Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics* **26**, 589-595.

Li H. and Homer N. (2010) A survey of sequence alignment algorithms for next-generation sequencing. *Brief Bioinform.* **11**, 473-483.

Li H., Handsaker B., Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R. and 1000 Genome Project Data Processing Subgroup (2009a) The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078-2079.

Li R., Li Y., Fang X., Yang H., Wang J., Kristiansen K. and Wang J. (2009b) SNP detection for massively parallel whole-genome resequencing. *Genome Res.* **19**, 1124-1132.

Mardis E.R. (2008) The impact of next-generation sequencing technology on genetics. *Trends Genet.* **24**, 133-141.

Meyerson M. and Hayes D.N. (2005) *Microarray Approaches to Gene Expression Analysis*. Humana Press, Totowa, NJ, 121-148.

Meyerson M., Gabriel S. and Getz G. (2010) Advances in understanding cancer genomes through second-generation sequencing. *Nat. Rev. Genet.* **11**, 685-696.

Millenaar F.F., Okyere J., May S.T., van Zanten M., Voesenek L.A. and Peeters A. J. (2006) How to decide? Different methods of calculating gene expression from short oligonucleotide array data will give different results. *BMC Bioinformatics* **7**, 137.

Muirhead R.J. (2005) *Aspects of Multivariate Statistical Theory*. Wiley, Hoboken, NJ, 380-428.

Novoradovskaya N., Whitfield M.L., Basehore L.S., Novoradovsky A., Pesich R., Usary J., Karaca M., Wong W.K., Aprelikovaa O., Fero M., Perou C.M., Botstein D. and Braman J. (2004) Universal Reference RNA as a standard for microarray experiments. *BMC Genomics* **5**, 20.

Provenzano M. and Mocellin S. (2007) Complementary techniques: validation of gene expression data by quantitative real time PCR. *Adv. Exp. Med. Biol.* **593**, 66-73.

Qiao X., Zhang H.H., Liu Y., Todd M.J. and Marron J.S. (2010) Weighted Distance Weighted Discrimination and its asymptotic properties. *J. Amer. Statist. Assoc.* **105**, 401-414.

Shao J. (2003) *Mathematical Statistics, 2nd ed*. Springer, New York, NY, 60.

Shen Y., Wan Z., Coarfa C., Drabek R., Chen L., Ostrowski E.A., Liu Y., Weinstock G.M., Wheeler D.A., Gibbs R.A. and Yu F. (2010) A SNP discovery method to assess variant allele probability from next-generation resequencing data. *Genome Res.* **20**, 273-280.

Sherry S.T., Ward M. and Sirotkin K. (1999) dbSNP-database for single nucleotide polymorphisms and other classes of minor genetic variation. *Genome Res*. **9**, 677-679.

Sorlie T., Perou C.M., Tibshirani R., Aas T., Geisler S., Johnsen H., Hastie T., Eisen M.B., van de Rijn M., Jeffrey S.S., Thorsen T., Quist H., Matese J.C., Brown P.O., Botstein D., Eystein Lonning P. and Borresen-Dale A.L. (2001) Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Natl. Acad. Sci. USA* **98**, 10869-10874.

Trapnel C., Pachter L. and Salzberg S.L. (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics* **25**, 1105-1111.

Venter J.C., Levy S., Stockwell T., Remington K. and Halpern A. (2003) Massive parallelism, randomness and genomic advances. *Nat. Genet.* **33** (Suppl.), 219-227.

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

Wang K., Singh D., Zeng Z., Coleman S.J., Huang Y., Savich G.L., He X., Mieczkowski P., Grimm S.A., Perou C.M., MacLeod J.N., Chiang D.Y., Prins J.F. and Liu J. (2010) MapSplice: accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Res.* **38**, e178.

Wu T.D. and Nacu S. (2010) Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics* **26**, 873-881.

Yata K. and Aoshima M. (2012) Effective PCA for high-dimension, low-sample-size data with noise reduction via geometric representations. *J. Multivariate Anal.* **105**, 193-215.

Zhang J., Chiodini R., Badr A. and Zhang G. (2011) The impact of next-generation sequencing on genomics. *J. Genet. Genomics* **38**, 95-109.