NONPARAMETRIC METHODS FOR MACHINE LEARNING AND ASSOCIATION TESTING

Erika S. Helgeson

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 Biostatistics in the Gillings School of Global Public Health.

Chapel Hill
2017

Approved by:

Eric Bair

Michael Kosorok

Yufeng Liu

J. S. Marron

Gary Slade

ABSTRACT

Erika S. Helgeson : Nonparametric methods for machine learning and association testing
(Under the direction of Eric Bair)

As data collection becomes easier, non-parametric machine learning methods are increasing in popularity due to their ability to quickly discover informative data structures useful for prediction. Unsupervised clustering methods can be especially valuable for identifying subgroups in high dimensional gene expression data. Another important goal is prediction of disease or symptom outcomes from a given genotype. One way to achieve this goal is to first identify genetic factors associated with the outcome of interest. This may be especially challenging if the association is investigated in a sample with selection stratified with respect to a third variable, as occurs when studying secondary phenotypes in case-control studies.

In Chapter 2 we develop a non-parametric, cluster significance testing algorithm. This algorithm compares the strength of identified clusters to the strength of spurious clusters produced from unimodal reference data. The method utilizes dimension reduction and sparse covariance estimation, making it is especially relevant for high dimensional data sets. We also extend the method to estimate the number of clusters present. The method is applied to several simulated and real-world data sets. We find it has comparable accuracy to existing methods and, in addition, can be used in a wider array of settings.

We next develop a permutation-based sparse biclustering algorithm built upon the method of Witten and Tibshirani (2010) which iteratively employs a cluster significance testing step. Biclustering identifies a submatrix such that the pattern of the features for the observations within the submatrix are different than the pattern outside of the submatrix. We present simulation and real data results with comparison to existing methods illustrating

the accuracy of the proposed method in assigning observations to clusters and identifying distinguishing features.

In the last chapter we develop a permutation-based method for assessing the association between genetic factors and secondary phenotypes within a case control study. Conventional inverse-probability-of-sampling-weighted (IPW) regression (Monsees et al., 2009 and Richardson et al., 2007) may produce invalid estimates of association strength in situations where most of the variation in the secondary phenotype is found in the cases. Simulation and real data results indicate the proposed method has better type-I error rates and comparable power to the conventional IPW method and can be used to identify novel SNPs associated with clinical orofacial pain.

I would like to dedicate this Doctoral dissertation to the teachers and professors who believed in me, challenged me, and encouraged me to never give up on my dreams.

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Dr. Eric Bair, for persuading me to come to UNC and for his help, guidance, and entertaining stories throughout my time here.

Secondly, I would like to thank my committee members: Dr. Michael Kosorok, Dr. Yufeng Liu, Dr. Steve Marron, and Dr. Gary Slade for taking the time to review my work and for their helpful advice.

Next, I would like to acknowledge Qian Liu for her preliminary work in developing a sparse-clustering based biclustering method.

Last, but not least, I would like to thank my friends, family, and supporting crew, "Team Helgeson." I would not be where I am today without your love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANOVA	Analysis of Variance
AS	Aftersensation
ASE	Average Standard Error
ASW	Average Silhouette Width
AUC	Area Under the Curve
Auto	Autonomic
Ave	Average
BCs	Biclusters
BCSS	Between Cluster Sum of Squares
BFS	Bootstrapping for Significance
BIC	Bayesian Information Criterion
BP	Blood Pressure
CH	Calinski-Harabasz
CI	Cluster Index
CSQ	Coping Strategies Questionnaire
DF	Degrees of Freedom
EM	Expectation-Maximization
EPQ-R	Eysenck Personality Questionnaire-Revised
FDRs	False Discovery Rates
FMR	Feature Misclassification Rate
FNR	False Negative Rate
FPR	False Positive Rate
GWAS	Genome Wide Association Study
HDLSS	high-dimension, low-sample size
HRV	Heart Rate Variability
HSSVD	Heterogeneous SSVD

IGP	In-Group Proportion
IP	Interpersonal
IPW	Inverse-Probability-of-sampling-Weighted
KDE	Kernel Density Estimation
KRS	Kohn Reactivity Scale
LAS	Large Average Submatrix
LES	Life Experiences Survey
LumA	Luminal A
LumB	Luminal B
LumI	Luminal I
MAD	median absolute deviation from the median
MAF	Minor Allele Frequency
MEM	Modal ExpectationMaximization
MPC	Multimodality of Principal Curves
NA	Not Applicable
Obs	Observations
OMR	Object Misclassification Rate
OPPERA	Orofacial Pain: Prospective Evaluation and Risk Assessment
OR	Odds Ratio
PCS	Pain Catastrophizing Scale
PILL	Pennebaker Inventory of Limbic Languidness
POMS	Profile of Mood States
PPT	Pressure Pain Threshold
PSQI	Pittsburgh Sleep Quality Index
Psy	Psychosocial
QST	Quantitative Sensory Testing
Q-Q	Quantile-Quantile

SCBiclust	Sparse-Clustering Biclustering
SCL	Symptom Checklist
SE	Standard Error
Sec	Seconds
SigClust	Statistical Significance of Clustering
SNP	Single-Nucleotide Polymorphism
sparseBC	Sparse Biclustering
SS	Single Stimulus
SSVD	Sparse Singular Value Decomposition
SVD	Singular Value Decomposition
TCI	Theoretical Cluster Index
TMD	Temporomandibular Disorder
TS	Temporal Summation
TTSS	Theoretical Total Sum of Squares
TWSS	Theoretical Within Cluster Sum of Squares
UN-PaC	Unimodal Non-Paramateric Cluster
Var	Variance
WCD	Within Cluster Dispersion

CHAPTER 1: LITERATURE REVIEW

1.1 Cluster Significance Testing and Cluster Number Estimation

Often in an exploratory analysis of a data set it is of interest to discover if there are any natural groupings present. Graphical visualizations can be useful for low dimensional data sets, but for higher dimensions other methods are needed. Clustering techniques are common tools used in the exploration of complex data sets. Since no response variable is identified, and no model is being constructed, clustering falls under the category of unsupervised learning. The versatility of clustering lies in the fact that it can be used in any data set where the similarity between observations can be measured. Many clustering algorithms have been proposed including hierarchical clustering, k-means clustering, and spectral and graph based methods. The exploratory nature of clustering has been very useful for identifying patterns in the field of bioinformatics. Unfortunately, most clustering methods will always group the data into clusters even if the data is actually homogeneous. Given that there are so many clustering methods available it is important to know whether the identified clusters do, in fact, represent distinct subgroups or if they are merely a spurious finding. Another related problem is being able to estimate the number of clusters present in the data set.

Several methods have been developed which allow the strength of identified clusters to be assessed. These methods generally test the null hypotheses that the data is homogeneous by comparing a statistic from the clustered data to the statistic from an appropriate null distribution. This null distribution is generated under certain assumptions such as a specific parametric distribution or non-parametric assumptions about the shape of unclustered data. In the first part of the dissertation we develop a non-parametric cluster significance testing

algorithm where the null data is generated under a unimodal assumption. We then compare our algorithm with existing cluster validation approaches in a series of simulation studies.

A conventional way to determine the number of clusters present in the data is to examine the within-cluster homogeneity or between-cluster heterogeneity, or a combination of both measures, over a range of cluster numbers. The optimal number of clusters, k, is chosen to maximize this value. Another way to estimate the number of clusters is examine the stability of the clustering. If the addition of new data does not affect the cluster centroids, it is assumed that the number of clusters is correct. The number of clusters can also be assessed by examining prediction accuracy, the ability to correctly predict class labels. Many of these methods incorporate a null distribution, either through data permutation or bootstrapping, in order have a more rigorous assessment of cluster number. We propose a method that uses a unimodal null distribution and we choose the number of clusters to be the value that maximizes the difference in cluster strength calculated from the observed data and null data.

1.1.1 An Overview of Cluster Significance Testing Approaches

Several cluster significance tests have been proposed based on various assumptions about cluster strength and cluster shape. Liu et al. (2008) take a parametric approach for their SigClust method and define a cluster to be data coming from a single Gaussian distribution. The significance test of Maitra et al. (2012) avoids defining a parametric distribution, but assumes that the clusters are compact and can be transformed to be spherical and similar to each other. Ahmed and Walther (2012) reduce a multidimensional data set to a single dimension by projecting the data onto its principle curve. They then employ Silverman's bandwidth test (Silverman, 1981) to test for more than one mode. Cheng and Ray (2014) take a local modal inference approach and test if two identified modes come from a unimodal or bimodal distribution. Kapp and Tibshirani (2006) use a validation measure based on the proportion of observations in a cluster whose nearest neighbors are also in the same

cluster. In this section we briefly describe the above methods which will be compared to our proposed method in the first part of the dissertation.

1.1.1.1 SigClust

The SigClust method developed by Liu et al. (2008) defines a cluster to be data generated from a single multivariate Gaussian distribution. To assess the presence of clustering the method compares the two-means clustering index (CI) calculated from the data to the CI calculated from a single multivariate Gaussian distribution. To optimize this method for high-dimension, low-sample size (HDLSS) settings, such as microarray data, the method combines invariance principles and factor analysis to estimate the covariance matrix. As an extension, Huang et al. (2015) incorporate a soft thresholding approach to reduce type-I error inflation.

Assume the observed data, X, is of dimension $d \times n$, with d features and n observations and a clustering algorithm has split the data into two clusters. Liu et al. (2008) use the CI as the test statistic for their hypothesis test. The CI is defined to be:

$$CI = \frac{\sum_{k=1}^{2} \sum_{j \epsilon C_k} \left\| x_j - \bar{x}^k \right\|^2}{\sum_{j=1}^{n} \left\| x_j - \bar{x} \right\|^2} \tag{1.1}$$

Where C_k is the sample index set for the kth cluster, $\bar{x}^k$ is the sample mean for the kth cluster, and k=1 and 2. Note, the CI is location and rotation invariant and a small value for the CI indicates that a large proportion of the variance is explained by the given clustering.

To test the null hypothesis Liu et al. (2008) compare the CI from the observed data to a CI generated from a null Gaussian distribution, represented as $N(\mu, \Sigma)$. Since the CI is location invariant only the covariance matrix needs to be estimated. Liu et al. (2008) make use of the rotational invariance of the CI and use a factor analysis model to greatly reduce the number of parameters necessary for estimating the covariance matrix. As a first step, eigendecomposition is used to reduce Σ to $\Sigma = MDM^T$ where M is an orthogonal

matrix and D is a diagonal matrix with entries $\lambda_1, \ldots, \lambda_d$ equal to the eigenvalues of Σ. Replacing X with $M^T X$ results in the covariance matrix being the d-parameter matrix D. D is further simplified by assuming that the covariance has relatively few biologically meaningful components. Liu et al. (2008) model D as:

$$D = \Sigma_B + \sigma_N^2 xI \tag{1.2}$$

The diagonal matrix Σ_B represents the real biology and typically has a sparse structure. σ_N^2 represents the level of background noise and is estimated as

$$\hat{\sigma}_N^2 = \frac{\text{MAD}_{d \times n \, data \, set}}{\text{MAD}_{N(0,1)}} \tag{1.3}$$

Where MAD is the median absolute deviation from the median.

Lastly, D is estimated by $\hat{D}$ which is a diagonal matrix of elements $\hat{d}_j$ where

$$\hat{d}_j = \begin{cases} \tilde{\lambda}_j - \tau \; if \; \tilde{\lambda}_j > \tau + \sigma_N^2 \\ \sigma_N^2 \; if \; \tilde{\lambda}_j \leq \tau + \sigma_N^2 \end{cases} \tag{1.4}$$

For the original "hard thresholding" method (Liu et al., 2008), $\tau = 0$. For the "soft-thresholding" method (Huang et al., 2015), the positive constant τ is chosen such that $\Sigma_{k=1}^{d}(\frac{1}{\sigma_N^2} - \frac{1}{\tilde{\lambda}_k - \tau})_+ = M$ and M is chosen by setting the sum of the eigenvalues for the estimated covariance matrix equal to the sum of the sample covariance.

The reference distribution is then generated from $(x_1, ..., x_d)$ where $x_j \sim N(0, \hat{d}_j)$. A p-value can then be calculated by comparing the CI from the reference distribution to the CI from the observed data.

In the second part of the dissertation we will compare our proposed significance testing method to SigClust method using hard thresholding and SigClust using soft thresholding

with 1000 permutation replications each. Our method also uses the cluster index given in equation 1.1 as a test statistic.

1.1.1.2 Bootstrapping for Significance of Clustering

Maitra et al. (2012) take a different approach to formulating a cluster significance test by assuming clusters are compact and can be transformed to be spherical and similar. Their method is distribution-free and instead uses a bootstrap approach for testing the null hypothesis that a smaller model (less clusters) better fits the data than a larger model (more clusters). This method can also be applied to estimate the number of clusters present. Unlike SigClust this method is not optimized for HDLSS data.

To illustrate the method first let the original data set be represented by $X_1, X_2, \ldots, X_n$ where X_i is a p-dimensional vector. Assume that each $X_i \sim \sum_{k=1}^{K} \zeta_{ik} f_k(x)$ for $i = 1, \ldots, n$ where K is the number of clusters; f_k is the density of an observation in the kth cluster; $\zeta_{ik} = I_{X_i \epsilon \mathcal{G}_k}$ where $I_{(\cdot)}$ is the indicator function and G_k is the sample set for the kth cluster. Under this framework, the null hypothesis, the data is from a family of distributions with K clusters, is compared to the alternative, the data comes from a family of distributions with more than K clusters. As a test statistic, Maitra et al. (2012) use the improvement of the within-cluster sum of squares when the data is grouped into K* (>K) clusters instead of K clusters, $s_{K;K^*} = W_K - W_{K^*}$ where $W_K = \sum_{i=1}^{n} \sum_{k=1}^{K} \zeta_{ik}^K (x_i - \mu_k)'(x_i - \mu_k)$. Here μ_k represents the mean for the kth cluster.

For the case when K=0, no clustering of the data, the authors propose that the null distribution should be generated by sampling from the uniform distribution on the p-dimensional hyper-rectangle as in Tibshirani et al. (2001). For the $K > 0$ scenario the authors develop methodologies for generating the null distribution under the assumption of homogeneous spherical structures and then general ellipsoidal clusters.

For the homogeneous spherical structures case: under the null hypothesis the sample $\Xi = X_1, \ldots, X_n$ is jointly distributed as: $\prod_{i=1}^{n} \sum_{k=1}^{K} \zeta_{ik}^K * \frac{1}{\sigma} * g(\frac{x_i - \mu_k}{\sigma})$ Where g :

$\mathbb{R}^p \to \mathbb{R}^p$ is invariant under orthogonal transformation. Using the K-clusters solution the following estimates can be obtained: $\hat{\zeta}_{ik}^K$'s, $\hat{\mu}_k = \sum_{i=1}^n \hat{\zeta}_{ik}^K X_i / \sum_{i=1}^n \hat{\zeta}_{ik}^K$, and $\hat{\sigma} = \frac{1}{np} \sum_{i=1}^n \sum_{k=1}^K \zeta_{ik}^K (X_i - \hat{\mu}_k)'(X_i - \hat{\mu}_k)$.

The residuals from this solution, $\hat{\epsilon}_1, \hat{\epsilon}_2, \ldots, \hat{\epsilon}_n$, where $\hat{\epsilon}_i = (X_i - \sum_{k=1}^K \hat{\zeta}_{ik}^K \hat{\mu}_k)/\hat{\sigma}$ now form a sample for the density $g(\cdot)$, but the following resampling strategy is needed to increase power. Let $W_i = Z_i / \|Z_i\|$ where Z_i is a p-variate standard normal random vector. A random permutation $(\ell_1, \ldots, \ell_n)$ of $1, \ldots, n$ is generated. The ith resampled residual is then given by $\epsilon_i^* = \|\hat{\epsilon}_{\ell_i}\| W_i$. The resampled null distribution is then given by $\Xi^* = X_1^*, \ldots, X_n^*$ where $X_i^* = \sum_{k=1}^K \hat{\zeta}_{ik}^K \hat{\mu}_k + \hat{\sigma}\epsilon_i^*$.

For the general ellipsoidal clusters case a similar resampling strategy is used, but the residuals are defined differently. Under the null hypothesis the sample $\Xi = X_1, \ldots, X_n$ is jointly distributed as: $\Pi_{i=1}^n \sum_{k=1}^K \zeta_{ik}^K \frac{1}{\det(|\Sigma_k|^{\frac{1}{2}})} g(\Sigma_k^{-1/2}(x_i - \mu_k))$ where $g : \mathbb{R}^p \to \mathbb{R}^p$. Using the K-clusters solution the following estimates can be obtained: $\hat{\zeta}_{ik}^K$'s, $\hat{\mu}_k = \sum_{i=1}^n \hat{\zeta}_{ik}^K X_i / \sum_{i=1}^n \hat{\zeta}_{ik}^K$, and $\hat{\Sigma}_k = \sum_{i=1}^n \hat{\zeta}_{ik}^K (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)' / \sum_{i=1}^n \hat{\zeta}_{ik}^K$. The residuals are given by $\hat{\epsilon}_1, \hat{\epsilon}_2, \ldots, \hat{\epsilon}_n$ where $\hat{\epsilon}_i = \sum_{k=1}^K \hat{\zeta}_{ik}^K \Sigma_k^{-\frac{1}{2}} (X_i - \Sigma_{k'=1}^K \hat{\zeta}_{ik'}^K \hat{\mu}_{k'})$. The resampled null distribution is then given by $\Xi^* = X_1^*, \ldots, X_n^*$ where $X_i^* = \sum_{k=1}^K \hat{\zeta}_{ik}^K (\hat{\mu}_k - \hat{\Sigma}_k^{\frac{1}{2}} \epsilon_i^*)$.

To calculate the p-value, the null distribution resampling process is repeated to obtain M data sets. For each null data set Ξ_j^* the test statistic $s_{j,(K;K^*)}^*$ is calculated. The p-value for the $s_{K;K^*}$ calculated from the original data set is the proportion of times $s_{j,(K;K^*)}^*$ is greater than $s_{K;K^*}$.

Since the inverse of the covariance matrix is required for clustering the null distribution when clusters are non-homogeneous the method is not applicable to the HDLSS setting. For situations when the number of features is less than the number of observations we will compare our proposed method to the method of Maitra et al. (2012) under the assumption that the clusters are heterogeneous.

1.1.1.3 Principal Curve Unimodality Test

If one were to define a cluster as a single unimodal distribution then cluster significance testing would simply reduce to testing for multimodality. Thus if more than one mode is present in the data it signifies that multiple clusters are present. Much work has been done in determining the number of modes present in univariate data, but work in the multivariate case has been more limited. This is due to many factors including the complication of defining multivariate unimodality, the "curse of dimensionality", and computational constraints. Instead of tackling the problem of multivariate modality testing, Ahmed and Walther (2012) assess the number of subpopulations by first reducing the data to one dimension using principal curves and then employing Silverman's bandwidth test (Silverman, 1981).

The principal curve of the data $X \epsilon \mathbb{R}^p$ is defined to be

$$f(\lambda) = \mathrm{E}(X|\lambda_f(X) = \lambda) \tag{1.5}$$

Where $f(\lambda)$ is a smooth curve in $\mathbb{R}^p$, λ is a real variable, and the projection index $\lambda_f(X)$ is the largest value of λ for which $f(\lambda)$ is closest to X. The principal curve is the mean of the points that project onto it and thus gives a univariate approximation of X. Using the nonlinear principal curve is much more flexible than a linear projection and is more likely to avoid the problem of projecting different clusters near the same location. See (Hastie and Stuetzle, 1989) and (Tarpey and Flury, 1996) for more information on principal curves.

After the data has been reduced to one-dimension, Silverman's bandwidth test (Silverman, 1981) is used to test for the presence of more than one cluster. The null hypothesis for this test is that the univariate data has k modes, here k=1, and it is tested against the alternative that $k^* > k$ modes are present. As a first step, the data is approximated using the kernel density estimate:

$$\hat{f}(t;h) = (nh)^{-1} \sum_{i=1}^{n} K(h^{-1}(t - X_i)) \tag{1.6}$$

Where h is the bandwidth; $K(\cdot)$ is the Gaussian kernel function; and $X_1, \ldots, X_n$ are the observed data. Silverman (1981) shows that a critical bandwidth, h_k can be determined such that

$$h_k = \inf\{h : \hat{f}(\cdot; h) \; has \; at \; most \; k \; modes\}. \tag{1.7}$$

Larger values of h_1 indicate that more smoothing must be done to make the data unimodal. To test for unimodality the critical bandwidth, h_1, is compared to the bandwidth from a reference distribution which is known to be unimodal. Observations, y_i, from the reference unimodal distribution $\hat{f}(\cdot; h_1)$ are generated by:

$$y_i = (1 + h_1^2/\sigma^2)^{-1/2}(X_{I(i)} + h_1\epsilon_i) \tag{1.8}$$

Where $\epsilon_i \sim N(0, 1)$; σ^2 is the sample variance; and the $X_{I(i)}$'s are sampled uniformly, with replacement, from the data X. The bootstrap critical bandwidth, h_i, is determined for each bootstrap replication. The p-value for the test of the null hypothesis is given by:

$$p = \sum_{1}^{B} \frac{I(h_i > h_1)}{B} \tag{1.9}$$

where B is the number of bootstrap replicates and $I(\cdot)$ is the indicator function.

The method proposed by Ahmed and Walther (2012) can be used to test for clustering, assuming that a single cluster would have at most 1 mode. It can be extended to find the number of clusters present through an iterative mode testing process. For algorithm comparison in later sections we will use the default number of bootstrap simulations, B=10,000. The method we propose in the Section 2.2 uses a process similar to equation 1.8 to generate the null distribution for hypothesis testing.

1.1.1.4 Multivariate Modality Inference

Instead of reducing the data to a univariate representation and performing a global modality test, Cheng and Ray (2014) use a local test to assess modes identified in an estimate of the multivariate density. Their method uses the Model EM method and Ridgeline EM method of Li et al. (2007) to identify potential modes and saddle points, respectively. The test statistic is based on the difference between the estimated density at the mode and saddle point. Significance of the modes is assessed by comparing this statistic to its asymptotic distribution.

Specifically, Cheng and Ray (2014) test the null hypothesis that pair of identified modes, $\boldsymbol{x_{m_1}}$ and $\boldsymbol{x_{m_2}}$ come from a unimodal density against the alternative that they are from a bimodal density. As a first step in this process, a sphering transformation is applied to the data and the multivariate kernel density estimator is used as a non-parametric estimate of the probability density. For simplicity, only one bandwidth parameter, h, is used in the estimation of the density $\hat{f}(\boldsymbol{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} K(\frac{\boldsymbol{x}-\boldsymbol{X_i}}{h})$. $\boldsymbol{X_i} = (X_{i1}, ..., X_{id})$; i=1,...,n are are i.i.d samples from the population with some unknown probability density *f*. $K(\cdot)$ is a real-valued multivariate kernel function which here is chosen to be the multivariate normal density function. The authors recommend that the "normal reference rule" (Scott, 1992) be used for estimating the bandwidth which in this case is given by $h = \{\frac{4}{(d+2)n}\}^{\frac{1}{d+4}}$.

Once the multivariate density has been estimated putative modes are identified using the Modal EM (MEM) algorithm (Li et al., 2007). The MEM uses an EM-style algorithm to solve for the local maxima of a mixture density. Li et al. (2007) also proposed Ridgeline EM for finding the ridgeline between two modes. Once the ridgeline is obtained, the saddle point, the point on the ridgeline with the lowest density, can easily be computed.

To test the null hypothesis the estimated density at the smallest mode (denoted $\hat{\boldsymbol{x}}_{\boldsymbol{m}}$) is compared to the density at the saddle point denoted $\hat{\boldsymbol{x}}_{\boldsymbol{s}}$. Cheng and Ray (2014) recommend re-estimating the density using the Gaussian kernel density estimator with bandwidth chosen to be $h = (\frac{c}{n})^{\frac{\gamma}{d+4}}$ where $1 < \gamma < 1 + \frac{4}{d}$ and $c = \frac{4}{d+2}$. The authors then show that the test

statistic, $\sqrt{\hat{f}(\hat{\boldsymbol{x}}_{\boldsymbol{m}})} - \sqrt{\hat{f}(\hat{\boldsymbol{x}}_{\boldsymbol{s}})}$ is asymptotically distributed as $N\left(0, \frac{1}{2nh^d}\left(\frac{1}{2\sqrt{\pi}}\right)^d\right)$. The asymptotic distribution can then be used to make inference about the modality of the density containing the two putative modes.

1.1.1.5 IGP

A closely related concept to cluster significance is cluster reproducibility. If clusters are found in one data set we hope that they would be found in similar data sets. Kapp and Tibshirani (2006) develop a validation procedure for cluster reproducibility based on the in-group proportion (IGP) which is defined to be the proportion of observations in a cluster whose nearest neighbors are also located in the same cluster. The IGP procedure can be used to validate individual clusters and therefore test cluster significance.

Let A be an $p \times n$ data set with p features and n observations which are classified into k clusters. Let C be the $p \times k$ matrix of cluster centroids. Suppose another data set X is collected with the same p features and q observations such that each observation can be classified into one of the k clusters or a "below cutoff group". The cluster membership of an observation, j, in X can be determined using the rule:

$$Class_X(j) = \begin{cases} 0 & if\ \max_{1\le u\le k}\ d(X[,j], C[,u]) < c, \\ \arg\max_{1\le u\le k} d(X[,j], C[,u]), & if\ \max_{1\le u\le k}\ d(X[,j], C[,u]) \ge c \end{cases}$$

Where the function d(x,y) is Person's centered correlation for vectors x and y. The cluster membership function is equivalent to classifying an observation to a cluster with whose centroid it most highly correlates. Observations whose correlations with cluster centroids are less than the cutoff, c, are classified to a "below cutoff group"

Once cluster membership has been established the IGP can be calculated and the reproducibility of the clusters can be assessed. Let u be the cluster label for all the observations

whose $Class_X = u$. Then the in-group proportion is defined to be:

$$IGP(u, X) = \frac{\#\{j|Class_X(j) = Class_X(j^N) = u\}}{\#\{j|Class_X(j) = u\}}$$

For the jth observation in X, where $j^N = \arg\max_{i \neq j} d(X[, j], X[, i])$.

To validate the clusters identified in X, the IGPs from the data set are compared to the IGPs from a null distribution in which centroids are randomly placed in the data. Kapp and Tibshirani (2006) propose a null distribution in which the samples are permuted within a box aligned with their principal components. Since the IGP depends on the size of the cluster, the IGP for a given cluster is compared only to IGPs from null distributions that came from groups of the same size. The p-value for this test is the fraction of the IGPs from the null distribution that are as close or closer to 1 than the IGP for the cluster identified in the data set.

For the comparison of methods in the first part of the dissertation, clusters were identified from a sample simulation and the reproducibility of the clusters is assessed in an additional generation of the same data set. Default settings were used including no cutoff for clustering.

1.1.2 An Overview of Cluster Number Estimation Approaches

There are several available approaches for estimating the number of clusters in a data set. Rousseeuw (1987) and Calinski and Harabasz (1974) propose choosing the number of clusters which maximizes their proposed measure of cluster strength. Tibshirani et al. (2001) incorporate a reference distribution with no clusters into their method and choose the number of clusters to maximize the difference between the within-cluster dispersion from the data and the reference distribution. Fang and Wang (2012) tackle the problem by choosing the number of clusters which minimizes cluster instability. These methods will be compared to our proposed method in the first part of the dissertation.

1.1.2.1 Calinski-Harabasz index

Let $W(k)$ be the within-cluster sum of squares of a k-cluster assignment of the data. $W(k) = \sum_{r=1}^{k} \frac{1}{n_r} D_r$ where n_r is the number of observations in cluster r and $D_r = \sum_{i,i' \epsilon C_r} d_{ii'}^2$ with $d_{ii'}^2 = \sum_j (x_{ij} - x_{i'j})^2$, $i \neq i'$. Let B_k be the between-cluster sum of squares for a k-cluster assignment of the data. $B(k) = \frac{1}{n} \sum_{i,i'=1}^{n} d_{ii'}^2 - W(k)$

The Calinski-Harabasz index (Calinski and Harabasz, 1974) is defined to be

$$CH(k) = \frac{B(k)(n-k)}{W(k)(k-1)}$$

The estimated number of clusters is chosen to be the k which maximizes $CH(k)$.

1.1.2.2 Average silhouette width

Rousseeuw (1987) proposed choosing the number of clusters which maximizes the "average silhouette width," a measure assessing cluster tightness and separation. Borrowing notation from (Hennig, 2014), let $d(x_i, x)$ represent the dissimilarity between object x_i and object x. Let $a(i,k) = \frac{1}{|C_j|-1} \sum_{x \in C_j} d(x_i, x)$ where C_j represents the cluster containing x_i; and let $b(i,k) = \min_{C_l \neq C_j} \frac{1}{|C_l|} \sum_{x \in C_l} d(x_i, x)$. Then for a given object, x_i, and number of clusters k the silhouette is defined to be

$$s(i,k) = \frac{b(i,k) - a(i,k)}{\max\{a(i,k) - b(i,k)\}} \tag{1.10}$$

The estimated number of clusters k^* is chosen to maximize $\frac{1}{n} \sum_{i=1}^{n} s(i,k)$.

1.1.2.3 Gap statistic

Tibshirani et al. (2001) propose the "gap statistic", a measure based on the within-cluster dispersion, to estimate the number of clusters present.

Suppose X is an $n \times p$ data set with n independent observations and p features which has been grouped into k clusters. Let $d_{ii'} = \sum_j (x_{ij} - x_{i'j})^2 \; i \neq i'$. Then the pooled-within-cluster sum of squares around the cluster means is defined to be $W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r$ where $D_r = \sum_{i,i' \epsilon C_r} d_{ii'}$, and n_r is the number of observations in cluster r.

To determine the number of clusters, the distribution of $\log(W_k)$ is compared to its expectation in a reference distribution. The features for the reference distribution are generated from a uniform distribution over a box aligned with the principal components of the data.

Define the gap statistic to be

$$Gap_n(k) = \mathrm{E}_n^*\{\log(W_k)\} - \log(W_k) \tag{1.11}$$

Where E_n^* represents the expectation from the reference distribution with sample size n. The number of clusters, $\hat{k}$, is chosen to maximize $Gap_n(k)$, equivalently the point in the distribution at which the two curves are farthest apart.

To carry out this method an iterative process then employed such that for each choice of k, B reference data sets are generated. The gap statistic for k is given by $Gap(k) = (1/B) \sum_b \log(W_{kb}^*) - \log(W_k)$ and the standard error is given by $s_k = [(1/B) \sum_b \{\log(W_{kb}^*) - \bar{l}\}^2]^{\frac{1}{2}} \sqrt{1 + 1/B)}$ where $\bar{l} = (1/B) \sum_b \log(W_{kb}^*)$. To account for variability, the number of clusters, $\hat{k}$, is chosen to be the smallest k such that $Gap(k) \geq Gap(k+1) - s_{k+1}$

The gap statistic can be adapted for any clustering algorithm and the authors illustrate its flexibility in identifying the number of clusters identified though k-means and hierarchical clustering. A limitation of the gap statistic is it assumes that the clusters are well-separated and uniform.

1.1.2.4 Prediction Strength

Tibshirani and Walther (2005) take a different approach and look at determining the number of clusters as a model selection problem where the number of clusters is chosen to maximize prediction strength. The authors propose a two-fold cross validation method.

First, the $n_{tr} \times p$ training data, X_{tr} is grouped in k clusters via the clustering operation $C(X_{tr}, k)$. Let $D[C(\dots), X_{tr}]$ be a $n_{tr} \times n_{tr}$ matrix with ii'th element $D[C(\dots), X_{tr}]_{ii'} = 1$ if observations i and i' fall into the same cluster and zero otherwise.

Let X_{te} be an independent test sample of size $n_{te} \times p$ drawn from the same population as the training set. Cluster the test set into k-clusters via an operation $C(X_{te}, k)$. Summarize the cluster co-memberships via the $n_{te} \times n_{te}$ matrix $D[C(X_{te}, k), X_{te}]$.

Let $A_{k1}, A_{k2}, \dots, A_{kk}$ be the indices of the test observations in test clusters $1, 2, \dots, k$. Let $n_{k1}, n_{k2}, \dots, n_{kk}$ be the number of observations in these clusters. The "prediction strength" of the clustering $C(\cdot, k)$ is defined by

$$ps(k) = \min_{1 \le j \le k} \frac{1}{n_{kj}(n_{kj} - 1)} \sum_{i \neq i' \in A_{kj}} D[C(X_{tr}, k), X_{te}]_{ii'} \tag{1.12}$$

Equivalently, ps(k) equals the proportion of observation pairs in that cluster that are also assigned to the same cluster by the training set centroids. The largest value of k which gives a prediction strength higher than a chosen threshold (typically between 0.8 and 0.9) is estimated to be the number of clusters in the data set.

1.1.2.5 Cluster Instability

Fang and Wang (2012) take on the problem of cluster number estimation by choosing the number of clusters which minimizes cluster instability where instability is estimated using the bootstrap. Let $X^n = \{x_1, \dots, x_n\}$ be a random sample of size n from an unknown distribution $F(x)$ with $x \in \mathbb{R}^p$. Let $\hat{F}$ be the empirical distribution, putting probability 1/n

on each of the observed values x_i for $i = 1, \ldots, n$. Define the clustering $\psi(x)$ as a mapping $\psi : \mathbb{R}^p \to \{1, \ldots, k\}$ where k is the given number of clusters.

The number of clusters is estimated as follows:

1. Generate B bootstrap sample pairs $(X_b^{n*}, \tilde{X}_b^{n*})$ $b = 1, \ldots, B$ with each sample consisting of n observations generated from $\hat{F}$.

2. Construct $\psi_{X_b^{n*},k}$ and $\psi_{\tilde{X}_b^{n*},k}$, based on X_b^{n*} and $\tilde{X}_b^{n*}$, $b = 1, \ldots, B$, respectively.

3. For each pair $\psi_{X_b^{n*},k}$ and $\psi_{\tilde{X}_b^{n*},k}$ calculate the empirical clustering distance:

$$d_{\hat{F}}(\psi_{X_b^{n*},k}, \psi_{\tilde{X}_b^{n*},k}) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} |I\{\psi_{X^{n*},k}(x_i) = \psi_{X^{n*},k}(x_j)\} - I\{\psi_{\tilde{X}^{n*},k}(x_i) = \psi_{\tilde{X}^{n*},k}(x_j)\}|$$

4. The cluster instability $s(\psi, k, n)$ can be estimated by

$$\hat{s}_B(\psi, k, n) = \frac{1}{B} \sum_{b=1}^{B} d_{\hat{F}}(\psi_{X_b^{n*},k}, \psi_{\tilde{X}_b^{n*},k})$$

5. The optimal number of clusters is estimated by $\hat{k} = \hat{k}(n) = \underset{2 \le k \le K}{\arg\min}\, \hat{s}_B(\psi, k, n)$.

1.2 Biclustering

1.2.1 An Overview of Biclustering Approaches

Biclustering partitions both features and observations into clusters, accounting for the interaction between both rows and columns of a data set. Specifically, biclustering aims to identify a submatrix, U, within the original data set, X, such that the entries of U are more similar to each other than the remaining entries in X. Biclustering can be particularly useful in high dimensional data sets where clusters may differ based on only a subset of features. Cluster methods using all features to distinguish observations may fail to detect this data

structure. Biclustering may be especially relevant if one wishes to identify specific features which distinguish cluster membership.

Several strategies have been proposed to tackle the problem of identifying biclusters. One general strategy involves the use of mixture models to identify biclusters within the data set. Both the Plaid algorithm (Lazzeroni and Owen, 2002) and the LAS algorithm (Shabalin et al., 2009) use this strategy to identify biclusters. Another strategy uses singular value decomposition (SVD) to find signals within the data set which can be represented as a biclusters. The SSVD method (Lee et al., 2010) uses a penalized version of SVD to identify biclusters and the HSSVD method (Chen et al., 2013) expands upon this method by capturing the variance structure of a data set in addition to the mean structure. The sparse biclustering method of Tan and Witten (2014) frames biclustering as a penalized maximum likelihood estimation problem and shows how under certain conditions this is equivalent to computing the SVD of the data matrix. While not specifically a biclustering method, the sparse clustering approach of Witten and Tibshirani (2010) identifies clusters by maximizing a weighted version of the between cluster sum of squares subject to a lasso-penalty on the features. In the last part of the dissertation we expand upon the sparse clustering method of Witten and Tibshirani (2010) to identify biclusters. In this section we briefly describe the above methods, which will be compared to our proposed method.

1.2.1.1 Plaid

The Plaid method of Lazzeroni and Owen (2002) uses two-way ANOVA models to identify biclusters such that features may belong to one cluster, more than one cluster, or no clusters at all. Suppose we can represent the data set as a $p \times n$ matrix X. Let $i = 1, \ldots, p$ represent the features and $j = 1 \ldots, n$ represent the observations. Given K clusters the data is modeled as a sum of layers:

$$X_{ij} = \sum_{k=0}^{K} \theta_{ijk} \rho_{ik} \kappa_{jk} \tag{1.13}$$

where $k = 0$ represents the background layer, $\rho_{ik} = 1$ if feature i is in the k'th biclusters and $\rho_{ik} = 0$ otherwise, and $\kappa_{jk} = 1$ if observation j is in the k'th bicluster and $\kappa_{jk} = 0$ otherwise. θ_{ijk} is the layer effect for cluster k and can be defined as: $\theta_{ijk} = \mu_k,\ \theta_{ijk} = \mu_k + \alpha_{ik},\ \theta_{ijk} = \mu_k + \beta_{jk},\ or\ \theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}$. The shared response for all features in a layer by all observations in the layer is given by μ_k, α_{ik} is the specific effect of layer k on feature i, and β_{jk} is the specific effect of layer k on observation j. If α_{ik} is used it is subject to the constraint $\sum_i \rho_{ik}\alpha_{ik} = 0$. To estimate the above parameters, an iterative procedure is used to minimize the sum of squared errors.

To determine the number of clusters in the data set they compare the importance of cluster k, $\sigma_k^2 = \sum_{i=1}^{n}\sum_{j=1}^{p}\rho_{ik}\kappa_{jk}\theta_{ijk}^2$, to the importance of cluster k, found from the algorithm applied to a random permutation, r, of the residual matrix after applying k-1 layers, $\tilde{\sigma}_k^{2,r}$. For $r = 1, \ldots, R$ permutations a new layer k is added to the model if $\sigma_k^2 > \max_{1 \le r \le R}(\tilde{\sigma}_k^{2,r})$ and no layer is added otherwise.

1.2.1.2 Large Average Submatrix (LAS)

The LAS method of Shabalin et al. (2009) finds biclusters by identifying submatrices whose average is significantly larger than would be expected from a Gaussian random matrix. The Bonferroni-based significance score penalizes larger matrices in search of a submatrix with highest mean. Let the observed data be expressed as a matrix X with $i = 1, \ldots, m$ rows and $j = 1, \ldots, n$ columns. Given K clusters the data is represented as:

$$X_{ij} = \sum_{k=1}^{K} \alpha_k I(i \in A_k, j \in B_k) + \epsilon_{ij} \text{ if } K \neq 0$$
$$X_{ij} \sim N(0,1) \text{ otherwise} \tag{1.14}$$

where A_k and B_k are the row and column sets of the kth submatrix, respectively; α_k is the level of the kth submatrix; and ϵ_{ij} are i.i.d. $N(0,1)$ variables. To compare the submatrix average, τ, to the expected average of an equal sized submatrix from a Gaussian random

matrix, the following score value is calculated:

$$S(U) = -\log\left[\binom{m}{k}\binom{n}{l}\Phi(-\tau\sqrt{kl})\right] \tag{1.15}$$

The LAS algorithm uses a greedy search procedure to find the first submatrix of X that approximately maximizes the score function. For subsequent biclusters the same algorithm is conducted on the residual matrix, which is defined as the matrix computed by subtracting the average of the submatrix from each of its elements in X. The algorithm terminates when the score value drops below a certain value or when the maximum number of user-defined biclusters are produced.

1.2.1.3 Sparse Singular Value Decomposition (SSVD)

Instead of using a model based approach, the SSVD method of Lee et al. (2010) uses a penalized SVD approach to identify sparse left- and right- singular vectors. The non-zero entries of these singular vectors correspond to the objects and features that form the bicluster. Given a data set X with n rows (observations) and p columns (features) the SVD of X can be written as:

$$X = UDV^T = \sum_{k=1}^{r} s_k u_k v_k^T \tag{1.16}$$

Where the rank of X is r; $U = (u_1, \ldots, u_r)$ is a matrix of orthonormal left singular vectors; $V = (v_1, \ldots, v_r)$ is a matrix of orthonormal right singular vectors; and $D = diag(s_1, ..., s_r)$ is a diagonal matrix with positive singular values $s_1 \geq ... \geq s_r$ on the diagonal. $s_1 u_1 v_1^T$ is the closest rank-one approximation to X under the Frobenius norm.

To find sparse values for the singular vectors u, and v, the SSVD algorithm minimizes the following penalized sum of squares with respect to s, u, and v using the adaptive lasso

penalties suggested by Zou (2006).

$$||X - suv^T||_F^2 + \lambda_u s \sum_{i=1}^{n} w_{i,1}|u_i| + \lambda_v s \sum_{j=1}^{d} w_{2,j}|v_j| \tag{1.17}$$

For a description of the weights, $w'_{1,i}s$ and $w'_{2,j}s$, see Witten and Tibshirani (2010). Each layer represents a bicluster. By identifying the non-zero entries of u_k and v_k for a given layer k, one can identify the observations and features associated with a given bicluster, k.

1.2.1.4 Heterogeneous SSVD (HSSVD) method

The HSSVD method of Chen et al. (2013) builds upon the SSVD framework but allows for the identification of variance biclusters in the presence of heterogeneity of variance across subgroups. Variance biclusters are defined as biclusters which have a different pattern of variance within the bicluster as compared to the entries outside of the bicluster. The method defines biclusters as subsets of the data which share the same distinct mean and variance. Elements in the background layer are assumed to have the same mean and variance. The HSSVD approach uses a random effect model to express the observed $n \times p$ matrix X as

$$\boldsymbol{X} = \boldsymbol{\Xi} + \rho^2 \boldsymbol{\Sigma} \times \boldsymbol{\Phi} + b\boldsymbol{J} \tag{1.18}$$

$\boldsymbol{\Xi}_{n\times p} = (\xi_{ij})$ is a matrix representing the signal. $\boldsymbol{\Phi}_{n\times p} = (\phi_{ij})$ is a background noise matrix with i.i.d. entries with mean 0 and variance 1. The $\boldsymbol{\Sigma}_{n\times p} = (\sigma_{ij})$ term is a matrix accounting for the heterogeneity in variance signal. ρ is a finite positive number serving as a common scale factor. $\boldsymbol{J}_{n\times p}$ is a unit matrix. b is a finite number serving as a common location factor. In order to impose a sparsity constraint the majority of the ξ_{ij} values are assumed to be 0, the majority of the σ_{ij} values are assumed to be 1, and the mean structure $\boldsymbol{\Xi}$ and variance structure $\boldsymbol{\Sigma}$ are assumed to be low rank. Mean biclusters are first detected after scaling the data. Then variance biclusters are identified after subtracting out the mean biclusters.

1.2.1.5 Sparse Biclustering

Tan and Witten (2014) develop an penalized maximum likelihood based approach to identify biclusters in the data. Their method hinges on the assumption that that all data matrix elements are independent and normally distributed and many biclusters have mean terms that are approximately zero. Using these assumptions they maximize a penalized log-likelihood of the data to identify biclusters.

Let the matrix entries be denoted X_{ij} with $i = 1, \ldots, n$ observations and $j = 1, \ldots, p$ features. Assume that the n observations belong to K unknown and non-overlapping classes $C_1, \ldots, C_k$ and the p features belong to R unknown and non-overlapping classes, $D_1, \ldots, D_R$. Then $X_{ij} \sim N(\mu_{kr}, \sigma^2)$ for $i \in C_k, j \in D_r$. For set number of K and R, $C_k,\ D_r$, and μ_{kr} can be estimated by maximizing the ℓ_1 penalized log likelihood which is equivalent to:

$$\underset{C_1,\ldots,C_K,D_1,\ldots,D_R,\mu\in\mathbb{R}^{K\times R}}{minimize} \left\{ \frac{1}{2} \sum_{k=1}^{K} \sum_{r=1}^{R} \sum_{i\in C_k} \sum_{j\in D_r} (X_{ij} - \mu_{kr})^2 + \lambda \sum_{k=1}^{K} \sum_{r=1}^{R} |\mu_{kr}| \right\}$$

Where λ is a nonnegative tuning parameter.

The number of row clusters K and column clusters R is chosen based on minimizing prediction error using the bicluster means to impute missing values. Specifically, a random subset of the data is removed from the data and biclustering is performed on the resulting data matrix. The missing data points are imputed based on the estimated bicluster mean. K and R are chosen to minimize the squared error loss between the true data points and imputed data points.

The tuning parameter λ is chosen using a BIC-based approach. Specifically for K and R known or K and R selected using $\lambda = 0$, sparse clustering is performed on the data for various choices of λ. For each value of λ a $(np) \times (q + 1)$ design matrix is constructed with first column equal to 1. The remaining columns contain 1's or 0's indicating whether a given data entry is part of a nonzero-mean bicluster in the sparse biclustering output. A

least squares regression model is used to predict the data based on the design matrix and the BIC is computed:

$$BIC = np \times \log(RSS) + np \log(q)$$

where RSS is the residual sum of squares. λ is chosen to minimize the BIC.

1.2.1.6 Sparse clustering

In a data set with many features, it might be expected that only a subset of the features are responsible for clustering. The sparse clustering method of Witten and Tibshirani (2010) capitalizes upon this assumption by using a lasso-type penalty to adaptively select features. This subset of features is then used to cluster observations. This method can be used for both k-means and hierarchical clustering. For a $n \times p$ data set $\boldsymbol{X}$ with n observations and p features the general sparse clustering is the solution to the problem

$$\underset{\boldsymbol{w};\boldsymbol{\Theta}\in D}{maximize}\{\Sigma_{j=1}^{p} w_j f_j(\boldsymbol{X_j}, \boldsymbol{\Theta})\} \text{ subject to } ||w||^2 \leq 1, ||w||_1 \leq s, w_j \geq 0 \; \forall j \qquad (1.19)$$

where where $f_j(\boldsymbol{X_j}, \boldsymbol{\Theta})$ is some function that involves only the jth feature of the data; $\boldsymbol{\Theta}$ is a parameter, often the clustering indices, restricted to lie in a set D; and w_j is a weight corresponding to the jth feature. When Θ is held fixed the solution to the weights can be solved by soft-thresholding as follows:

$$\boldsymbol{w} = \frac{S(\boldsymbol{a}_+, \Delta)}{||S(\boldsymbol{a}_+, \Delta)||_2} \qquad (1.20)$$

Where a_+ denotes the positive part of a and $a_j = f_j(\boldsymbol{X_j}, \boldsymbol{\Theta})$. S is the soft-thresholding operator, $S(x, c) = sign(x)(|x| - c)_+$. $\Delta = 0$ if that leads to $||w||_1 \leq s$, otherwise $\Delta > 0$ is chosen such that $||w||_1 = s$ where s is a specified tuning parameter.

For application to k-means clustering the sparse clustering method maximizes a weighted version of the between cluster sum of squares. Given K clusters in the data set, the between

cluster sum of squares for feature j ($BCSS_j$) is defined to be

$$BCSS_j = \sum_{j=1}^{p}(\frac{1}{n}\sum_{i=1}^{n}\sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{K}\frac{1}{n_k}\sum_{i,i'\in C_k} d_{i,i',j}) \tag{1.21}$$

Where $d_{i,i',j} = (X_{ij} - X_{i'j})^2$; n_k is the number of observations in cluster k; and C_k is the set of indices of observations belonging to cluster k. The sparse K-means clustering criterion is simply the solution to equation 1.19 with $f_j(\boldsymbol{X_j}, \boldsymbol{\Theta}) = BCSS_j$.

Sparse hierarchical clustering is implemented on a weighted dissimilarity matrix. Let $\boldsymbol{U}$ be the overall dissimilarity matrix $\{\sum_j d_{i,i',j}\}_{i,i'}$. Then sparse hierarchical clustering simply reduces to equation 1.19 with $f_j(\boldsymbol{X_j}, \boldsymbol{\Theta}) = \sum_{i,i'} d_{i,i',j}U_{i,i'}$ and the additional constraint that $\sum_{i,i'} U_{i,i'}^2 \leq 1$.

To determine the optimal tuning parameter, s, a permutation approach related to the gap statistic of Tibshirani et al. (2001) is applied. The number of clusters must be pre-specified before the algorithm is initiated.

In the second part of the dissertation we build upon the sparse clustering method of Witten and Tibshirani (2010) and incorporate an iterative cluster significance testing method in order to determine the optimal number of clusters for the data set. We also show how the method can be applied to identify mean biclusters and compare our proposed algorithm to the methods listed above.

1.3 Association between Intermediate Phenotypes and Genetic Markers in a Case-Control Study

Genome-wide association studies are expensive to conduct in terms of both time and resources so it is often in the researcher's best interest to maximize the investment. To achieve this goal, a case-control design is often used because it leads to the greatest power to detect associations between the disease of interest and genetic factors, especially if the disease is rare. Even though this is a stratified sampling design, Prentice and Pyke (1979)

have shown that the maximum likelihood estimates of the odds ratios are generally unbiased, leading to a substantial decrease in cost compared to a prospective cohort study. To further maximize the investment researchers may be interested in analyzing the association between the genetic factors and phenotypes which were collected secondary to the primary disease outcome. Since the data is not a simple random sample, and sampling was not based on the secondary phenotype, the estimates of the association between genetic factors and secondary phenotype may be biased. Regardless, some studies present the associations without controlling for the case-control sampling scheme, or they employ less than optimal methods such as analyzing the association only in cases or only in controls, or including disease status as a covariate in a regression model. As shown in Monsees et al. (2009) and Lin and Zeng (2009) these methods can lead to biases in estimating the association between the genetic factors and the secondary phenotypes.

1.3.1 Overview of Approaches

Several recent approaches have been proposed to control for the biases induced by the case-control sampling design. Richardson et al. (2007) and Monsees et al. (2009) propose using a inverse-probability-of-sampling-weighted (IPW) regression for testing the association between the secondary phenotype and genetic factors. Since this approach down-weights cases to better reflect their prevalence in the general population a sandwich-type estimator is needed to find the variance of the resulting coefficients. Lin and Zeng (2009) instead use a likelihood based approach with distributional constraints on the phenotypes to model the association between the genetic factors and secondary phenotypes. He et al. (2012) take a more comprehensive approach and use Gaussian copulas to model both the primary and secondary phenotypes within a case-control sampling scheme, allowing for the two phenotypes to be correlated. Wang and Shete (2009) propose a method for binary phenotypes which iteratively solves non-linear equations involving disease prevalence. They

also propose a bootstrap method for estimating the empirical confidence intervals for the estimated odds ratios.

1.3.1.1 Inverse-probability-of-sampling-weighted regression

The IPW method (Monsees et al., 2009 and Richardson et al., 2007) uses a simple weighting scheme to account for the over-representation of disease cases in the sample. Let n_0 be the number of controls in the sample and n_1 be the number of cases. Monsees et al. (2009) propose weighting the cases by $w_1 = 1$ and the controls by $w_0 = n_0/n_1$. Weighted regression can then be performed by utilizing the sandwich estimator for variance estimation.

Wang and Shete (2009) slightly modify the weights to account for a retrospective case-control study design. They propose using $w_1 = 1$ for cases and for controls, $w_0 = (n_1)(1-p)/(n_0 p)$, where p is the disease prevalence in the population. In the last chapter of the dissertation we will compare a similar weighted method to the proposed method.

1.3.1.2 Maximum likelihood estimation

The method proposed by Lin and Zeng (2009) uses a maximum likelihood estimation approach to assess the association between a secondary phenotype and genetic factors collected within a case-control study. Specifically, for disease status D (1= disease, 0=no disease), secondary phenotype Y, and the genotype score for a SNP of interest, X, the relationship between Y and X is modeled as

$$P(Y = y|X) = N(\beta_0 + \beta_1 X, \sigma^2)$$

if Y is continuous. If Y is dichotomous the relationship is modeled as:

$$P(Y = 1|X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

Y and X are related to D through the equation

$$P(D=1|X,Y)=\frac{e^{\gamma_0+\gamma_1 X+\gamma_2 Y}}{1+e^{\gamma_0+\gamma_1 X+\gamma_2 Y}}.$$

Because the sampling was conditional on disease status the likelihood takes the form

$$\prod_{i=1}^{n} P(Y_i,X_i|D_i)=\prod_{i=1}^{n}\left\{\frac{P(D_i=1|X_i,Y_i)P(Y_i|X_i)P(X_i)}{P(D_i=1)}\right\}^{D_i}\times\left\{\frac{P(D_i=0|X_i,Y_i)P(Y_i|X_i)P(X_i)}{P(D_i=0)}\right\}^{1-D_i}$$

where $P(D_i=1)=\sum_y\sum_x P(D_i=1|x,y)P(y|x)P(x),\ P(D_i=0)=1-P(D_i=1)$, and $P(D_i=0|X_i,Y_i)=1-P(D_i=1|X_i,Y_i)$

This likelihood can be maximized using the Newton-Raphson algorithm and likelihood-based statistics can be used to make inference about β_1. The authors also develop additional approaches to handle the scenarios when the disease is rare, the disease rate is known, or additional environmental covariates need to be included in the model.

1.3.1.3 Gaussian copula approach

He et al. (2012) use a Gaussian copula-based approach to model the association between secondary phenotype and genetic factors while also allowing there to be dependence between disease status and secondary phenotype. They start with a general framework for Gaussian copulas modeling the joint density between multiple outcome variables assumed to be from an exponential family. For outcome y_j let η_j represent the canonical parameter, ϕ_j represent the dispersion parameter, and μ_j be the expected value. Let Φ be the standard univariate normal distribution function and Γ be a $m\times m$ correlation matrix. Let $F_j(y_j)$ denote the cumulative distribution function of y_j and $q=(\Phi^{-1}(F_{m_1+1}(y_{m_1+1}),\ldots,F_m(y_m)))$. Then for a binary case indicator y_1 and multiple secondary phenotypes $y_2,\ldots,y_m$ the Gaussian

copula is given by:

$$P(y_1,\ldots,y_m) = \prod_{j=2}^{m} f(y_j,\eta_j,\phi_j)$$
$$\times \left[1 - \int_{-\infty}^{\Phi^{-1}(1-\mu_1)} \frac{1}{\sqrt{2\pi|\Gamma|}} \exp\left\{\frac{-1}{2}(z,q)\Gamma^{-1}(z,q)^T + \frac{1}{2}qq^T\right\} dz\right]^{I\{y_1=0\}}$$
$$\times \left[\int_{-\infty}^{\Phi^{-1}(1-\mu_1)} \frac{1}{\sqrt{2\pi|\Gamma|}} \exp\left\{\frac{-1}{2}(z,q)\Gamma^{-1}(z,q)^T + \frac{1}{2}qq^T\right\} dz\right]^{I\{y_1=1\}}$$

Let g=(0,1,2) denote the genotype at the test SNP for an individual. The retrospective likelihood for the individual is given by:

$$P(y_2,\ldots,y_m,g|y_1) = \frac{P(y_1,\ldots,y_m|g)P(g)}{P(y_1)} = \frac{P(y_1,\ldots,y_m|g)P(g)}{\sum_{g=0}^{2} P(y_1|g)P(g)}. \tag{1.22}$$

The genotype, g, can be related with the marginal mean model for each of the phenotypes through the use of a link function $h(\mu_j) = \beta_{0,j} + \beta_{1,j} \times g$, where specification of the link function depends on the distribution of the phenotype. For disease status, the marginal mean can be modeled by $\log[\mu_1/(1-\mu_1)] = \beta_{1,0} + \beta_{1,1} \times g$ where $\mu_1 = P(y_1 = 1|g)$. For the jth secondary phenotype, $h(\mu_j) = \beta_{j,0} + \beta_{j,1} \times g$, the marginal mean model will be determined by its distribution. The overall likelihood is then simply the product of the likelihoods across all individuals.

Next they establish the maximum likelihood estimation for the parameters

$$\theta = (\{\beta_{j,0}, \beta_{j,1}, \phi_j\, \gamma_j\}_{j=1}^{m}, p)$$

where γ_j represents the correlation between y_1 and y_j. They first fix the disease prevalence and update the intercept parameter for the primary phenotype by $\beta_{1,0} = \beta^*_{1,0} + \log[K/(1-K)]$, where $\beta^*_{1,0}$ is the intercept estimate obtained from a logistic regression on the disease status variable with the SNP genotype included as a covariate. The maximum likelihood estimate of the parameters can then be found using a Gauss-Newton type algorithm

using step halving. The variances of the parameters are approximated by numerical Fisher information. A Wald test can then be used to test the association between the SNP and secondary phenotype of interest.

1.3.1.4 Non-linear equation modeling for binary phenotype

When the secondary phenotype is binary Wang and Shete (2009) propose iteratively solving a system of nonlinear equations to estimate the odds ratio corrected for the case-control sampling scheme. They also propose a bootstrapping method to estimate the empirical confidence intervals for the corrected odds ratio.

Let D denote the disease status, T denote the secondary phenotype, and X denote the SNP variable from a dominant or recessive genetic model, X=0,1. The relationships between D, T, and X can be modeled as follows:

$$P(T = k|X = i) = p_{k|i} = \frac{\exp(\alpha_0 + \alpha_1 i)}{1 + \exp(\alpha_0 + \alpha_1 i)}$$

$$P(D = j|T = k, X = i) = p_{j|ki} = \frac{\exp(\beta_0 + \beta_1 i + \beta_2 k)}{1 + \exp(\beta_0 + \beta_1 i + \beta_2 k)}$$

for i, j, k=0, 1. The odds ratio for the association between the SNP and the secondary phenotype is given by $\exp(\alpha_1)$ and can be estimated by OR=$n_{11}n_{00}/n_{10}n_{01}$. Let E_{ki} represent the expected value of n_{ki} conditional on n_1 and n_0. Let p_i i=0,1 represents the probabilities associated to the genotypic frequencies of the SNP of interest. The probability, q_1, is the prevalence of the disease (f_D) in the general population and q_0 is calculated as $1 - f_D$. The odds ratio can then be written using the expected number of individuals:

$$\begin{aligned} OR &= \frac{E_{11}E_{00}}{E_{10}E_{01}} \\ &= \frac{((1-f_D)N_1^2 p_{1|11} + f_D N_0^2 p_{0|11})p_{1|1} \times ((1-f_D)N_1^2 p_{1|00} + f_D N_0^2 p_{0|00})p_{0|0}}{((1-f_D)N_1^2 p_{1|10} + f_D N_0^2 p_{0|10})p_{1|0} \times ((1-f_D)N_1^2 p_{1|01} + f_D N_0^2 p_{0|01})p_{0|1}} \end{aligned} \tag{1.23}$$

The estimated prevalence of the primary disease is given by:

$$f_D = P(Y = 1) = \sum_i \sum_k p_{1|ki} p_{k|i} p_i \tag{1.24}$$

The estimated prevalence of the secondary phenotype is given by:

$$f_T = P(T = 1) = \sum_i p_{1|i} p_i \tag{1.25}$$

The solution to non-linear system of equations 1.23-1.25 gives a corrected OR for the SNP associated with the secondary phenotype under an a dominant or a recessive genetic model.

When an additive genetic model is assumed the proposed corrected odds ratio is the average of the odds ratio of X=1 versus X=0 (denote this odds ratio as $\widetilde{OR_1}$), and the odds ratio of X=2 versus X=0 (denote this odds ratio as $\widetilde{OR_1'}$). $\widetilde{OR_1}$ can be estimated by solving the system of equations 1.23-1.25, For $\widetilde{OR_1'}$ the equation for the odds ratio takes a slightly different form:

$$OR = \exp\left(\frac{1}{2}\log\left(\frac{E_{12}E_{00}}{E_{02}E_{10}}\right)\right) \tag{1.26}$$
$$= \exp\left(\frac{1}{2}\log\left(\frac{((1-f_D)N_1^2 p_{1|12} + f_D N_0^2 p_{0|12})p_{1|2} \times ((1-f_D)N_1^2 p_{1|00} + f_D N_0^2 p_{0|00})p_{0|0}}{((1-f_D)N_1^2 p_{1|02} + f_D N_0^2 p_{0|02})p_{0|2} \times ((1-f_D)N_1^2 p_{1|10} + f_D N_0^2 p_{0|10})p_{1|0}}\right)\right)$$

$\widetilde{OR_1'}$ can be obtained by solving the system of nonlinear Equations 1.24-1.3.1.4.

The empirical confidence intervals are estimated using a bootstrapping approach. B samples are taken from the normal distribution with mean $\hat{\alpha}_1$ and variance $\hat{s}^2$ where $\hat{s}$ is the standard estimate of $\hat{\alpha}_1$. Denote the bootstrap samples as α_{1u}^* for $u = 1, \ldots, B$. The bootstrap $\widehat{OR}$ is then estimated as $\widehat{OR}_u^* = \exp(\alpha_{1u}^*)$, $u = 1, \ldots, B$. For each $\widehat{OR}_u^*$ the bootstrap corrected $\widetilde{OR}_u^*$ is computed by solving the previously defined system of equations. The $100(1-\gamma)\%$ confidence interval for $\widetilde{OR}$ is then given as $(\widetilde{OR}_{[B\gamma/2]}^*, \widetilde{OR}_{[B(1-\gamma/2)]}^*)$ where $\widetilde{OR}_{[u]}^*$ is the uth ordered bootstrap estimate.

CHAPTER 2: NON-PARAMETRIC CLUSTER SIGNIFICANCE TESTING WITH REFERENCE TO A UNIMODAL NULL DISTRIBUTION

2.1 Introduction

In an initial analysis of a data set, one often seeks to determine if there are any natural subtypes present. Clustering is a common unsupervised method where subgroups in the data are identified without specifying a response variable. This exploratory tool can be applied to any data set where similarities between observations can be measured. Several commonly used methods include hierarchical clustering, k-means clustering, and spectral and graph-based methods. Clustering has been especially useful for identifying patterns in complex high-dimensional data sets. After clusters have been identified, the next logical step is to to determine if the putative clusters truly represent distinct subgroups rather than noise.

Several methods have been developed that assess the significance of identified clusters. These methods generally test the null hypotheses that the data cannot be partitioned into clusters by comparing a measure of cluster quality from the observed data to what would be expected from a null distribution. The null distribution is generated under certain assumptions such as a specific parametric distribution or assumptions about cluster shape.

The SigClust method of Liu et al. (2008) takes a parametric approach and defines a cluster as a single Gaussian distribution. To test the null hypothesis that the data cannot be partitioned into clusters, the test statistic (the cluster index) from the observed data is compared to the cluster index from a single, multivariate, Gaussian distribution. For high dimensional low sample size (HDLSS) covariance estimation, they use a combination

of invariance principles and a factor analysis model. Since SigClust places a Gaussian constraint on the clusters it is not suited for significance testing in non-normal settings.

The significance test of Maitra et al. (2012) avoids defining a parametric distribution, but assumes that the clusters can be transformed to be spherical and that they are compact and similar to each other. Their bootstrapping method requires the estimation of the data covariance matrix and is not optimized for the HDLSS setting.

Kapp and Tibshirani (2006) also propose a non-parametric method for evaluating the significance of putative clusters. They use a validation measure called the in-group proportion (IGP), which is the proportion of observations in a cluster whose nearest neighbors are also in the same cluster. They use the IGP to assess cluster reproducibility, the ability to find the same clusters in a similar data set. Cluster reproducibility is assessed by comparing the IGP from the observed data to the IGP generated from a null distribution where the features have been permuted within each observation. Generation of the null distribution can be computationally expensive for large data sets and ignores the possible correlation between features.

Some authors have proposed evaluating cluster significance by testing the data for unimodality. Ahmed and Walther (2012) employ a global test by reducing the data to a principal curve. Then Silverman's bandwidth test (Silverman, 1981) is used to test for multimodality in the univariate curve. Although this is a promising approach, important data structures may be lost when the data is reduced to a unidimensional summary.

Closely related to the problem of cluster validation is the challenge of determining the number of clusters in a data set. Conventional methods choose the number of clusters that maximizes the value of a measure of cluster strength, such as the average silhouette width (Rousseeuw, 1987) or the Calinski-Harabasz (CH) Index (Calinski and Harabasz, 1974). Tibshirani et al. (2001) use the within cluster dispersion (WCD) to measure cluster strength. The number of clusters is chosen to maximize the WCD between the data and a null distribution generated from a uniform distribution aligned with the principal components

of the data. Tibshirani and Walther (2005) treat determining the number of clusters as a model selection problem where the number of clusters is chosen to maximize prediction strength. Test and training sets are used to assess the predictive value of identified clusters. The largest value of K that gives a prediction strength greater than a chosen threshold is estimated to be the number of clusters in the data set. Fang and Wang (2012) choose the number of clusters which minimizes cluster instability. Bootstrap samples are used to assess the stability of cluster assignments. Each of these methods has its limitations, and as Hennig (2014) noted, different criteria for evaluating cluster strength may lead to different values of K. Also, even incorrect clustering assignments may be stable and/or have high predictive value.

In this paper, we develop a novel nonparametric cluster significance test with application to HDLSS data. Our method, the unimodal nonparametric cluster (UNPaC) test, defines a cluster as a subset of the data coming from a unimodal distribution. We test the null hypothesis that all the data belongs to a single cluster by comparing the cluster index (CI) from the data to the CI from an appropriate unimodal null distribution. This method can also be extended to estimate the number of clusters present. The method is versatile for a wide variety of settings, including situations were normality assumptions do not hold.

The article is organized as follows. We first present the UNPaC algorithm and its theoretical properties. Then we extend the method to estimate the number of clusters. Next, simulation studies are conducted to compare UNPaC to existing methods for testing the significance of clusters and estimating the number of clusters in a data set. We present applications of UNPaC to real data in sections 2.6 and 2.7. We finish with a discussion of our method and results.

2.2 The UNPaC Test Method

Let $X_{n\times p}$ represent the observed data with n observations and p covariates or features. For ease of explanation, assume X is centered such that each feature has a mean of zero. We

define a cluster as a unimodal distribution where a mode is a single point where the density is maximized. Our null and alternative hypotheses can be represented as:

H_0: The data comes from a unimodal distribution.

H_a: The data do not come from a unimodal distribution.

To perform this hypothesis test, a unimodal reference data set X^0 is generated from a distribution such that it is as close to the observed data distribution as possible under the restriction of unimodality. To do this, Gaussian kernel density estimation (KDE) is used to estimate the density of each feature j: .

$$\hat{f}_j(t; h_j) = (nh_j)^{-1} \sum_{i=1}^{n} K(h_j^{-1}(t - X_{ij})) \tag{2.27}$$

where h_j is the bandwidth; $K(\cdot)$ is the Gaussian kernel function; and $X_{1j}, \ldots, X_{nj}$ are the data entries for feature j. Silverman (1981) showed that a critical bandwidth, h_{kj}, can be determined such that

$$h_{kj} = \inf\{h_j : \hat{f}_j(\cdot; h_j) \text{ has at most } k \text{ modes}\}. \tag{2.28}$$

$\hat{f}_j(t; h_{1j})$ then represents an estimation of the density under the unimodal constraints. Efron and Tibshirani (1993) show that bootstrap samples can be drawn from a rescaled version of $\hat{f}_j(t; h_{1j})$ as follows:

$$X_{ij}^0 = (1 + h_{1j}^2/\sigma_j^2)^{-1/2}(X_{I(ij)} + h_{1j}\epsilon_i) \tag{2.29}$$

where $\epsilon_i \sim N(0, 1)$ (We use the notation $\mathrm{N}(a, b)$ to represent a normal distribution with mean a and standard deviation b.), σ^2 is the sample variance for feature j, and $X_{I(ij)}$ are sampled uniformly, with replacement, from the observed data for feature j.

Figure 2.1 illustrates how this reference distribution closely approximates the observed data. In this example, the observed data is composed of 10,000 observations and 2 features.

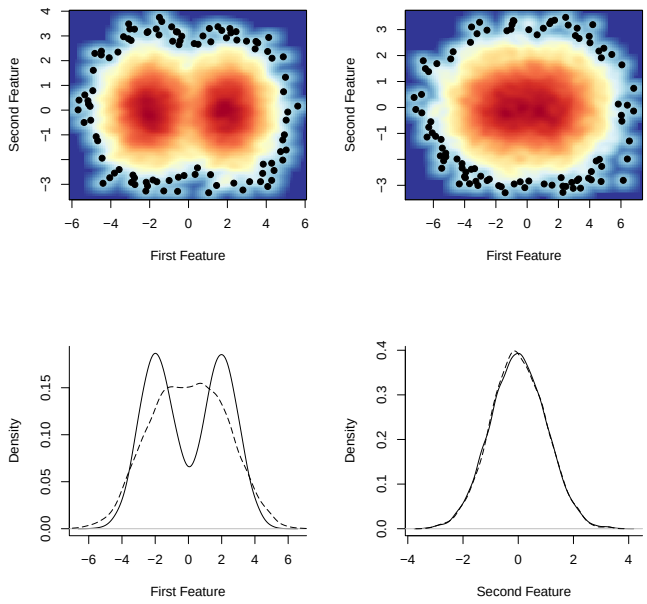

Figure 2.1: *Illustration of the proposed reference distribution.* Two clusters are present in the data and they differ only with respect to the mean of the first feature. The first row gives the bivariate density for the observed data (left) and the reference distribution (right) The second row gives the univariate density for the first feature (left) and second feature (right). The solid lines are from the observed data and the dashed lines are from the reference distribution. Note that the null distribution is bivariate unimodal and also unimodal for each feature, separately. The null data also closely approximates the density of the second feature.

The first cluster consists of 5,000 observations with a standard normal distribution. The second cluster consists of the remaining 5,000 observations and is distributed as $N((4,0)^T, \boldsymbol{I})$ with $\boldsymbol{I}$ indicating an independent covariance structure with feature variance equal to one. The features have been centered in the figure.

To ensure that the first and second moments from the reference distribution approximate the moments from the observed data, we first scale *X* such that the features have variance equal to one before generating X^0 using the methodology described above. The original covariance structure in X can be preserved in the null data by multiplying X^0 by the Cholesky root of the estimated covariance of X.

If the number of features, p, is less than the number of observations, n, a sample covariance estimate can easily be calculated. When $p > n$, we use the graphical lasso (Friedman et al., 2008) to produce a sparse approximation of the covariance structure. The

authors do not give guidelines for choosing ρ, a tuning parameter responsible for controlling the amount of ℓ_1 shrinkage, but we have found that $\rho = 0.02$ produces good results for most applications.

To reduce the computation time for high dimensional data sets, we propose to first use a dimension reduction technique similar to the method proposed by Bair and Tibshirani (2004). Specifically, we chose a subset of features which have a strong association with the cluster assignment. Unless otherwise noted, the features are chosen such that the p-value for testing the null hypothesis that the mean value of the feature is the same in both clusters (based on a t-test) is less than $\alpha = 0.10$.

To measure the strength of clusters, we use the two-means cluster index (CI), which is the ratio of the within cluster sum of squares and the total sum of squares.

$$\text{CI} = \frac{\sum_{k=1}^{2} \sum_{j \in C_k} ||x_j - \bar{x}^{(k)}||^2}{\sum_{j=1}^{n} ||x_j - \bar{x}||^2} \tag{2.30}$$

Here $\bar{x}^{(k)}$ is the mean of cluster k for $k \in \{1, 2\}$, C_k is the sample index for the kth cluster, and $\bar{x}$ is the overall mean. Smaller values of the CI indicate that more of the overall variation in the data is explained by the clustering.

The UNPaC test procedure can be summarized as follows:

1. Identify putative clusters in X. This can be done by applying k-means clustering with *k*=2 after scaling the features to have variance equal to one. (See supplementary materials for a description of how this method can be applied to hierarchical clustering.) The scaled data will be denoted by X^s.

2. Optional dimension reduction for high dimensional data sets. First select features of X which are strongly associated with the putative clusters. Then replace X with the reduced data set, X^*, and rerun Step 1.

3. Calculate the two-means clustering index for X (or X^*), CI_{data}.

4. Estimate the covariance of X (or X^*). When $p > n$ we use the graphical lasso (Friedman et al., 2008) with sparsity parameter $\rho = 0.02$.

5. Generate the multivariate unimodal reference data X^0.

   (a) For each feature j in X^s (or X^{*s}), find the smallest bandwidth estimator such that the Gaussian KDE for that feature has one mode.

   (b) Sample data from the unimodal density according to (3.37) to generate X_j^0.

   (c) Multiply $\{X_1^0, \ldots, X_p^0\}$ by the Cholesky root of the estimated covariance matrix to generate X^0.

   (d) Cluster X^0 using the same clustering algorithm that was used in step 1.

   (e) Calculate the cluster index for X^0.

6. Repeat step 5 N_{sim} times. For all examples in this paper, we let $N_{\text{sim}} = 1000$.

7. Using the CIs from the simulated data, calculate the p-value as follows: $\sum_{b=1}^{N_{\text{sim}}} \{CI_b > CI_{\text{data}}\}/N_{\text{sim}}$. Alternatively, a normal approximation can be calculated by comparing $Z = (CI_{\text{data}} - \mu_{\text{CI}})/\sigma_{\text{CI}}$ to the standard normal distribution where μ_{CI} and σ_{CI} represent the mean and standard deviation of the null CIs, respectively.

We can conclude that a test is statistically significant if the p-value is less than a pre-specified level α, which is typically chosen to be 0.05.

2.3 Theoretical Properties

We now establish several important theoretical properties for the reference distribution and statistical test. Tibshirani et al. (2001) show that for the gap statistic, a statistic also based on the within cluster sum of squares, there is no least favorable multivariate unimodal reference distribution when the number of features is greater than 1. Also, the maximum likelihood estimator of a unimodal density does not exist (see, e.g. (Birge, 1997) and

(Balabdaoui et al., 2009)). Thus we we will not be able to choose a reference distribution that is optimal in all situations. However, we can show that the proposed reference distribution has optimal characteristics including convergence in first and second moments and multivariate unimodality. We also show that the test has optimal sensitivity and specificity, asymptotically.

When $n > p$, it is trivial to show convergence of first and second moments of the reference distribution to the moments of the observed data. When $p > n$, convergence of the moments depends on the method used to estimate the covariance (in this paper the graphical lasso is used) and is beyond the scope of this paper.

Recall that h_{1j} can be chosen such that $\hat{f}(\cdot, h_{1j})$ has at most one mode (Silverman, 1981). Thus, before multiplying by the Cholesky root, each feature in the reference distribution is unimodal. We show that the final joint reference distribution is multivariate unimodal using definition 2.5 from Sager (1978):

Let $\boldsymbol{t} = (t_1, .., t_p)$ and $d(\boldsymbol{t}, \boldsymbol{y}) = |\boldsymbol{t} - \boldsymbol{y}|$. A point $\boldsymbol{\theta}$ is the multivariate mode of F if for each $\epsilon > 0$, $\exists\ \delta > 0$ s.t. $d(\boldsymbol{t}, \boldsymbol{\theta}) > \epsilon$ implies $f(\boldsymbol{t})+ \delta < f(\boldsymbol{\theta})$.

Theorem 1. If each feature is independent and unimodal with the feature mode given by $\max(f_j(t_j)) = m_j$ and we have that the unique mode for $f(t_1, .., t_p)$ can be represented by $\max_{\boldsymbol{t}}\{f(t_1, .., t_p)\} =< m_1, ..., m_p >$, then after multiplying by the Cholesky root of the estimated covariance matrix, the resulting joint distribution, $h(y_1, ..., y_p)$, is multivariate unimodal.

We also establish the asymptotic convergence of the null CI to the data CI when the data is unclustered and divergence when the data is clustered. We assume that the data has a multivariate Gaussian distribution. As $n \to \infty$, the sample data approaches continuity, so the focus of the proof is on the theoretical cluster index (TCI), the cluster index assuming

that the data is continuous. The TCI is defined to be the theoretical within cluster sum of squares (TWSS) divided by the theoretical total sum of squares (TTSS). Let the feature space be partitioned into two non-overlapping subspaces S_1 and S_2 and let $\boldsymbol{\mu_k} = \int_{\boldsymbol{x} \in S_k} \boldsymbol{x} f(\boldsymbol{x}) d\boldsymbol{x}$. Using the same notation as in Huang et al. (2015), the theoretical sum of squares is given by:

$$TCI = \frac{TWSS}{TTSS} = \frac{TWSS_1 + TWSS_2}{TTSS} = \frac{\int_{\boldsymbol{x} \in S_1} ||\boldsymbol{x} - \boldsymbol{\mu_1}||^2 f(\boldsymbol{x}) d\boldsymbol{x} + \int_{\boldsymbol{x} \in S_2} ||\boldsymbol{x} - \boldsymbol{\mu_2}||^2 f(\boldsymbol{x}) d\boldsymbol{x}}{\int ||\boldsymbol{x}||^2 f(\boldsymbol{x}) d\boldsymbol{x}}$$

Theorem 2. Let $\boldsymbol{x} = (x_1, \ldots, x_p)$ be a p-dimensional random vector with multivariate normal distribution $\boldsymbol{x} \sim N(\boldsymbol{0}, \boldsymbol{D})$ where $\boldsymbol{D}$ is a known covariance matrix with diagonal entries $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$. Let TCI_{GAUSS} represent the theoretical cluster index of $\boldsymbol{x}$. For the choice of S_1 and S_2 which minimizes the TWSS, $TCI_{\text{GAUSS}} = 1 - \{2\lambda_1 / (\pi \sum_{j=1}^{p} \lambda_j)\}$. The theoretical cluster index for the null distribution, TCI_{null} approaches TCI_{GAUSS} as n $\to \infty$.

Theorem 3. Let $\boldsymbol{x} = (x_1, \ldots, x_p)$ be a p-dimensional random vector with the data distributed as $\eta f(x) + (1 - \eta) g(x)$. $f(x) \sim N(\boldsymbol{0}, \boldsymbol{D})$; $g(x) \sim N(\boldsymbol{\mu}, \boldsymbol{D})$; $\eta \in (0, 1)$ is the mixing proportion; $\boldsymbol{\mu} = (a, \ldots, a)^T$ with nonzero constant a; and $\boldsymbol{D}$ is a known diagonal matrix with elements $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$. Suppose η stays constant as $n \to \infty$ and the cluster algorithm is able to correctly classify the n_1 observations that arise from $f(x)$ into cluster 1 and the n_2 observations that arise from the $g(x)$ into cluster 2. Let TCI_{mix} represent the theoretical cluster index from the mixture distribution. Then $\lim_{n \to \infty} TCI_{\text{mix}} < \lim_{n \to \infty} TCI_{\text{null}}$.

We also establish asymptotic properties for the cluster index as $p \to \infty$.

Theorem 4. Let $X = \eta N(\boldsymbol{0}, \boldsymbol{D}) + (1 - \eta) N(\boldsymbol{\mu}, \boldsymbol{D})$ where X is an $n \times p$ matrix and $\eta \in (0, 1)$. Let n_1 be the number of observations from $N(\boldsymbol{0}, \boldsymbol{D})$ and n_2 be the number of observations from $N(\boldsymbol{\mu}, \boldsymbol{D})$. Define $\boldsymbol{\mu} = (a, ..., a)^T$ with a $\neq 0$. Let $\boldsymbol{D}$ be a known diagonal

covariance matrix with elements $\lambda_1 > ... > \lambda_p$. Assume $n_1 + n_2 = n \geq 3$ with n fixed, $\min(n_1, n_2) > 0$, $\Sigma_{j=1}^{p} \lambda_j = O(p^{\beta})$ with $0 \leq \beta < 1$, $\max_j(\lambda_j + \eta(1-\eta)\alpha^2) \leq M$ with $M > 0$ a fixed constant. Assume that a finite critical bandwidth h_{1j} can be chosen for each feature such that $\max_j(h_{1j}) < L$ with $L > 0$ a fixed constant. Then the corresponding p-value for the UNPaC test converges to 0 in probability as $p \to \infty$.

2.4 Determining the Number of Clusters

The UNPaC method can be extended to estimate the number of clusters present in X.

1. Optional dimension reduction for high dimensional data sets. Here we recommend selecting the 5% of features with the highest values of the critical bandwidth times variance.

2. Cluster X with the number of clusters taking values of $k = 1, \ldots, K_{\max}$.

3. For each k, calculate the cluster index.

4. Generate B unimodal reference distributions. We use $B = 100$ simulations. (See section 2.2 steps 1 and 2a through 5c).

5. Calculate the cluster index for the reference distribution CI_{bk}^{*} for $b = 1, \ldots, B$, $k = 1, \ldots, K_{\max}$.

6. An optional cluster significance test can be used to test the null hypothesis that $k = 1$ to the alternative that $k = k' > 1$. In the simulation studies, we perform a test of the null hypothesis of $k = 1$ clusters to the alternative that $k = 2$. If the test rejects the null hypothesis, proceed to steps 7 and 8. Otherwise choose $k^* = 1$.

7. Let $CI_{\text{diff}}(k) = \frac{1}{B}\Sigma_{b=1}^{B}\{CI_{bk}^{*} - CI_k\}$.

8. Choose the number of clusters k^* to be the value of k that maximizes $CI_{\text{diff}}(k)$.

2.5 Simulations

In this section, a simulation study is conducted to compare the UNPaC test to existing methods. We test cluster significance in both low and high dimensional situations (sections 2.5.2 and 2.5.3, respectively). In the low dimensional simulations, sample covariance estimation was used, and 100 replications were performed. In the high dimensional simulations, the dimension reduction technique discussed in Section 2.2 Step 2 was implemented, and the graphical lasso (Friedman et al., 2008) with ρ=.02 was used to estimate the covariance structure. Ten replications were performed for each high-dimensional example.

For estimating the number of clusters, we also consider both low and high dimensional situations (sections 2.5.4 and 2.5.5, respectively). For each low dimensional data set, 100 simulations were performed and the maximum number of clusters $K_{\max}$ was set to be 10. For high dimensional scenarios, the dimension reduction strategy given in Section 2.4 Step 1 was implemented, and covariance was estimated using the graphical lasso (Friedman et al., 2008) with ρ=.02. Due to the increased computational time, each high dimensional data set was simulated 10 times, and $K_{\max}$ was set to be 5.

2.5.1 Technical details

All methods were implemented in R version 3.2.2. For cluster significance testing, k-means clustering with $k = 2$ was applied to the scaled and centered data. If possible, the same cluster identities were fed into each testing method. Only clusters with at least 2 observations were accepted. The UNPaC test method utilizes the "sparcl" v 1.0.3 and "glasso" v 1.8 R packages.

The SigClust v. 1.1.0 algorithm was used with 1000 simulations, and the covariance was estimated via soft-thresholding (Huang et al., 2015), sample covariance estimation, and/or hard-thresholding (Liu et al., 2008). When $n > p$, the bootstrapping for significance (BFS) method described in Maitra et al. (2012) was implemented with 1000 replicates and

the assumption of heterogeneous clusters. Clustering was significant if the p-values for both the test of 0 versus 1 cluster and 1 versus 2 clusters were less than 0.05. The multimodality of principal curves (MPC) method of Ahmed and Walther (2012) was implemented by first using "princurve" v. 1.1-12 with the maximum number of iterations set to 100 and default parameter specifications. If convergence for the principal curve occurred, then Silverman's bandwidth test (Silverman, 1981) with 10,000 bootstrap samples was conducted using code located at http://www-bcf.usc.edu/ gourab/code-bmt/tables/table-2/silverman.test.R.

The IGP method of Kapp and Tibshirani (2006) was implemented by first calculating cluster centroids. These centroids, along with an additional simulated data set, were then used in the "clustRepo" v 0.5-1.1 implementation of the method with 1000 permutations. Since the IGP calculates reproducibility, for comparative purposes, we calculated the number of times the p-values were less than 0.05 for both clusters.

For determining the number of clusters, k-means was used for all methods. The gap method (Tibshirani et al., 2001) was implemented using the "cluster" v.2.0.3 package with default specifications of 100 bootstrap samples. The value of k^* was chosen to be the smallest value of k within one standard error of the value of k that maximizes the gap statistic. The calculation of the CH Index (Calinski and Harabasz, 1974) and average silhouette width (Rousseeuw, 1987) and implementation of the prediction strength method (Tibshirani and Walther, 2005) and cluster stability method (Fang and Wang, 2012) were all done using the "fpc" v.2.1-10 package with default parameter specifications unless otherwise noted. Since k-means was used to cluster the data, the centroid method was used to classify non-clustered points when implementing the cluster stability method of (Fang and Wang, 2012).

2.5.2 Low Dimensional Cluster Significance Simulations

Four null examples were used to assess the type I error of the cluster assessment methods. The "5-d sphere" example consisted of 1000 observations with five features. The features

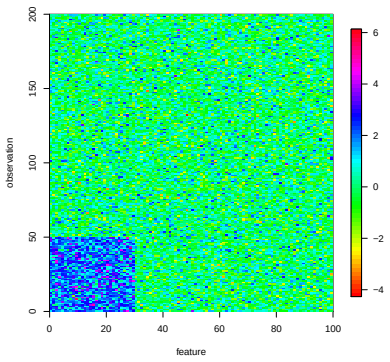

Figure 2.2: *Heat map of "Normal clustered" simulation.*

were generated uniformly on the surface of a 5-dimensional sphere. The "Null normal" example consisted of 200 observations with 100 features. Each data entry X_{ij} followed an i.i.d. N(0,1) distribution for $i = 1, \ldots, 200$, $j = 1, \ldots, 100$. The "Null correlated" example also consisted of 200 observation with 100 normally distributed features, but some of the features were correlated. Specifically, each observation vector, X_i, was distributed as $N(\mathbf{0}, \Sigma)$ where $\Sigma_{jj} = 1$ and $\Sigma_{jk} = 0.2I(j \leq 40, k \leq 40)$ for features $j = 1, \ldots, 100$, $k = 1, \ldots, 100$. The "Null t" example had 200 observations with 100 features and each data entry X_{ij} had an i.i.d t_2 distribution. (Here t_d represents the t distribution with d degrees of freedom).

Four clustered examples were also generated. Each example contained two clusters. The "Normal clustered" example had 200 observations with 100 normally distributed features. Observation vectors in the first cluster ($i \leq 50$) had the following distribution: $X_i \sim N(\boldsymbol{\mu}, \boldsymbol{I})$ with $\mu_j = 2I(j \leq 30)$, $j = 1, \ldots, 100$. The observation vectors in the second cluster ($51 \leq i \leq 200$), had a standard multivariate Gaussian distribution, $X_i \sim N(\mathbf{0}, \boldsymbol{I})$. A heat map of one sample simulation is given in Figure 2.2.

The "t clustered" example had 200 observations with 100 t-distributed features. Data entries in the first cluster ($i \leq 40$) had the following distribution: $X_{ij} \sim t_{2,12}$ for $j \leq 30$ and $X_{ij} \sim t_2$ for $31 \leq j \leq 100$. (Here $t_{2,12}$ represents the t distribution with 2 degrees of

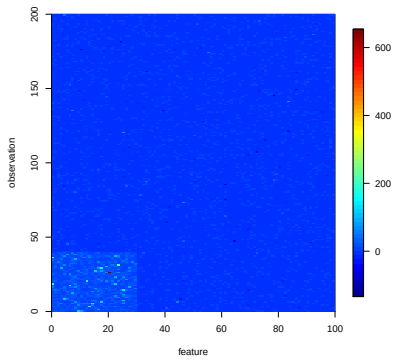

Figure 2.3: *Heat map of "T clustered" simulation.*

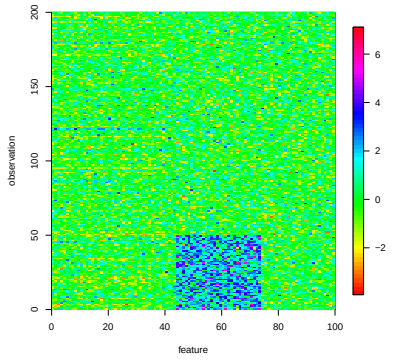

Figure 2.4: *Heat map of "Correlated clusters" simulation.*

freedom and a non-centrality parameter of 12). Each X_{ij} in the second cluster ($41 \leq i \leq 200$), followed a t_2 distribution. A heat map for one sample simulation is given in Figure 2.3

The "Correlated clusters" example had 200 observations and 100 features. The background data had the following distribution: $Z_i \sim N(\mathbf{0}, \Sigma)$, where $\Sigma_{jj} = 1$ and $\Sigma_{jk} = 0.2I(j \leq 40, k \leq 40)$ for features $j = 1, \ldots, 100$, $k = 1, \ldots, 100$. To generate the data entries X_{ij} for the final simulated data set, we let $X_{ij} = Z_{ij} + Y_{ij}$ for $i \leq 50$ and $45 \leq j \leq 74$, where each $Y_{ij} \sim N(2, 1)$. Otherwise we let $X_{ij} = Z_{ij}$. Figure 2.4 gives a heat map for one sample simulation.

The final example ("Elongated clusters") had 202 observations and 3 features. First, let $t_i = -0.50 + 0.01(i-1)$ and $\epsilon_{ij} \sim N(0, 0.10)$. The entries for the first cluster were generated

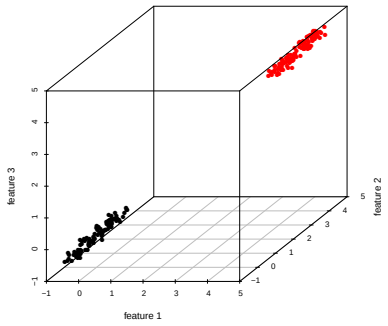

Figure 2.5: *Illustration of "Elongated clusters" simulation.*

as $X_{ij} = t_i + \epsilon_{ij}$ for $i \leq 101$. For observations in the second cluster ($102 \leq i \leq 202$), $X_{ij} = (t_{i-101}) + \epsilon_{ij} + 4$. A visual representation of this simulation is given in Figure 2.5.

Table 2.1: *Comparison of cluster detection accuracy for low dimensional clustering examples.* The number of times each method gave a p-value < 0.05 out of 100 simulations is recorded.

Simulation Name	Number of Simulations with p-value < 0.05						
	UNPaC	SigClust 1	SigClust 2	SigClust 3	IGP	BFS	MPC
5-d sphere	12	NA	72	98	1	0	97
Null normal	0	0	0	0	0	0	61
Null correlated	0	20	0	92	99	0	29
Null t	2	0	0	0	0	98	87
Normal clustered	100	100	100	100	18	100	100
t clustered	91	60	60	60	8	100	82
Correlated clusters	100	100	100	100	58	82	30
Elongated clusters	100	100	100	100	0	100	100

SigClust 1, SigClust 2, and SigClust 3 represent SigClust implemented using soft-thresholding (Huang et al., 2015), sample covariance estimation, and hard-thresholding (Liu et al., 2008), respectively. IGP= In Group Proportion (Kapp and Tibshirani, 2006). BFS=Bootstrap for Significance (Maitra et al., 2012). MPC= Multimodality of Principal Curves (Ahmed and Walther, 2012).

The results from the low dimensional simulations are given in Table 2.1. Overall, UNPaC performed very well. The method had perfect performance in the "Null normal," "Null correlated," "Normal clustered," "Correlated clusters," and "Elongated clusters" examples. UNPaC had a low significant cluster detection rate in the "5-d sphere", but was outperformed by IGP and BFS. In the "Null t" example, UNPaC only detected (spurious) clusters in 2 percent of the simulations. In this scenario, SigClust and IGP did not detect any significant clusters. UNPaC also had very good performance in the "t clustered" example, detecting clusters in 91% of the simulations. Only BFS performed better in this example. It should be

noted that MPC had a very high probability of concluding clusters were present even in null examples.

2.5.3 High Dimensional Cluster Significance Simulations

The type-I error rates of the cluster validation methods were also assessed in three high dimensional null examples. Each example had 100 observations and 10,000 features. The "Null normal" example was simply an independent standard Gaussian matrix with each data entry X_{ij} having a N(0,1) distribution for $i = 1, \ldots, 100$, $j = 1, \ldots, 10,000$. In the "Null correlated" simulation, each observation vector X_i followed a $N(\mathbf{0}, \Sigma)$ distribution where $\Sigma_{jj} = 1$ and $\Sigma_{jk} = I(|j-k| < 42)(.80)^{|j-k|}$ for $i = 1, \ldots, 100$, $j = 1, \ldots, 10,000$, and $k = 1, \ldots, 10,000$. In the "Null t" example, each data entry X_{ij} had a t_2 distribution.

Two clustered simulation examples were also generated. Each example contained two clusters. For the "Normal clustered" example, observation vectors in the first cluster ($i \leq 30$) had the following distribution: $X_i \sim N(\boldsymbol{\mu}, \boldsymbol{I})$ with $\mu_j = 2I(j \leq 50)$, $j = 1, \ldots, 10,000$. The observation vectors in the second cluster ($31 \leq i \leq 100$) had a standard multivariate Gaussian distribution $X_i \sim N(\mathbf{0}, \boldsymbol{I})$.

Data entries in the first cluster for the "t clustered" example ($i \leq 30$) followed the following distribution: $X_{ij} \sim t_{2,12}$ for $j \leq 100$ and $X_{ij} \sim t_2$ for $j > 100$. The data entries X_{ij} in the second cluster ($31 \leq i \leq 100$) followed a t_2 distribution.

Table 2.2: *Comparison of cluster detection accuracy for high dimensional clustering examples*. The number of times each method gave a p-value < 0.05 out of 10 simulations is recorded. The average number of features (p^*) selected by the dimension reduction step in UNPaC is also noted.

Simulation Name	p^*	**Number of Simulations with p-value < 0.05**					
		UNPaC	SigClust 1	SigClust 2	SigClust 3	IGP	MPC
Null normal	1218.7	0	0	0	0	0	7
Null correlated	1499.3	0	0	0	0	0	8
Null t	1076.2	0	0	0	0	0	8
Normal clustered	1086.7	8	0	0	0	3	7
T clustered	994	9	0	0	1	2	9

SigClust 1, SigClust 2, and SigClust 3 represent SigClust implemented using soft-thresholding (Huang et al., 2015), sample covariance estimation, and hard-thresholding (Liu et al., 2008), respectively. IGP= In Group Proportion (Kapp and Tibshirani, 2006). MPC= Multimodality of Principal Curves (Ahmed and Walther, 2012).

The UNPaC test's performance is especially noteworthy in the high dimensional examples (Table 2.2). UNPaC was either the top performer or tied as the top performer in all simulations. SigClust and IGP had perfect performance in the null examples but had poor performance in the clustered examples. Again, MPC tended to produce low p-values in most of the simulations regardless of whether clustering was actually present.

2.5.4 Low Dimensional Number of Clusters Simulations

Next, UNPaC was compared to existing methods for estimating the number of clusters in low dimensional examples. The "Null 1" example consisted of 100 observations with 20 features. The background data entries Z_{ij} had a $N(0,1)$ distribution. The final data entries X_{ij} were given by $X_{ij} = Z_{ij} + Y_{ij}$ for observations $76 \leq i \leq 100$ and features $j \leq 10$, where $Y_{ij} \sim N(0, 20)$. Otherwise $X_{ij} = Z_{ij}$. The "Null 2" example also had $N(0,1)$ background data entries Z_{ij}, and the final data entries X_{ij} were given by

$$X_{ij} = \begin{cases} Z_{ij} + \epsilon_{1_{ij}} & \text{if } j \leq 10 \text{ and } i \leq 20 \\ Z_{ij} + \epsilon_{2_{ij}} & \text{if } j \leq 10 \text{ and } 21 \leq i \leq 50 \\ Z_{ij} + \epsilon_{3_{ij}} & \text{if } j \leq 10 \text{ and } 51 \leq i \leq 75 \\ Z_{ij} + \epsilon_{4_{ij}} & \text{if } j \leq 10 \text{ and } 76 \leq i \leq 100 \\ Z_{ij} & \text{otherwise} \end{cases}$$

Here each $\epsilon_{1_{ij}} \sim N(0,1)$, $\epsilon_{2_{ij}} \sim N(0,3)$, $\epsilon_{3_{ij}} \sim N(0,5)$, and $\epsilon_{4_{ij}} \sim N(0,7)$.

Next, the performance of the methods was assessed when clusters were present. The "Three Clusters" example had 100 observation with 2 independent, normally distributed features. Cluster one had 25 observations with a feature mean of $(0,0)$, cluster two had 25 observations with a feature mean of $(0,5)$, and cluster three had 50 observations with a feature mean of $(5,-3)$. The "Four Clusters" simulation contained 100 observations with 20 features. The background features Z_{ij} were i.i.d $N(0,1)$. The final data entries X_{ij} were

given by

$$X_{ij} = \begin{cases} Z_{ij} + \epsilon_{1_{ij}} & \text{if } j \leq 10 \text{ and } i \leq 20 \\ Z_{ij} + \epsilon_{2_{ij}} & \text{if } j \leq 10 \text{ and } 21 \leq i \leq 50 \\ Z_{ij} + \epsilon_{3_{ij}} & \text{if } j \leq 10 \text{ and } 51 \leq i \leq 75 \\ Z_{ij} + \epsilon_{4_{ij}} & \text{if } j \leq 10 \text{ and } 76 \leq i \leq 100 \\ Z_{ij} & \text{otherwise} \end{cases}$$

where each $\epsilon_{1_{ij}} \sim N(1,1)$, $\epsilon_{2_{ij}} \sim N(8,1)$, $\epsilon_{3_{ij}} \sim N(15,1)$, and $\epsilon_{4_{ij}} \sim N(20,1)$.

Table 2.3: *Comparison of cluster selection methods for low dimensional examples*. The number of clusters, k, selected for each method in 100 simulations is presented. PredSt= method proposed by Tibshirani and Walther (2005), BootK= method proposed by Fang and Wang (2012), ASW= average silhouette width (Rousseeuw, 1987), and CH= Calinski-Harabasz index (Calinski and Harabasz, 1974).

	Null 1 k					*Null 2* k					*Three Clusters* k					*Four Clusters* k				
Method	1	2	3	4	≥5	1	2	3	4	≥5	1	2	3	4	≥5	1	2	3	4	≥5
UNPaC	87	0	0	0	13	100	0	0	0	0	0	5	95	0	0	0	50	0	48	2
Gap	60	14	9	8	9	100	0	0	0	0	0	12	88	0	0	0	29	29	41	1
PredSt	62	38	0	0	0	100	0	0	0	0	0	50	50	0	0	0	93	7	0	0
BootK	0	40	0	1	59	0	1	0	0	99	0	84	15	0	1	0	99	0	0	1
ASW	0	10	2	0	88	0	50	19	10	21	0	6	94	0	0	0	100	0	0	0
CH	0	4	2	0	94	0	92	7	1	0	0	0	100	0	0	0	0	0	100	0

The results of the simulations are shown in Table 2.3. UNPaC generally performed very well. It outperformed all of the other methods for the "Null 1" example and had perfect performance in the "Null 2" example. It had 95% accuracy in the "Three clusters" example and outperformed all methods except the CH index. All methods except the CH index struggled with the "Four Clusters" example, but it should be noted that the CH index tended to identify spurious clusters.

2.5.5 High Dimensional Number of Clusters Simulations

The performance of the methods was also evaluated in five high dimensional examples. Two examples were generated with no clusters. The "HD Null" example consisted of 100 observations with 10,000 iid $N(0,1)$ features. In the "HD Correlated" example, each

observation vector X_i followed a $N(\mathbf{0}, \mathbf{\Sigma})$ distribution where $\Sigma_{jj} = 1$ and $\Sigma_{jk} = I(|j-k| < 42)(.80)^{|j-k|}$. Here $i = 1, \ldots, 100$, $j = 1, \ldots, 10,000$, and $k = 1, \ldots, 10,000$.

Next, the performance was assessed in three simulations where clusters were present. In the "HD 2 Normal Clusters" example, observation vectors in the first cluster ($i \leq 30$) had the following distribution: $X_i \sim N(\boldsymbol{\mu}, \boldsymbol{I})$ with $\mu_j = 2I(j \leq 50)$, $j = 1, \ldots, 10,000$. The observation vectors in the second cluster ($31 \leq i \leq 100$) had a standard multivariate Gaussian distribution $X_i \sim N(\mathbf{0}, \boldsymbol{I})$. The "HD 3 Normal Clusters" example was similar with $X_i \sim N(\boldsymbol{\mu}, \boldsymbol{I})$ except $\mu_{ij} = [2^{I(i \leq 30)}][5^{I(30 < i \leq 60)}]I(j \leq 50)I(i \leq 60)$, $j = 1, \ldots, 10,000$. In the "HD 2 t Clusters" example, observations in the first cluster ($i \leq 30$) had the following distribution: $X_{ij} \sim t_{2,12}$ for $50 \leq j < 100$ and $X_{ij} \sim t_2$ otherwise. Data entries in the second cluster ($31 \leq i \leq 100$) followed a t_2 distribution.

Table 2.4: *Comparison of cluster selection methods for high dimensional examples*. The number of clusters, k, selected for each method in 10 simulations is presented. N=Normal. Corr=Correlated. PredSt= method proposed by Tibshirani and Walther (2005), BootK= method proposed by Fang and Wang (2012), ASW= average silhouette width (Rousseeuw, 1987), and CH= Calinski-Harabasz index (Calinski and Harabasz, 1974).

	HD Null k				*HD Corr* k				*HD 2 N Clusters* k				*HD 3 N Clusters* k				*HD 2 t Clusters* k			
Method	1	2	3	≥4	1	2	3	≥4	1	2	3	≥4	1	2	3	≥4	1	2	3	≥4
UNPaC	10	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0	8	1	1	0
Gap	10	0	0	0	10	0	0	0	10	0	0	0	10	0	0	0	10	0	0	0
PredSt	4	6	0	0	10	0	0	0	0	10	0	0	0	10	0	0	0	0	0	10
BootK	0	10	0	0	0	10	0	0	0	10	0	0	0	2	2	6	0	10	0	0
ASW	0	0	0	10	0	0	0	10	0	10	0	0	0	10	0	0	0	10	0	0
CH	0	10	0	0	0	10	0	0	0	10	0	0	0	10	0	0	0	10	0	0

The results are shown in Table 2.4. UNPaC had perfect performance in all of the HD examples except the "HD 2 t clusters" example. It is interesting to note that the UNPaC cluster significance test was able to correctly determine that clusters were present in a similar situation (the "T clustered example" in section 2.5.3). Since the significance test appears to have higher power than the method for determining the number of clusters, we recommend performing cluster significance testing as outlined in section 2.2 in addition to the method for choosing the number of clusters described in Section 2.4.

2.5.6 Simulation Summary

In the low dimensional cluster significance testing simulations, UNPaC tended to correctly fail to reject the null when no clusters were present and had good power to detect clusters when they were present. No competing method had similar accuracy across all simulations. In the high dimensional setting, UNPaC with dimension reduction and the graphical lasso also had good performance and even produced better results than SigClust in a situation where clusters were normally distributed.

The extension of UNPaC to estimating the number of clusters was also examined. UNPaC performed very well in both the low-dimensional setting (section 2.5.4) and high-dimensional setting (section 2.5.5). UNPaC did sometimes fail to identify all four clusters in the "Four clusters" simulations, but it still outperformed the majority of the methods. UNPaC also struggled in the "HD 2 t clustered scenario", but it did have good accuracy in a similar situation when testing the significance of the identified clusters.

2.6 An Application to Data from the OPPERA Study

We used data collected from the Orofacial Pain: Prospective Evaluation and Risk Assessment (OPPERA) study to illustrate the usefulness of the UNPaC method. This study has been described previously in Slade et al. (2011). In brief, OPPERA is a prospective cohort study with a case-control arm aimed at identifying risk factors for temporomandibular disorder (TMD). TMD is diagnosed based on painful symptoms in the masticatory muscles and temporomandibular joint (Schiffman et al., 2014), but several etiological mechanisms could be responsible for this disorder. Various measures of sensitivity to experimental pain, psychological distress, and autonomic function were evaluated for study participants, both with and without TMD, using questionnaires and clinical assessments.

Bair et al. (2016) identified three clinically important subgroups within the OPPERA study. The data used in their analysis consisted of 115 phenotypic features collected from

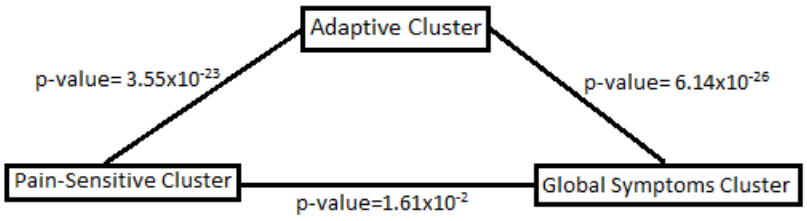

Figure 2.6: *OPPERA cluster significance tests:* Normal approximation p-values for testing significance of clusters identified in the OPPERA study.

1031 TMD cases and 3247 TMD-free controls. They used supervised clustering (Bair and Tibshirani, 2004) and the gap statistic (Tibshirani et al., 2001) to determine that three subgroups were present after selecting 25 features most strongly associated with TMD. These subgroups are called the Adaptive Cluster (1426 total individuals), the Pain-Sensitive Cluster (2062 total individuals), and the Global Symptoms Cluster (790 total individuals) based on their risk factor characteristics. Specifically, the Adaptive Cluster has low pain sensitivity and low psychological distress, the Pain-Sensitive Cluster has high pain sensitivity and low psychological distress, and the Global Symptoms Cluster has high pain sensitivity and high psychological distress.

We examined the strength of these putative clusters using UNPaC by performing pairwise comparisons between each of the three previously identified clusters. For our analysis, we used the same data that was used for cluster identification in (Bair et al., 2016), namely the scaled and centered data for the 25 features most strongly associated with TMD. For comparative purposes, instead of presenting the permutation p-value for our tests, we use the normal approximation to calculate p-values.

We find that all three clusters are well separated, but the Pain-Sensitive Cluster and the Global Symptoms Cluster are the most similar (Figure 2.6). To examine the possibility that more than three clusters were present in the OPPERA data, we used UNPaC to reassess the number of clusters present. Four clusters were found (Figure 2.8) with 1077 individuals (49 TMD cases) in cluster "A," 1544 individuals (324 TMD cases) in cluster "B," 1292 individuals (428 TMD cases) in Cluster "C," and 365 individuals (220 TMD cases) in cluster

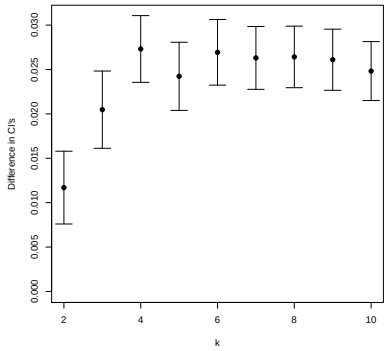

Figure 2.7: *Optimal number of clusters for OPPERA data.* Plot of difference in cluster indices (CI's) between observed data and reference distribution for a range of number of clusters, k. k=4 maximizes the difference in CI's.

"D." Comparing the four clusters identified by UNPaC to the three found in Bair et al. (2016) we find that Cluster "A" is nearly identical to the Adaptive Cluster and Cluster "D" is nearly identical to the Global Symptoms Cluster (Figure 2.8). These clusters have similar feature characteristics to the respective clusters identified in Bair et al. (2016) (Table 2.8). The Pain Sensitive Cluster is split into two clusters with one cluster including some individuals from the Adaptive Cluster and the other including some individuals from the Pain Sensitive Cluster.

Examination of the standardized means of the features across the four clusters (2.8) reveals an interesting pattern. We find a more nuanced gradation of pain sensitivity and psychological distress symptoms than was previously identified. Specifically, we find that the original Pain-Sensitive cluster is split into two clusters dependent on the type of pain experienced. Specifically, individuals in Cluster "B" were more sensitive to pressure pain than individuals in Cluster "C." However, individuals in Cluster "C" reported greater sensitivity to mechanical pain and had greater aftersensation ratings than individuals in Cluster "B." This could be an important clinical finding, since different neurological mechanisms may be responsible for these different types of pain sensitivity, and the optimal treatment is likely to depend on the neurological mechanism that is causing the pain.

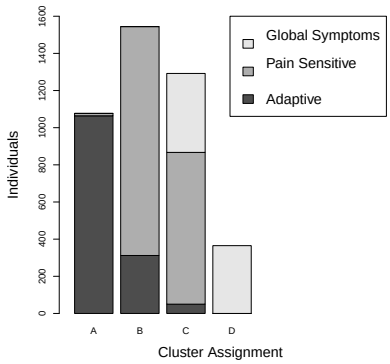

Figure 2.8: *Clusters identified using UNPaC on OPPERA data.* Overlap between clusters identified using UNPaC (Clusters "A","B","C", and "D") and clusters found in Bair et al. (2016) ("Adaptive Cluster","Pain Sensitive Cluster", and "Global Symptoms Cluster")

2.7 An Application to Breast Cancer Data

We applied UNPaC to test the significance of clusters and estimate the number of clusters in a breast cancer microarray data set supplied by C. M. Perou. This data set has previously been analyzed by Liu et al. (2008). The data set contained 306 genes from 254 individuals classified as having luminal A, luminal B, luminal I, her 2, or basal breast cancer types. Hierarchical clustering with average linkage was performed on the data. The dissimilarity structure used for the clustering was 1 minus the Pearson correlation.

The normalized p-values for testing the significance of k=3 through 10 clusters compared to the null were all very small ($p < 0.005$) indicating that more than two distinct clusters are indeed present (Table 2.5). From Figure 2.9, it can be seen that 5 clusters is the optimal number of clusters for the data. Further examination of the subgroups identified by UNPaC (Figure 2.10) reveals that the clustering closely follows the previously identified clusters, but that there may be some overlap between the subgroups.

Table 2.5: *Normalized p-values for testing the significance of k=2:10 versus no clusters in the breast cancer microarray data.*

k	2	3	4	5	6	7	8	9	10
p	9.8×10^{-2}	4.7×10^{-3}	3.4×10^{-14}	4.0×10^{-21}	1.4×10^{-21}	2.9×10^{-25}	9.7×10^{-28}	2.3×10^{-30}	3.8×10^{-32}

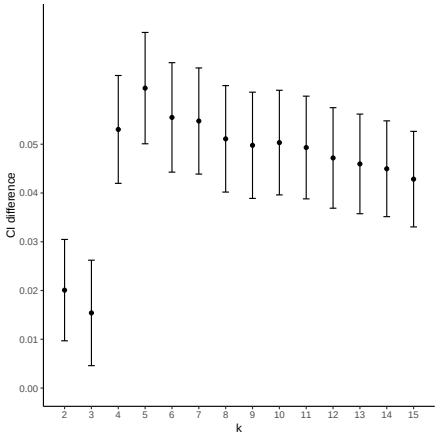

Figure 2.9: *Optimal number of clusters for breast cancer microarray data.* Plot of difference in cluster indices (CI's) between observed data and reference distribution for a range of number of clusters, k. k=5 maximizes the difference in CI's.

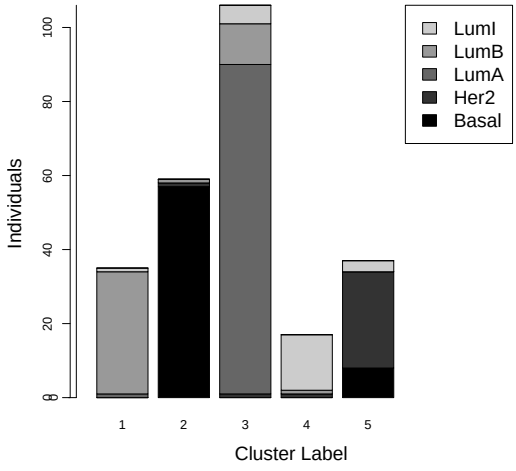

Figure 2.10: *Clusters identified using UNPaC on breast cancer microarray data.* Overlap between clusters identified using UNPaC (Clusters "1","2","3", "4", and "5") and cancer subtypes ("Basal","Her2", "LumA","LumB", and "LumI")

2.8 Discussion

Clustering can be useful for discovering underlying structure in data, but it is important to determine if the putative clusters represent truly distinct subgroups rather than noise. In this paper, we have developed a nonparametric approach to test the null hypothesis of no clusters in the data by comparing the cluster index from the given clustering to what would be expected under a reference unimodal distribution. By extending this method to compare the difference in cluster indices from the null and observed data, the number of clusters present in the data can be estimated. Through our simulation studies and applications to the OPPERA study and breast cancer microarray data, we have shown that UNPaC is a useful tool for testing the strength of putative subgroups and estimating the number of clusters.

In the simulation studies, we found that our method compares favorably with competing methods, even outperforming other methods under certain conditions. Since our method does not require parametric assumptions, it is especially useful when the data deviates from normality. By using the information from multiple features, UNPaC is better able to assess the clustering structure of the data than methods which reduce the data to a univariate summary. Unlike other nonparametric methods, our method is specifically adapted for the high dimensional setting through the use of dimension reduction techniques. Dimension reduction may even increase testing precision. If only a portion of the features are responsible for the clusters, then removing some of the extraneous features would reduce the noise in the data and produce more accurate cluster assignments.

One important aspect of our method is that it is agnostic to the method that was used to cluster the data. The method can be applied to test the significance of clusters produced using other clustering methods, including hierarchical clustering. Also, the L_2 distance used to calculate the cluster index could easily be replaced with another distance measure.

It should be noted that UNPaC is specifically built under the assumption that a single cluster (or unclustered data) comes from a unimodal distribution. Thus, like most clustering

significance testing methods, UNPaC can only be applied to continuous data. It is not suited for testing clustering when the underlying data structure is naturally bimodal (such as bivariate data) or multi-modal (such as categorical data).

The current implementation of our method identifies clusters that differ based on feature means. An avenue of future research would be to apply these method to identify clusters which differ based on feature variance. One possible approach would be to perform our test on the singular values instead of the original data. Also, the cluster index used as a test statistic is very susceptible to outliers, and future research is needed to analyze how our method can best be applied in this situation. In some situations, simple preprocessing steps, such as outlier removal, can be a reasonable step before testing for clustering. Future research on this topic could include using a weighted cluster index in order to reduce the effect of outliers on the cluster assignment.

Table 2.6: *Pain Sensitivity, Psychosocial, and Autonomic Feature Summary Statistics by Cluster for Four Clusters Identified in OPPERA data.* AS= Aftersensation, AUC=Area Under the Curve, BP=Blood Pressure, CSQ=Coping Strategies Questionnaire, EPQ-R=Eysenck Personality Questionnaire-Revised, HRV=Heart Rate Variability,IP=Interpersonal KRS=Kohn Reactivity Scale, LES=Life Experiences Survey, PCS=Pain Catastrophizing Scale, PILL=Pennebaker Inventory for Limbic Languidness, POMS=Profile of Mood States, PPT=Pressure Pain Threshold, PSQI=Pittsburgh Sleep Quality Index, PSS=Perceived Stress Scale, SCL90-R=Symptom Checklist-90-Revised, SS=Single Stimulus, TMJ=temporomandibular joint, TS=Temporal Summation. See Bair et al. (2016) for more information about the data.

	A mean	A sd	B mean	B sd	C mean	C sd	D mean	D sd	P-value [1]	A vs B [2]	A vs C [2]	A vs D [2]	B vs C [2]	B vs D [2]	C vs D [2]
PPT: Temporalis	296.76	81.07	161.06	46.05	178.88	64.25	169.92	77.21	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0357	0.0432
PPT: Masseter	275.36	73.34	146.24	41.95	161.71	58.36	150.98	71.07	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.2219	0.0085
PPT: TMJ	244.08	63.07	136.19	39.18	150.22	53.55	136.85	60.92	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.8445	0.0002
PPT: Trapezius	512.75	94.35	272.84	99.58	309.35	127.54	288.54	130.52	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0316	0.0071
PPT: Epicondyl	525.22	91.81	292.87	107.86	330.73	131.99	315.28	140.10	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0044	0.0600
Mechanical Pain Threshold	325.22	176.15	197.50	153.40	201.67	155.14	177.15	151.78	<0.0001	<0.0001	<0.0001	<0.0001	0.4735	0.0219	0.0069
Mechanical SS (256 mN)	5.88	9.50	10.70	13.93	11.88	15.67	14.00	17.65	<0.0001	<0.0001	<0.0001	<0.0001	0.0355	0.0009	0.0383
Mechanical SS (512 mN)	10.98	15.19	19.68	21.29	21.17	22.63	23.09	23.98	<0.0001	<0.0001	<0.0001	<0.0001	0.0731	0.0128	0.1711
Mechanical AS (256 mN, 15 s.)	1.98	5.91	3.90	8.62	5.16	10.35	6.74	13.82	<0.0001	<0.0001	<0.0001	<0.0001	0.0005	0.0002	0.0433
Mechanical AS (256 mN, 30 s.)	0.95	3.69	2.04	6.13	2.74	7.12	3.67	10.47	<0.0001	<0.0001	<0.0001	<0.0001	0.0053	0.0043	0.1101
Mechanical AS (512 mN, 15 s.)	5.05	10.92	10.28	16.63	12.46	18.10	13.50	20.67	<0.0001	<0.0001	<0.0001	<0.0001	0.0009	0.0057	0.3814
Mechanical AS (512 mN, 30 s.)	2.84	7.89	5.96	12.44	7.43	13.75	8.20	16.79	<0.0001	<0.0001	<0.0001	<0.0001	0.0030	0.0167	0.4227
Mechanical Windup (256 mN)	6.53	9.73	11.95	13.74	11.89	13.32	12.79	15.19	<0.0001	<0.0001	<0.0001	<0.0001	0.8955	0.3370	0.3045

[1] P-value for testing the null hypothesis that the mean value does not differ between all of the clusters

[2] P-value for the null hypothesis that the mean value does not differ between the respective clusters

Mechanical Windup (512 mN)	12.85	14.67	20.05	17.66	19.65	17.00	20.48	17.70	<0.0001	<0.0001	<0.0001	<0.0001	0.5399	0.6820	0.4297
Heat Pain Tolerance	47.26	2.05	45.50	2.33	45.61	2.35	45.46	2.58	<0.0001	<0.0001	<0.0001	<0.0001	0.2046	0.7916	0.3145
TS: First Pulse (46 degrees)	26.37	26.55	42.76	31.47	41.62	31.42	44.32	32.54	<0.0001	<0.0001	<0.0001	<0.0001	0.3384	0.4056	0.1587
TS: First Pulse (48 degrees)	34.62	29.26	51.83	32.19	51.27	31.74	53.58	32.55	<0.0001	<0.0001	<0.0001	<0.0001	0.6378	0.3558	0.2283
TS: First Pulse (50 degrees)	47.10	32.79	65.15	31.19	64.17	31.13	67.76	31.13	<0.0001	<0.0001	<0.0001	<0.0001	0.4023	0.1504	0.0519
TS: Delta (46 degrees)	15.46	24.84	18.52	24.50	17.74	24.69	18.47	25.43	0.0136	0.0018	0.0259	0.0494	0.3956	0.9714	0.6236
TS: Delta (48 degrees)	20.13	24.69	21.77	24.99	21.73	24.95	21.01	24.04	0.3362	0.0956	0.1185	0.5480	0.9627	0.5893	0.6178
TS: Delta (50 degrees)	20.08	25.62	18.33	23.91	18.80	24.10	15.79	25.16	0.0301	0.0767	0.2112	0.0052	0.6070	0.0804	0.0423
Thermal AS (46 degrees, 15 s.)	5.54	11.96	10.13	15.26	11.58	16.39	14.06	20.10	<0.0001	<0.0001	<0.0001	<0.0001	0.0156	0.0005	0.0312
Thermal AS (48 degrees, 15 s.)	9.14	15.29	15.24	19.26	17.24	20.40	19.31	22.88	<0.0001	<0.0001	<0.0001	<0.0001	0.0077	0.0018	0.1198
Thermal AS (50 degrees, 15 s.)	11.32	16.82	17.78	21.02	20.35	21.91	23.15	24.79	<0.0001	<0.0001	<0.0001	<0.0001	0.0016	0.0002	0.0517
Thermal AS (46 degrees, 30 s.)	3.10	8.76	5.62	11.14	6.44	12.01	8.03	14.83	<0.0001	<0.0001	<0.0001	<0.0001	0.0611	0.0037	0.0608
Thermal AS (48 degrees, 30 s.)	5.45	11.37	9.56	15.02	11.14	16.88	12.35	18.06	<0.0001	<0.0001	<0.0001	<0.0001	0.0089	0.0063	0.2508
Thermal AS (50 degrees, 30 s.)	6.52	12.68	10.33	15.63	12.52	17.53	14.71	20.53	<0.0001	<0.0001	<0.0001	<0.0001	0.0005	0.0002	0.0643
TS: AUC (46 degrees)	333.46	261.99	503.61	278.45	488.09	278.27	516.56	286.79	<0.0001	<0.0001	<0.0001	<0.0001	0.1393	0.4357	0.0924
TS: AUC (48 degrees)	442.06	278.96	608.38	260.56	602.17	260.51	621.25	260.18	<0.0001	<0.0001	<0.0001	<0.0001	0.5277	0.3959	0.2168
TS: AUC (50 degrees)	558.78	281.01	711.40	230.03	704.73	229.50	720.57	220.19	<0.0001	<0.0001	<0.0001	<0.0001	0.4415	0.4786	0.2299
Global KRS	67.18	11.79	71.97	10.91	75.52	11.21	79.06	11.97	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
PILL Global	28.32	17.74	32.28	18.48	51.48	24.82	82.36	32.65	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
POMS Agreeable-Hostile	29.01	4.82	29.90	4.17	24.39	5.31	20.91	6.21	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
POMS Elated-Depressed	27.97	4.84	28.74	4.27	21.89	5.36	16.94	6.26	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001

POMS Confident-Unsure	26.72	4.61	26.90	3.89	20.09	4.78	16.14	5.69	<0.0001	0.3051	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
POMS Energetic-Tired	25.45	5.94	25.48	5.40	17.83	5.75	13.74	6.47	<0.0001	0.9202	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
POMS Clearheaded-Confused	29.47	4.69	30.22	3.96	23.80	5.21	18.60	6.23	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
POMS Composed-Anxious	28.19	5.10	28.87	4.46	20.85	5.09	15.43	6.23	<0.0001	0.0004	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
POMS Overall Positive Affect	88.15	12.47	88.50	12.25	77.30	11.73	72.71	12.99	<0.0001	0.4739	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
POMS Overall Negative Affect	44.20	10.53	42.51	8.75	61.53	13.56	77.40	15.17	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
PSS Perceived Stress Scale	12.34	5.44	12.24	5.19	19.45	4.75	24.80	5.31	<0.0001	0.6232	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
PSQI Global	4.21	2.59	4.10	2.45	6.62	3.23	10.01	3.75	<0.0001	0.2406	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SCL 90R Depression	0.21	0.23	0.18	0.19	0.68	0.39	1.82	0.64	<0.0001	0.0041	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SCL 90R Somatization	0.17	0.21	0.20	0.23	0.47	0.38	1.34	0.65	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SCL 90R Anxiety	0.09	0.14	0.08	0.12	0.35	0.28	1.36	0.64	<0.0001	0.0714	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SCL 90R Obsessive Compulsive	0.28	0.31	0.24	0.27	0.74	0.46	1.91	0.65	<0.0001	0.0007	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SCL 90R IP Sensitivity	0.18	0.25	0.15	0.20	0.56	0.42	1.59	0.71	<0.0001	0.0052	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SCL 90R Hostility	0.17	0.25	0.14	0.20	0.45	0.40	1.30	0.79	<0.0001	0.0091	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SCL 90R Phobia	0.03	0.11	0.03	0.09	0.14	0.25	0.83	0.78	<0.0001	0.6471	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SCL 90R Paranoid	0.21	0.35	0.14	0.26	0.46	0.47	1.45	0.82	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SCL 90R Psychotic	0.09	0.17	0.06	0.14	0.27	0.29	1.05	0.67	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SCL 90R Global Severity	0.17	0.15	0.15	0.13	0.49	0.24	1.46	0.47	<0.0001	0.0013	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CSQ Distraction Scale	2.31	1.55	2.32	1.52	2.38	1.44	2.63	1.50	0.0029	0.8554	0.2475	0.0006	0.2771	0.0005	0.0056
CSQ Catastrophizing	0.65	0.73	0.70	0.77	1.48	1.06	2.42	1.37	<0.0001	0.0715	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CSQ Ignoring Pain Scale	2.82	1.46	2.60	1.47	2.42	1.34	2.49	1.43	<0.0001	0.0001	<0.0001	0.0002	0.0009	0.1889	0.4352
CSQ Distancing Scale	1.25	1.44	1.06	1.31	1.30	1.39	1.68	1.63	<0.0001	0.0009	0.3344	<0.0001	<0.0001	<0.0001	0.0001

CSQ Coping Scale	3.74	1.42	3.48	1.48	3.55	1.33	3.73	1.34	<0.0001	<0.0001	0.0008	0.9131	0.1975	0.0017	0.0215
CSQ Praying Scale	2.01	1.93	2.15	2.05	2.39	2.03	2.88	2.03	<0.0001	0.0788	<0.0001	<0.0001	0.0022	<0.0001	<0.0001
State Anxiety Inventory	27.99	7.44	27.18	6.43	37.05	9.09	46.19	11.50	<0.0001	0.0037	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Trait Anxiety Inventory	31.99	7.17	31.39	6.31	43.31	7.08	53.33	8.84	<0.0001	0.0278	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
LES Sum of Negative Events	4.82	6.66	4.62	5.73	8.21	8.01	15.20	12.25	<0.0001	0.4247	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
LES Sum of Positive Events	5.91	5.56	6.19	6.06	5.70	5.65	6.91	7.07	0.0036	0.2201	0.3802	0.0143	0.0281	0.0732	0.0029
PCS Rumination	3.00	3.40	3.37	3.36	6.09	4.14	9.28	4.45	<0.0001	0.0056	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
PCS Magnification	1.28	1.64	1.39	1.68	3.07	2.49	5.39	3.16	<0.0001	0.0960	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
PCS Helplessness	2.09	2.72	2.54	3.12	6.01	4.70	10.67	5.89	<0.0001	0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
EPQ-R Extraversion	8.81	3.07	8.98	3.00	7.68	3.47	6.93	3.66	<0.0001	0.1545	<0.0001	<0.0001	<0.0001	<0.0001	0.0005
EPQ-R Lie	6.72	3.57	6.66	3.44	5.44	3.12	5.46	3.17	<0.0001	0.6778	<0.0001	<0.0001	<0.0001	<0.0001	0.8929
EPQ-R Neuroticism	2.74	2.47	3.15	2.49	6.89	2.71	9.06	2.51	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
EPQ-R Psychoticism	2.49	1.90	2.16	1.73	2.58	2.00	3.13	2.15	<0.0001	<0.0001	0.2380	<0.0001	<0.0001	<0.0001	<0.0001
HRV: Total Power	7.22	0.89	7.22	0.87	7.11	0.95	6.89	1.06	<0.0001	0.9828	0.0062	<0.0001	0.0028	<0.0001	0.0002
HRV: Very Low Frequency	6.15	0.88	6.10	0.85	5.98	0.92	5.79	1.04	<0.0001	0.1011	<0.0001	<0.0001	0.0007	<0.0001	0.0019
HRV: Low Frequency	5.95	0.97	5.90	0.98	5.79	1.05	5.54	1.17	<0.0001	0.2169	0.0002	<0.0001	0.0045	<0.0001	0.0003
Average Systolic BP	112.99	10.69	109.90	10.36	110.31	10.70	111.29	11.01	<0.0001	<0.0001	<0.0001	0.0105	0.2969	0.0280	0.1309
Average Diastolic BP	64.97	8.21	64.90	7.62	65.55	7.90	66.47	8.19	0.0020	0.8326	0.0814	0.0027	0.0273	0.0010	0.0580
Average Mean Arterial Pressure	83.56	8.52	82.07	8.13	82.62	8.52	83.69	8.84	<0.0001	<0.0001	0.0076	0.8042	0.0802	0.0015	0.0399
Average Heart Rate	62.34	10.46	62.71	10.34	63.81	10.47	65.85	11.48	<0.0001	0.3761	0.0007	<0.0001	0.0049	<0.0001	0.0024

2.9 Proof of Theorems

A1. Proof of Theorem 1

Before multiplying by the Cholesky root, for a given feature j, the Gaussian kernel density is given by

$$\hat{f}_j(t; h_{1j}) = \frac{1}{nh_{1j}} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2} \left[\frac{t_j - x_{ij}}{h_{1j}} \right]^2 \right\}$$

where h_{1j} is the minimum bandwidth such that $f_j(t_j)$ is unimodal and the x_{ij}'s are the observed data points for the jth feature.

After multiplying by the Cholesky root, each feature becomes a linear combination $Y_k = g_k(T_1, ..., T_p) = \Sigma_{i=1}^{p}(a_{ki}T_i)$ for constants $a_{k1}, \ldots, a_{kp}$. Equivalently, $T_k = g_k^{-1}(Y_1, ..., Y_p) = \sum_{i=1}^{p}(b_{ik} * Y_i)$ for constants $b_{k1}, \ldots, b_{kp}$. We need to show that the resulting multivariate distribution $h(y_1, ..., y_p)$ is multivariate unimodal.

Using the multivariate transformation of variables formula where $|J(y_1, ..., y_p)|$ is the Jacobian of the transformation, we have that

$$h_{Y_1,...,Y_p}(y_1, ..., y_p) = \{ f_{T_1,...,T_p}(g_1^{-1}(y_1, ...y_p), ..., g_p^{-1}(y_1, ..., y_p)) \times |J(y_1, ..., y_p)| \}.$$

Since each Y_k is a linear combination of the T_i's, the Jacobian will be a constant. Also, since the T_i's are independent

$$h_{Y_1,...,Y_p}(y_1, ...y_p) = |J(y_1, ..., y_p)| \cdot \prod_{i=1}^{p} f_i(g_i^{-1}(y_1, ..., y_p))$$

Note that $\prod_{i=1}^{p} f_i(g_i^{-1}(y_1, ..., y_p))$ is maximized when each $f_i(g_i^{-1}(y_1, ..., y_p))$ is maximized, which happens at the unique mode, m_i. This is also the only point for which definition 2.5 from Sager (1978) is satisfied. Thus, the multivariate mode of $h_{Y_1,...,Y_p}(y_1, ..., y_n)$ is the solution to the system of equations $m_i = g_i^{-1}(y_1, ..., y_p) = \sum_{k=1}^{p}(b_{ik}Y_k)$ for $k = 1, ..., p$

and $i = 1, ..., p$. Since each Y_k is a non-degenerate linear mapping of the $T_i's$, a unique solution exists. Hence $h(y_1, ..., y_n)$ is multivariate unimodal.

A2. Proof of Theorem 2

Huang et al. (2015) show that for the choice of S_1 and S_2 (the two non-overlapping subspaces of the feature space) that minimizes TWSS, $TCI_{\text{GAUSS}} = 1 - \frac{2}{\pi}\frac{\lambda_1}{\sum_{j=1}^{p}\lambda_j}$. To calculate TCI_{null}, first note that for the null distribution, the density of a given feature t_j is given by

$$f_j(t_j) = \frac{1}{n}\sum_{i=1}^{n}\frac{\sqrt{1+h_{1j}^2}}{h_{1j}\sqrt{\lambda_j 2\pi}}\exp\left\{-\frac{1}{2}\left[\frac{t_j\sqrt{1+h_{1j}^2}-\sqrt{\lambda_j}x_{ij}}{h_{1j}\sqrt{\lambda_j}}\right]^2\right\}$$

where h_{1j} is the minimum bandwidth such that $f_j(t_j)$ is unimodal and x_{ij} is the observed jth feature for the ith observation. Note that each x_{ij} is scaled and centered such that the sample mean is equal to 0 and the sample variance is equal to 1.

Let $f(\boldsymbol{t}) = \prod_{j=1}^{p} f_j(t_j)$. Then the total sum of squares for $\boldsymbol{x}$ is given by:

$$\begin{aligned} TTSS &= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}||\boldsymbol{t}||^2 f(\boldsymbol{t})dt_1,\cdots,dt_p = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\sum_{j=1}^{p}t_j^2\left(\prod_{j=1}^{p}f_j(t_j)\right)dt_1,\cdots,dt_p \\ &= \sum_{j=1}^{p}\int_{-\infty}^{\infty}t_j^2 f_j(t_j)dt_j = \sum_{j=1}^{p}\frac{\lambda_j}{1+h_1^2}\left[h_1^2 + 2h_{1j}\frac{1}{n}\sum_{i=1}^{n}x_{ij} + \frac{1}{n}\sum_{i=1}^{n}x_{ij}^2\right] = \sum_{j=1}^{p}\lambda_j \end{aligned}$$

Since the greatest variation is in the first feature, the separating plane will be the plane through $\boldsymbol{\mu} = (0, \ldots, 0)^T$ that is orthogonal to $(1, \ldots, 0)^T$. Let $\mu_1 = (\mu_{11}, \ldots, \mu_{1p})$. By symmetry, we have that $\mu_{12} =, \ldots, = \mu_{1p} = 0$. Next, we need to find μ_{11}. Using 2 as the normalization constant, we have that

$$\mu_{11} = 2\int_0^{\infty}t_1 f_1(t_1)dt_1 = \frac{2}{n}\sum_{i=1}^{n}\int_0^{\infty}t_1\frac{\sqrt{1+h_{11}^2}}{h_{11}\sqrt{\lambda_1 2\pi}}\exp\left\{-\frac{1}{2}\left[\frac{x_1\sqrt{1+h_{11}^2}-\sqrt{\lambda_1}x_{i1}}{h_{11}\sqrt{\lambda_1}}\right]^2\right\}dt_1$$

and hence

$$\lim_{n\to\infty}\mu_{11} = 2\int_{-\infty}^{\infty}\int_{0}^{\infty} t_1 \frac{\sqrt{1+h_{11}^2}}{h_{11}\sqrt{\lambda_1 2\pi}} \exp\left\{-\frac{1}{2}\left[\frac{t_1\sqrt{1+h_{11}^2}-\sqrt{\lambda_1}x_1}{h_{11}\sqrt{\lambda_1}}\right]^2\right\}\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}x_1^2\right\}dt_1 dx_1$$
$$= 2\int_0^{\infty}\frac{t}{\sqrt{\lambda 2\pi}}\exp\left\{-\frac{t^2}{2\lambda}\right\}dt = \sqrt{\frac{2\lambda_1}{\pi}}$$

Similarly $\mu_{21} = -\sqrt{\frac{2\lambda_1}{\pi}}$ and $\mu_{22} =, ..., = \mu_{2p} = 0$

Then we have shown that the theoretical within cluster sum of squares for the first identified cluster is given by:

$$TWSS_1 = \int_0^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}||\boldsymbol{t}-\boldsymbol{\mu_1}||^2 f(\boldsymbol{t})dt_1,\cdots,dt_p$$
$$= \int_0^{\infty}(t_1-\mu_{11})^2 f_1(t_1)dt_1 + \sum_{j=2}^{p}\int_0^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} t_j^2 f(\boldsymbol{t})dt_1,\ldots,dt_p$$

The limit as $n\to\infty$ for the first piece of the $TWSS_1$ is given by:

$$\lim_{n\to\infty}\int_0^{\infty}(t_1-\mu_{11})^2 f_1(t_1)dt_1$$
$$= \lim_{n\to\infty}\int_0^{\infty}\frac{\left(t_1-\sqrt{\frac{2\lambda_1}{\pi}}\right)^2}{n}\sum_{i=1}^{n}\int_0^{\infty}\frac{\sqrt{1+h_{11}^2}}{h_{11}\sqrt{2\pi\lambda_1}}\exp\left\{-\frac{1}{2}\left[\frac{\sqrt{1+h_{11}^2}t_1-\sqrt{\lambda_1}x_{i1}}{h_{11}\sqrt{\lambda_1}}\right]^2\right\}dt_1$$
$$= \int_0^{\infty}\left(t_1-\sqrt{\frac{2\lambda_1}{\pi}}\right)^2\frac{\sqrt{1+h_{11}^2}}{h_{11}\sqrt{2\pi\lambda_1}}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}\left[\frac{\sqrt{1+h_{11}^2}t_1-\sqrt{\lambda_1}x_1}{h_{11}\sqrt{\lambda_1}}\right]^2\right\}\exp\left\{-\frac{x_1^2}{2}\right\}dx_1 dt_1$$
$$= \int_0^{\infty}\left(t_1-\sqrt{\frac{2\lambda_1}{\pi}}\right)^2\frac{1}{\sqrt{\lambda_1 2\pi}}\exp\left\{-\frac{t_1^2}{2\lambda_1}\right\}dt_1 = \frac{\lambda_1}{2}-\frac{\lambda_1}{\pi}$$

The limit as $n \to \infty$ for the second piece of the $TWSS_1$ is given by:

$$\begin{aligned}
&\lim_{n\to\infty} \sum_{j=2}^{p} \int_0^\infty \int_{-\infty}^\infty \cdots \int_{-\infty}^\infty t_j^2 f(\boldsymbol{t}) dt_1, \ldots, dt_p \\
&\quad = \lim_{n\to\infty} \sum_{j=2}^{p} \int_0^\infty f_1(t_1) dt_1 \left[\int_{-\infty}^\infty \cdots \int_{-\infty}^\infty t_j^2 f(\boldsymbol{t}) \right] dt_2, \ldots, dt_p \\
&\quad = \lim_{n\to\infty} \sum_{j=2}^{p} \int_0^\infty f_1(t_1) dt_1 \int_{-\infty}^\infty t_j^2 f(t_j) dt_j \\
&\quad = \lim_{n\to\infty} \sum_{j=2}^{p} \frac{\lambda_j}{n} \sum_{i=1}^{n} \int_0^\infty \frac{\sqrt{1+h_{11}^2}}{h_{11}\sqrt{2\pi\lambda_1}} \exp\left\{ -\frac{1}{2} \left[\frac{\sqrt{1+h_{11}^2} t_1 - \sqrt{\lambda_1} x_{i1}}{h_{11}\sqrt{\lambda_1}} \right]^2 \right\} dt_1 \\
&\quad = \sum_{j=2}^{p} \frac{\lambda_j}{\sqrt{2\pi}} \int_0^\infty \int_{-\infty}^\infty \frac{\sqrt{1+h_{11}^2}}{h_{11}\sqrt{2\pi\lambda_1}} \exp\left\{ -\frac{1}{2} \left[\frac{\sqrt{1+h_{11}^2} t_1 - \sqrt{\lambda_1} x_1}{h_{11}\sqrt{\lambda_1}} \right]^2 \right\} \exp\left\{ -\frac{x_1^2}{2} \right\} dx_1 dt_1 \\
&\quad = \sum_{j=2}^{p} \lambda_j \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_1}} \exp\left\{ \frac{y_1^2}{\lambda_1} \right\} dt_1 = \frac{1}{2} \sum_{j=2}^{p} \lambda_j
\end{aligned}$$

Thus $\lim_{n\to\infty} TWSS_1 = 1/2 \sum_{j=1}^{p} \lambda_j - \frac{\lambda_1}{\pi}$. We have shown that $TWSS_1 = TWSS_2$, so the theoretical cluster index for our null distribution is given by:

$$\frac{TWSS_1 + TWSS_2}{TTSS} = \frac{\sum_{j=1}^{p} \lambda_j - \frac{2\lambda_1}{\pi}}{\sum_{j=1}^{p} \lambda_j} = 1 - \frac{2}{\pi} \frac{\lambda_1}{\sum_{j=1}^{p} \lambda_j} = TCI_{GAUSS}$$

A3. Proof of Theorem 3

Again, as the number of observations, n, approaches infinity, the cluster index from the data approaches the theoretical cluster index. We will show that the theoretical cluster index from the mixture distribution CI_{mix} is less than the theoretical cluster index from the null distribution, CI_{null}. First, note that the variance for feature j in the data is given by $\lambda_j + \eta(1\eta)a^2$. Thus $CI_{\text{null}} = 1 - \frac{2}{\pi} \frac{\lambda_1 + \eta(1-\eta)a^2}{p\eta(1-\eta)a^2 + \sum_{j=1}^{p} \lambda_j}$

Next we determine the theoretical total sum of squares for the mixture distribution about the overall mean $\boldsymbol{\mu} = ((1-\eta)a, \ldots (1-\eta)a)^T$.

$$\begin{aligned} TTSS_{\text{mix}} &= \int ||\boldsymbol{x} - \boldsymbol{\mu}||^2 \{\eta f(\boldsymbol{x}) + (1-\eta) g(\boldsymbol{x})\} d\boldsymbol{x} \\ &= \sum_{j=1}^{p} \int_{-\infty}^{\infty} (x_j - (1-\eta)a)^2 \{\eta f(x_j) + (1-\eta) g(x_j)\} dx_j \\ &= p(1-\eta)\eta a^2 + \sum_{j=1}^{p} \lambda_j \end{aligned}$$

The theoretical within cluster sum of squares is given by

$$TWSS_{\text{mix}} = \int_{\boldsymbol{x} \in S_1} ||\boldsymbol{x} - \boldsymbol{\mu_1}||^2 f(\boldsymbol{x}) d\boldsymbol{x} + \int_{\boldsymbol{x} \in S_2} ||\boldsymbol{x} - \boldsymbol{\mu_2}||^2 g(\boldsymbol{x}) d\boldsymbol{x}$$

where S_1 and S_2 are partitions of R^p chosen to minimize $TWSS_{\text{mix}}$, $\boldsymbol{\mu_1} = \int_{\boldsymbol{x} \in S_1} f(\boldsymbol{x}) d\boldsymbol{x}$, and $\boldsymbol{\mu_2} = \int_{\boldsymbol{x} \in S_2} g(\boldsymbol{x}) d\boldsymbol{x}$. Replace x_i with $y_i + aI(x_i \in C_2)$ where $I(x_i \in C_2) = 1$ if x_i is in cluster 2, $I(x_i \in C_2) = 0$ otherwise, and $y_1, \ldots y_n$ are from $N(\mathbf{0}, \mathbf{D})$. $\boldsymbol{D}$ is a known diagonal matrix with elements $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$. Let $h(\mathbf{y})$ represent the density of $\mathbf{y}$.

Then

$$TWSS_{\text{mix}} = TWSS^*_{\text{mix}} = \int_{\boldsymbol{y} \in S_1^*} ||\boldsymbol{y} - \boldsymbol{\mu_1^*}||^2 h(\boldsymbol{y}) d\boldsymbol{y} + \int_{\boldsymbol{y} \in S_2^*} ||\boldsymbol{y} - \boldsymbol{\mu_2^*}||^2 h(\boldsymbol{y}) d\boldsymbol{y}$$

where S_1^* and S_2^* are partitions of R^p chosen to minimize $TWSS^*_{\text{mix}}$, $\boldsymbol{\mu_1^*} = \int_{\boldsymbol{y} \in S_1^*} h(\boldsymbol{y}) d\boldsymbol{y}$, and $\boldsymbol{\mu_2^*} = \int_{\boldsymbol{y} \in S_2^*} h(\boldsymbol{y}) d\boldsymbol{y} - \boldsymbol{a}$.

Since the greatest variation is in the first feature, our separating plane will be the same as that for the un-clustered scenario, which is the plane through $(0, \ldots, 0)^T$ and orthogonal to $(1, \ldots, 0)^T$. Again we have $\mu_{12} =, ..., = \mu_{1p} = \mu_{22} =, ..., = \mu_{2p} = 0$ and $\mu_{11} = \sqrt{\frac{2\lambda_1}{\pi}} = -\mu_{21}$. Thus we have that $TWSS_{\text{mix}} = \sum_{j=1}^{p} \lambda_j - \frac{2\lambda_1}{\pi}$. Therefore

$$TCI_{\text{mix}} = \frac{\pi \sum_{j=1}^{p} \lambda_j - 2\lambda_1}{p(1-\eta)\eta a^2 + \sum_{j=1}^{p} \lambda_j} < \frac{\pi \sum_{j=1}^{p} \lambda_j + \pi\eta(1-\eta)a^2 p - 2\lambda_1 - 2\eta(1-\eta)a^2}{p\eta(1-\eta)a^2 + \sum_{j=1}^{p} \lambda_j} = TCI_{\text{null}}$$

A4. Proof of Theorem 4

To prove that the p-value for the UNPaC test converges in probability to 0 as $p \to \infty$ we employ the same strategy as Liu et al. (2008) by showing the following:

1. The cluster index from the observed clustered data X converges to 0 in probability as $p \to \infty$
2. The cluster index from the reference data is bounded away from 0 as $p \to \infty$.

We first assume that the clustering operation partitions the data such that the CI is minimized. Part 1 of the proof follows directly from the proof of Theorem 1 given in Liu et al. (2008). For part 2, we follow a similar strategy as Liu et al. (2008) and use the HDLSS geometry of Hall et al. (2005). Let $z_1, \ldots, z_p$ represent a sample from the reference distribution. Let $\phi(u)$ represent the standard normal density. Define X^s as X with scaled and centered features. In order to use this geometry, we need to ensure that three assumptions are met:

(a) The fourth moments of the data vectors are uniformly bounded.

(b) For a constant σ^2, $\lim_{p\to\infty} \frac{1}{p} \sum_{k=1}^{p} \text{var}(z_k) = \sigma^2$ for features $k = 1, ..., p$.

(c) The random vector is ρ mixing for functions that are dominated by quadratics.

Assumption b): First, note that the variance for feature k from the mixture distribution is given by $\lambda_k + \eta(1-\eta)a^2$. Since the variance of the observed data is known and the reference distribution preserves the covariance structure of the observed data:

$$\lim_{p\to\infty} \frac{1}{p} \sum_{k=1}^{p} \text{var}(z_k) = \lim_{p\to\infty} \frac{1}{p} \sum_{k=1}^{p} \{\lambda_k + \eta(1-\eta)a^2\}/p = \eta(1-\eta)a^2 = \sigma^2$$

Assumption a): Since n is finite, $\max_j(\frac{1}{n}\sum_{i=1}^{n} X_{ij}^{s\,4}) < C$ for a fixed constant $C > 0$. Since our procedure involves determining the unimodal Gaussian KDE for each feature of

X^s and then multiplying by the square root of the observed variance for X, the numerical 4th moment for the Gaussian KDE of the jth feature is given by:

$$\kappa_{j4}(t) = \int_{-\infty}^{\infty} t^4 \frac{\sqrt{1+h_{1j}^2}}{h_{1j}\sqrt{\lambda_j+\eta(1-\eta)a^2}} \frac{1}{n}\sum_{i=1}^{n} \phi\left(\frac{t\sqrt{1+h_{1j}^2} - x_{ij}\sqrt{\lambda_j+\eta(1-\eta)a^2}}{h_{1j}\sqrt{\lambda_j+\eta(1-\eta)a^2}}\right) dt$$

Use the change of variable transformation $u = \frac{t\sqrt{1+h_{1j}^2} - x_{ij}\sqrt{\lambda_j+\eta(1-\eta)a^2}}{h_{1j}\sqrt{\lambda_j+\eta(1-\eta)a^2}}$ and let $\kappa_m(x)$ represent the mth moment for the standard Gaussian distribution. Note: for the Gaussian kernel, $\kappa_1(x) = 0, \kappa_2(x) = 1, \kappa_3(x) = 0, \kappa_4(x) = 3$. Also, since the data has been scaled and centered, $\frac{1}{n}\sum_{i=1}^{n} {x_{ij}^s}^2$=1. Thus:

$$\kappa_{j4}(t) = \frac{(\lambda_j+\eta(1-\eta)a^2)^2}{n(1+h_{1j})^2} \sum_{i=1}^{n} \int_{-\infty}^{\infty} (x_{ij}^s + uh_{1j})^4 \phi(u) du$$

$$= \frac{(\lambda_j+\eta(1-\eta)a^2)^2}{n(1+h_{1j})^2} \sum_{i=1}^{n} ({x_{ij}^s}^4 + 4{x_{ij}^s}^3 h_{1j}\kappa_1(x) + 6{x_{ij}^s}^2 (h_{1j})^2\kappa_2(x) + 4x_{ij}^s (h_{1j})^3\kappa_3(x) + h_{1j}^4\kappa_4(x))$$

$$= \frac{(\lambda_j+\eta(1-\eta)a^2)^2}{(1+h_{1j})^2} \left\{ \frac{1}{n}\sum_{i=1}^{n} {x_{ij}^s}^4 + 6h_{1j}^2 + 3h_{1j}^4 \right\}$$

So $\max_i(\kappa_{i4}(t)) < M^2(C + 6L^2 + 3L^4)$.

Thus we can conclude that the fourth moments of all entries of z are bounded uniformly. Since the entries of z are independent, we also have that assumption c) is met. Therefore the HDLSS geometry from Hall et al. (2005) for data vectors obtained by truncating an infinite time series holds, and as $p \to \infty$, $\frac{1}{p^{1/2}}||z_j - z_l|| \to (2\sigma^2)^{1/2}$, where the convergence is in probability.

The proof that the CI for the reference data, z, converges away from 0 is similar to the proof of 2) for Theorem 1 in Liu et al. (2008). For the CI under the null hypothesis,

$$\text{CI} = \frac{\sum_{k=1}^{2}\sum_{j\in C_k}||z_j - \bar{z}^{(k)}||^2}{\sum_{j=1}^{n}||z_j - \bar{z}||^2} \geq \frac{\frac{1}{2}([n_1/2] + [n_2/2])||z_j - z_l||^2}{\frac{(n-1)^2}{n}||z_j - z_l||^2}$$
$$\rightarrow \frac{\frac{1}{2}([n_1/2] + [n_2/2])2\sigma^2 p}{\frac{(n-1)^2}{n}2\sigma^2 p} = \frac{n([n_1/2] + [n_2/2])}{2(n-1)^2}$$

Where [u] denotes the largest integer smaller than u. See Liu et al. (2008) for more details. The convergence is in probability. Since n is fixed, we can conclude that the CI under the null hypothesis is bounded away from 0 as $p \rightarrow \infty$. The desired result follows.

2.10 Extension to Hierarchical Clustering

One strength of the UNPaC method is the fact that it can be applied to a variety of clustering methods. All that is required is a cluster identity for each observation and a distance measure. Given this information, the proposed method is able to detect if the observations are clustered more closely together than would be expected under a unimodal distribution. Thus, the method can also be used to test the significance of clusters identified via hierarchical clustering. It should also be noted that the L_2 distance used in the cluster index could easily be replaced with the L_1 distance or other distance measures in order to accommodate different data structures and assumptions.

To implement our proposed method for hierarchical cluster significance testing, simply apply hierarchical clustering methods to generate cluster labels for both the observed and reference data sets. For the hierarchical clustering simulation described below, the Euclidean distance matrix is calculated and either single linkage or Ward's minimum variance hierarchical clustering methods (Ward, 1963) were applied to the distance matrix. The resulting tree was cut to produce two clusters.

S1. Hierarchical Clustering Data Set Simulations

A simulation study was conducted to compare UNPaC to existing methods for evaluating the significance of clusters identified using hierarchical clustering. We present the results from the IGP method of Kapp and Tibshirani (2006) and the BFS method of Maitra et al. (2012) to show how naively applying a means-based clustering approach to clusters produced by hierarchical clustering can produce incorrect results. We also present the results of the MPC method of Ahmed and Walther (2012). An extension of SigClust for hierarchical setting has been proposed (https://arxiv.org/pdf/1411.5259.pdf), and we will refer to this method as HSigClust. Since this method has not yet been published, we will compare the results of this method (implemented using R code located at https://github.com/pkimes/sigclust2) to the published SigClust method (Liu et al., 2008) that uses k-means clustering.

A null example containing 500 observation with 75 features was generated as follows:

$$X_{i,j} = \begin{cases} -2 + 5U_{i,j} & \text{if } j \in \{2, 4, \ldots, 24\} \\ 5 + 5U_{i,j} & \text{if } j \in \{1, 3, \ldots, 25\} \\ \epsilon_{i,j} & \text{otherwise} \end{cases}$$

Here the $U_{i,j}$'s are iid Uniform$(0, 1)$ and the $\epsilon_{i,j}$'s are iid $N(0, 1)$. Putative (spurious) clusters were identified using Ward's minimum variance method (Ward, 1963).

A clustered example containing 1200 observation with 75 features was simulated as follows:

$$X_{i,j} = \begin{cases} -2I(i \leq 500) + 5\sin(\theta_i + \pi I(i > 500)) + \epsilon_i & \text{if } j \in \{2, 4, \ldots, 24\} \\ 5I(i \leq 500) + 5\cos(\theta_i + \pi I(i > 500)) + \epsilon_i & \text{if } j \in \{1, 3, \ldots, 25\} \\ \gamma_{i,j} & \text{otherwise} \end{cases}$$

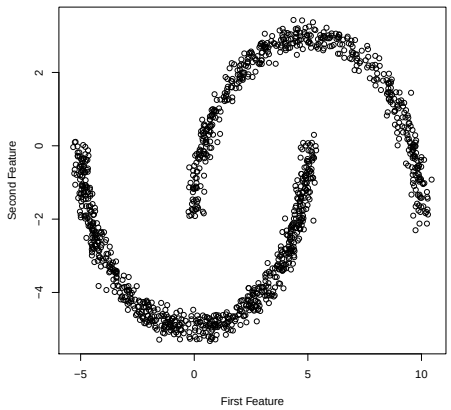

Figure 2.11: *Hierarchical simulation example*. Plot of the second feature versus the first feature for a single simulation from the clustered hierarchical simulation scenario. Note that the data has two non-spherical clusters.

Here the ϵ_i's are iid $N(0, 0.2)$, the θ_i's are iid Uniform$(0, \pi)$, and the $\gamma_{i,j}$'s are iid $N(0, 1)$. Figure 2.11 gives a visual representation of the cluster structure for one iteration of the simulation. Clusters were identified using single linkage.

Both the null and clustered examples were simulated 50 times. The results (shown in Table 2.7) demonstrate that the UNPaC method has good performance compared to the existing methods, although the null situation was challenging for all methods. Interestingly, the SigClust method using sample covariance estimation outperformed HSigClust in the null example. For the clustered example, all methods did well except for SigClust using sample covariance estimation. Since all of the methods except UNPaC, HSigClust, and MPC use means-based clustering approaches for the null data, it is not surprising that they give significant results.

Table 2.7: *Comparison of cluster detection accuracy for the hierarchical clustering examples.* The number of times each method gave a p-value < 0.05 for 50 simulations is recorded

Simulation Name	Number of Simulations with p-value < 0.05							
	UNPaC	SigClust 1	SigClust 2	SigClust 3	HSigClust	IGP	BFS	MPC
Null	12	50	15	50	46	46	46	50
Two clusters	50	50	0	50	49	50	50	50

SigClust 1, SigClust 2, and SigClust 3 represent SigClust implemented using soft-thresholding (Huang et al., 2015), sample covariance estimation, and hard-thresholding (Liu et al., 2008), respectively. HSigClust=the extension of SigClust for hierarchical clustering. IGP= In Group Proportion (Kapp and Tibshirani, 2006). MPC= Multimodality of Principal Curves (Ahmed and Walther, 2012).

CHAPTER 3: BICLUSTERING USING SPARSE CLUSTERING AND SIGNIFICANCE TESTING

3.1 Introduction

A related problem to clustering data observations is identifying distinguishing features for cluster membership. For instance, in understanding the etiology of a disease not only is it important to understand if there are subgroups present within the population, both individuals with the disease and currently disease-free individuals who may develop the disease, it is also important to determine in what way the subgroups differ, be it in the potential causes or in the expression of the disease. Identifying important features for disease subtypes could allow for more targeted preventative or treatment measures.

When subgroups are formed by differences in only a subset of features, biclustering techniques may be especially useful. Biclustering works by identifying a submatrix within the data such that the pattern of the features for the observations within the submatrix are different than the pattern of features outside of the submatrix. This is different than clustering which partitions the observations based off patterns in all of the features. However, once distinguishing features have been identified, the problem of biclustering simply becomes a clustering problem.

Witten and Tibshirani (2010) proposed a sparse clustering method based on maximizing a weighted version of the between cluster sum of squares. The produced weights can be viewed as the contribution of the features to the overall clustering. By performing clustering on the features with non-zero weights we can identify biclusters in the data. In this paper we show how this extension of the sparse clustering approach can be used to detect biclusters with heterogeneous means and more complicated structures identified through hierarchical

clustering. Since correlation in the data may lead to the identification of spurious biclusters we also incorporate a cluster significance test in our method. We present details for the proposed method, compare the proposed method to existing methods in extensive simulation studies, and apply the method to identify biclusters in real data sets.

3.2 Methods

3.2.1 Sparse Clustering

In a data set with many features, it might be expected that only a subset of those features are responsible for clustering. The sparse clustering method of Witten and Tibshirani (2010) capitalizes upon this assumption by using a lasso-type penalty to adaptively select features and then clustering is performed on those subset of features.

For a $n \times p$ data set $\boldsymbol{X}$ with n observations and p features the general sparse clustering is the solution to the problem

$$\underset{\boldsymbol{w};\boldsymbol{\Theta}\in D}{maximize}\{\sum_{j=1}^{p} w_j f_j(\boldsymbol{X_j}, \boldsymbol{\Theta})\} \text{ subject to } ||w||^2 \leq 1, ||w||_1 \leq s, w_j \geq 0 \; \forall j \tag{3.31}$$

where $f_j(\boldsymbol{X_j}, \boldsymbol{\Theta})$ is some function that involves only the jth feature of the data; $\boldsymbol{\Theta}$ is a parameter, often the clustering indices, restricted to lie in a set D; and w_j is a weight corresponding to the jth feature. When Θ is held fixed the solution to the weights can be solved by soft-thresholding as follows:

$$\boldsymbol{w} = \frac{S(\boldsymbol{a}_+, \Delta)}{||S(\boldsymbol{a}_+, \Delta)||_2} \tag{3.32}$$

Where a_+ denotes the positive part of a and $a_j = f_j(\boldsymbol{X_j}, \boldsymbol{\Theta})$. S is the soft-thresholding operator, $S(x, c) = sign(x)(|x| - c)_+$. $\Delta = 0$ if that leads to $||w||_1 \leq s$, otherwise $\Delta > 0$ is chosen such that $||w||_1 = s$ where s is a specified tuning parameter.

For application to k-means clustering the sparse clustering method maximizes a weighted version of the between cluster sum of squares. Given K clusters in the data, the between cluster sum of squares for feature j ($BCSS_j$) is defined to be

$$BCSS_j = \sum_{j=1}^{p}(\frac{1}{n}\sum_{i=1}^{n}\sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{K}\frac{1}{n_k}\sum_{i,i'\in C_k} d_{i,i',j}) \tag{3.33}$$

Where $d_{i,i',j} = (X_{ij} - X_{i'j})^2$; n_k is the number of observations in cluster k; and C_k is the set of indices of observations belonging to cluster k. The sparse K-means clustering criterion is simply the solution to equation 3.31 with $f_j(\boldsymbol{X_j}, \boldsymbol{\Theta}) = BCSS_j$.

Sparse hierarchical clustering is implemented by conducting hierarchical clustering on a weighted dissimilarity matrix. Let $\boldsymbol{U}$ be the overall dissimilarity matrix $\{\sum_j d_{i,i',j}\}_{i,i'}$. Then sparse hierarchical clustering simply reduces to equation 3.31 with $f_j(\boldsymbol{X_j}, \boldsymbol{\Theta}) = \sum_{i,i'} d_{i,i',j}U_{i,i'}$ and the additional constraint that $\sum_{i,i'} U^2_{i,i'} \leq 1$.

To determine the optimal tuning parameter, s, a permutation approach related to the gap statistic of Tibshirani et al. (2001) is applied. The number of clusters must be pre-specified before the algorithm is initiated.

3.2.2 Biclustering Via Sparse Clustering

Since the goal of biclustering is to identify a submatrix U of the data, X, such that the observations in U have a different feature pattern than the rest of the observations in X, one way of identifying U is to perform 2-means sparse clustering on X. The observations in the smaller cluster and the features with non-zero weights could then be considered as the observations and features, respectively, of the bicluster U.

To select the features in the bicluster we first fix the tuning parameter to be $s = \sqrt{p}$, resulting in no soft-thresholding of weights. Then features with weights larger than expected null weights are chosen to be the features in the biclusters. Specifically let $w_{(1)}, w_{(2)}, \ldots w_{(p)}$ denote the ordered weights produced by the sparse clustering procedure

and $w_{(1)_0}, w_{(2)_0}, \ldots w_{(p)_0}$ denote the ordered null weights. If there are no biclusters we would expect $w_{(j)} \equiv w_{(j)_0}$ for all features j. However if m features form the bicluster then we would expect $w_{(j)} > w_{(j)_0}$ for $j > m$ and $w_{(j)} < w_{(j)_0}$ for $j < m$. We make use of this fact for selecting features in the bicluster.

The proposed biclustering method, henceforth referred to as SCBiclust, can be summarized as follows.

1. Apply a modified version of the Witten and Tibshirani (2010) 2-means sparse biclustering algorithm such that we maximize:

$$\text{maximize}_{C_1,C_2,\mathbf{w}} \left\{ \sum_{j=1}^{p} w_j \sqrt{\left(\frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{2} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j} \right)} \right\} \quad (3.34)$$
$$\text{subject to } ||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq \sqrt{p}, w_j \geq 0 \, \forall \, j.$$

2. Compare the distribution of w_j's to the distribution of $w_{(j)_0}$'s. Specifically let:

$$m = \arg\max_j \left\{ (w_{(p-j+1)} - w_{(p-j+1)_0}) - (w_{(p-j)} - w_{(p-j)_0)}) \right\} \quad (3.35)$$

where the $w_{(k)}$ is the kth ordered weight produced from the sparse clustering in Step 1 and $w_{(k)_0}$ is the kth ordered null weight.

3. Use a cluster significance test to assess the strength of the clusters (C_1, C_2) identified in step 1. Here we use the SigClust test (Liu et al., 2008).

4. If the test in Step 3 rejects the null hypothesis return then the resulting bicluster consisting of the m features with largest weights and the observations that belong to the smallest cluster.

5. To identify additional biclusters define a matrix X' with entries x_{ij} $i = 1, \ldots, n$ $j = 1, \ldots, p$ as follows:

$$x'_{ij} = \begin{cases} x_{ij} & \text{if } x_{ij} \notin U_1 \\ x_{ij} - \bar{X}_{U_1,j} + \bar{X}_{U'_1,j} & \text{if } x_{ij} \in U_1 \end{cases} \tag{3.36}$$

where U_1 is the first identified bicluster. Then iteratively apply steps 1 - 4 to the matrix X' to identify multiple biclusters in the same data set. Terminate this procedure if the null hypothesis is rejected in Step 3 or if no feature weights exceed the given cutoff.

Note that the 2-means sparse biclustering algorithm in step 1 can easily be replaced with a sparse hierarchical clustering step. A simulation example is given in section 3.3.1 to illustrate this functionality.

3.2.2.1 Null Feature weights

To generate reference feature weights for bicluster identification two approaches can be used. The first approach assumes that the features are uncorrelated and makes use of distributional assumptions. The second approach relaxes this assumption and generates the features by applying sparse clustering to a null unimodal data set.

For the first approach let b_j represent the between cluster sum of squares for feature j. When $s = \sqrt{p}$ the optimal weights which maximize 3.34 are given by $w_j = \frac{\sqrt{b_j}}{\sqrt{\sum_k b_k^2}}$. When no clusters are present in the data, there is no difference in the means between the two clusters which implies $b_j \sim \chi_1^2$ for all j. If the b_j's are independent then we have that $w_j^2 \sim Beta(\frac{1}{2}, (p-1)/2)$. Thus to identify the distinguishing features we can compare the distribution of w_j to w_{j_0} generated from the numerical approximation to $E(\sqrt{B})$ where $B \sim Beta(\frac{1}{2}, (p-1)/2)$

Since it cannot always be assumed that the between between cluster sum of squares are independent, we also present another way to estimate the null weights. This method

generates reference data from a null unimodal distribution and then applies the sparse clustering algorithm to the null data producing null weights. The procedure is as follows

1. Estimate the covariance structure of the data. When $p > n$ we use the graphical lasso (Friedman et al., 2008) with sparsity parameter $\rho = 0.02$.

2. Generate multivariate unimodal reference distributions, X^0, B times (we use B=100 in this paper). For a given iteration, b, with $b = 1, \ldots, B$ do the following:

   (a) For each feature, j, find the smallest bandwidth estimator, h_{1j} such that the Gaussian KDE for that feature has one mode.

   (b) For each feature generate null data $X^0_{j_b}$ by

   $$X^0_{ij_b} = (1 + h^2_{1j}/\sigma^2_j)^{-1/2}(X_{I(ij)} + h_{1j}\epsilon_{ijb}) \tag{3.37}$$

   where $\epsilon_{ijb} \sim N(0, 1)$; σ^2 is the sample variance for feature j; and $X_{I(ij)}$ are sampled uniformly, with replacement, from the observed data for feature j.

   (c) Multiply $\{X^0_{1_b}, \ldots, X^0_{p_b}\}$ by the Cholesky root of estimated covariance matrix to generate X^0_b.

   (d) Cluster X^0_b using the sparse clustering method given in 3.34.

   (e) For each simulation record the ordered feature weights $w_{(1)b_0}, \ldots w_{(p)b_0}$

3. The estimated null weight for feature j is then given by $w_{(j)_0} = \sum_{b=1}^{B} w_{(j)b_0}/B$.

Note the beta-based method for generating null feature weights results in a substantial decrease in computation time. We first present simulation results using the beta-based weights method in 3.3.1 and then compare the the beta-based weights to the null distribution based weights in 3.3.2

3.2.3 Existing Biclustering Methods

Several strategies have been proposed to tackle the problem of identifying biclusters. One general strategy involves the use of mixture models to identify biclusters within the data. Both the Plaid algorithm of Lazzeroni and Owen (2002) and the LAS algorithm of Shabalin et al. (2009) use this strategy to identify biclusters. The sparse biclustering method of Tan and Witten (2014) frames biclustering as a penalized maximum likelihood estimation problem assuming each data entry is independent and normally distributed. Another strategy uses singular value decomposition (SVD) to find signals within the data which can be represented as a biclusters. The SSVD method of Lee et al. (2010) uses a penalized version of SVD to identify biclusters and the HSSVD method of Chen et al. (2013) expands upon this method by capturing the variance structure of a data set in addition to the mean structure.

3.3 Simulation Studies

3.3.1 Comparison of SCBiclust to Existing Methods

To test the accuracy of our proposed method using the beta-based null weights a simulation study was conducted and results were compared to the existing biclustering methods described above. For each simulation scenario 100 data sets were generated with the described data structure. The proposed method utilizes the SigClust R package and a modified version of the "sparcl" R package. The Plaid algorithm within the R package "biclust" was used with default settings after scaling across features. This package implements the recent advances to the Plaid algorithm proposed by Turner et al. (2003). The data transformation step for the LAS algorithm, available at https://genome.unc.edu/las/, was used if recommended by the method. For both HSSVD and SSVD the resulting singular vectors were first transformed for visualization such that positive values were given a value of 1 and negative values were given a value of -1. Values of zero remained the same. For comparison of accuracy the singular vectors were dichotomized to 0 or 1 since it was

of interest only to identify the submatrix responsible for the bicluster. Documentation on SSVD is available at http://www.unc.edu/∼haipeng/. HSSVD software is available at http://impact.unc.edu/impact7/HSSVD. The "sparseBC" R package was used for the sparse biclustering algorithm. We defined a valid bicluster to be a bicluster which consisted of at least two observations and two features. The number of valid biclusters was reported for each method and each simulation study, but only valid biclusters were used for calculating average accuracy.

To evaluate the reproducibility of the biclusters identified by each method the observations from the original data set X were equally and randomly split into two submatrices, X_1 and X_2. Let U be the primary bicluster identified in X and U_1 and U_2 be the primary biclusters identified within X_1 and X_2, respectively. We consider U as the "correct" bicluster. The observation misclassification rate (OMR) was calculated as the percentage of observations that were either in U_1 or U_2, but not in U or in U but not in U_1 or U_2. The percentage of false negatives (FNR) was calculated as the percentage of features in U that were not in U_1 or U_2. The percentage of false positives (FPR) was calculated as the percentage of features in U_1 or U_2 that were not in U. The feature misclassification rate (FMR) was calculated as the percentage of features that were identified as important in U_1 but not U_2, or identified in U_2, but not in U_1. This procedure was repeated 10 times for each simulation in Simulations 1.1, 1.2, and 5.

In simulations where there was a primary bicluster (Simulations 1.1, 1.2, and 5) we look at each method's ability to identify the observations and features belonging to that specific bicluster. The observation misclassification rate, feature false negative rate (FNR), and feature false positive rate (FPR) were calculated for each valid primary bicluster identified by each method. In situations where there was no primary bicluster (Simulations 3 and 4) the number of times each method identified bicluster 1, bicluster 2, or a larger bicluster that covered both bicluster 1 and 2 (which will be referred to as bicluster 1+2) was recorded.

Also, instead of recording the classification errors for observations and features separately the false positive and false negative rates of the entries in total were recorded.

We also examined the number of bicluster identified by each method for simulations 1.1, 1.2, and 4. Note SCBiclust will terminate if either none of the features weights exceed the expected weights or the cluster significance test does not reject the null hypothesis. The maximum number of biclusters was set to 7.

The computing time recorded in the results is for implementing the method for a set number of biclusters (either one bicluster for primary bicluster identification, or two biclusters for overlapping bicluster identification). In these scenarios the cluster significance step was not used.

3.3.1.1 Simulation 1 Primary Bicluster Identification

In this example, a 100 observation $\times$ 200 feature matrix data set, X, was comprised of one primary bicluster and three additional non-overlapping normally distributed biclusters. Each observation was independent. We first examine the situation when the features are independent (this will be referred to as Simulation 1.1).

The background entries followed a $N(0, 1)$ distribution, where $N(a, b)$ represents a normal random variable with mean a and standard deviation b. The four non-overlapping rectangular shaped biclusters were constructed in the following manner: bicluster 1, consisting of observations 1-20 and features 1-20 (denoted as [1-20, 1-20]) added a $N(2, 1)$ layer to the background, bicluster 2 [16-30, 51-80] added a $N(3, 1)$ layer to the background, bicluster 3 [51-90, 61-130] added a $N(3, 1)$ layer to the background, and bicluster 4 [66-100, 151-200] added a $N(2, 1)$ layer to the background. Bicluster 3 was the primary bicluster, since it was the largest bicluster and had the largest mean difference from the background, so we expected the algorithms to detect this bicluster as the first layer. Figure 3.12 shows the biclustering results from one of the simulations.

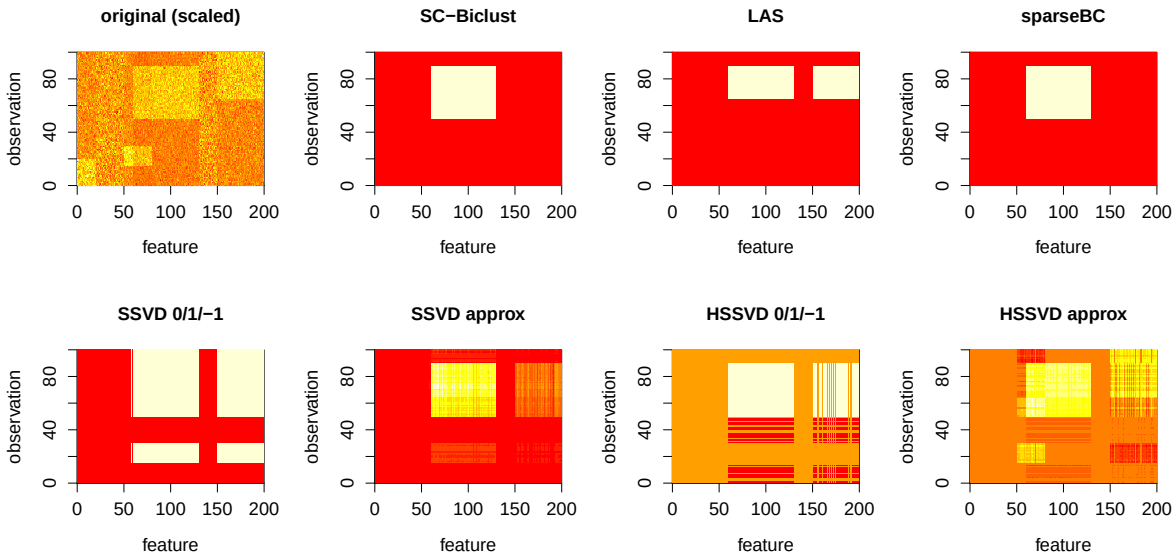

Figure 3.12: *Simulation 1.1 example: primary bicluster identification.* This is an illustration of a single simulated data set from simulation 1.1. The first panel shows a heat map of the (scaled) data. The primary bicluster is the rectangular yellow block in the middle. The remaining panels show the biclusters identified by SCBiclust, LAS, sparse biclustering, SSVD, and HSSVD, with the white regions corresponding to the biclusters. For SSVD and HSSVD, both the 0/1/-1 indicator matrix and the approximation matrix are plotted.

As an extension to this simulation, the first 20 features were generated under a correlated structure. We will denote this simulation as "Simulation 1.2." Specifically, the background structure consisted of

$$X[1-20, 1-20] \sim N[(\overbrace{2, \ldots, 2}^{20})^T, \Sigma]$$

$$X[21-100, 1-20] \sim N[(\overbrace{0, \ldots, 0}^{20})^T, \Sigma]$$

$$X_{ij} \sim N(0,1) \;\; otherwise$$

Where $[\Sigma]_{ii} = 1, [\Sigma]_{ii'} = 0.30$.

Biclusters 2, 3, and 4 were constructed as in Simulation 1.1. Figure 3.13 illustrates the biclustering results from one of the simulations.

The goal of this simulation study was to compare the accuracy of the proposed method to the existing methods at identifying the primary bicluster. In 100 simulations of each example, the Plaid algorithm failed to identify any biclusters. Each simulated data set for simulations

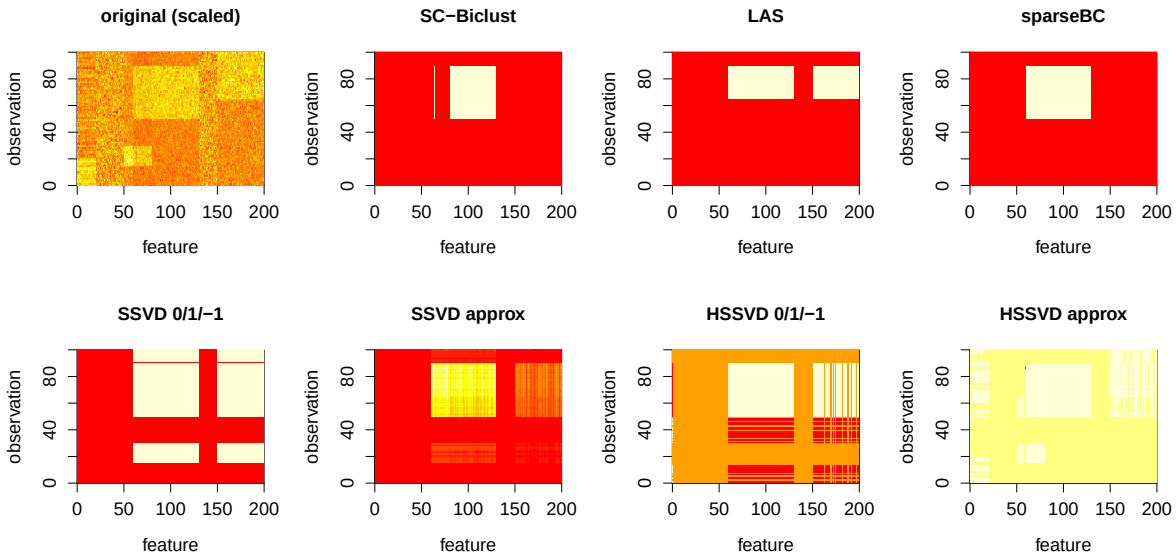

Figure 3.13: *Simulation 1.2 example: primary bicluster identification with correlated features.* This is an illustration of a single simulated data set from simulation 1.2. The first panel shows a heat map of the (scaled) data. The primary bicluster is the rectangular yellow block in the middle. The remaining panels show the biclusters identified by SCBiclust, LAS, sparse biclustering, SSVD, and HSSVD, with the white regions corresponding to the biclusters. For SSVD and HSSVD, both the 0/1/-1 indicator matrix and the approximation matrix are plotted.

1.1 and 1.2 was partitioned as described previously to evaluate the reproducibility of the biclusters.

3.3.1.2 Simulation 2: Departure from Normality

This simulation was similar to Simulation 1, but the data was generated from Cauchy distributions with infinite moments. Specifically, the background entries followed a a Cauchy$(0, 1)$ distribution, where Cauchy(a, b) represents a Cauchy random variable with location shift a and scale b. The non-overlapping biclusters were constructed in the following manner: bicluster 1 [1-20, 1-20] added a Cauchy$(75, 1)$ layer to the background, bicluster 2 [16-30, 51-80] added a Cauchy$(50, 1)$ layer to the background, bicluster 3 [51-90, 71-110] added a Cauchy$(200, 1)$ layer to the background, and bicluster 4 [71-100, 156-200] added a Cauchy$(75, 1)$ layer to the background.

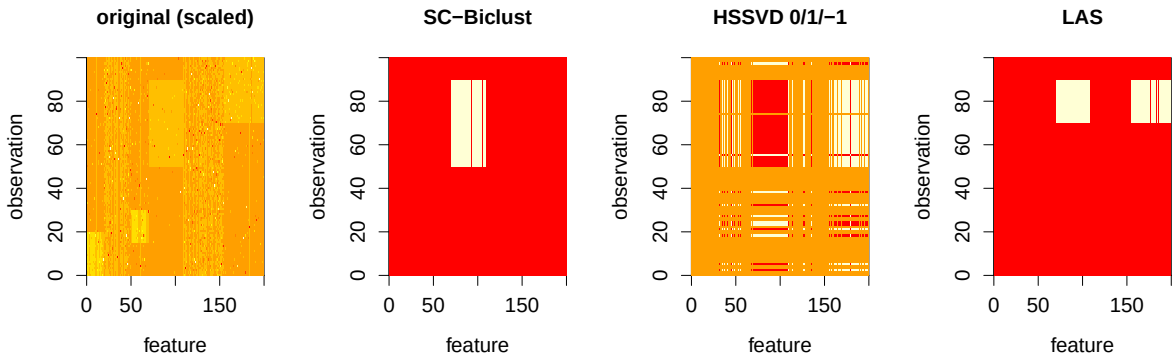

Figure 3.14: *Simulation 2 example: departure from normality.* This is an illustration of a single simulation from the second simulation scenario. The first panel shows a heat map of the (scaled) data. The primary bicluster is the rectangular yellow block in the middle. The remaining panels show the biclusters identified by SCBiclust, HSSVD, and LAS, with the white regions corresponding to the biclusters.

Bicluster 3 was the primary bicluster, and we expected the algorithms to detect this bicluster as the first layer. Each simulated data set was partitioned to evaluate the reproducibility of the biclusters. Figure 3.14 illustrates the biclustering results from one of the simulations.

3.3.1.3 Simulation 3: Two Biclusters with no Overlap

A simulation study was performed on data with two non-overlapping biclusters with feature means that were equal in magnitude, but opposite in direction. Each simulated data set was comprised of a 200×100 matrix with independent entries. The background entries followed a standard normal distribution with mean 0 and standard deviation 1. The two biclusters were constructed as follows: bicluster 1 [1-30, 1-50] added a N(1, 1) layer to the background, bicluster 2 [61:90,1:50] added a N(-1, 1) layer to the background. Under the given data structure, the Plaid algorithm failed to identify any biclusters for all the simulations. Reproducibility of the biclusters was not evaluated for this simulation scenario. Figure 3.15 illustrates the biclustering results from one of the simulations.

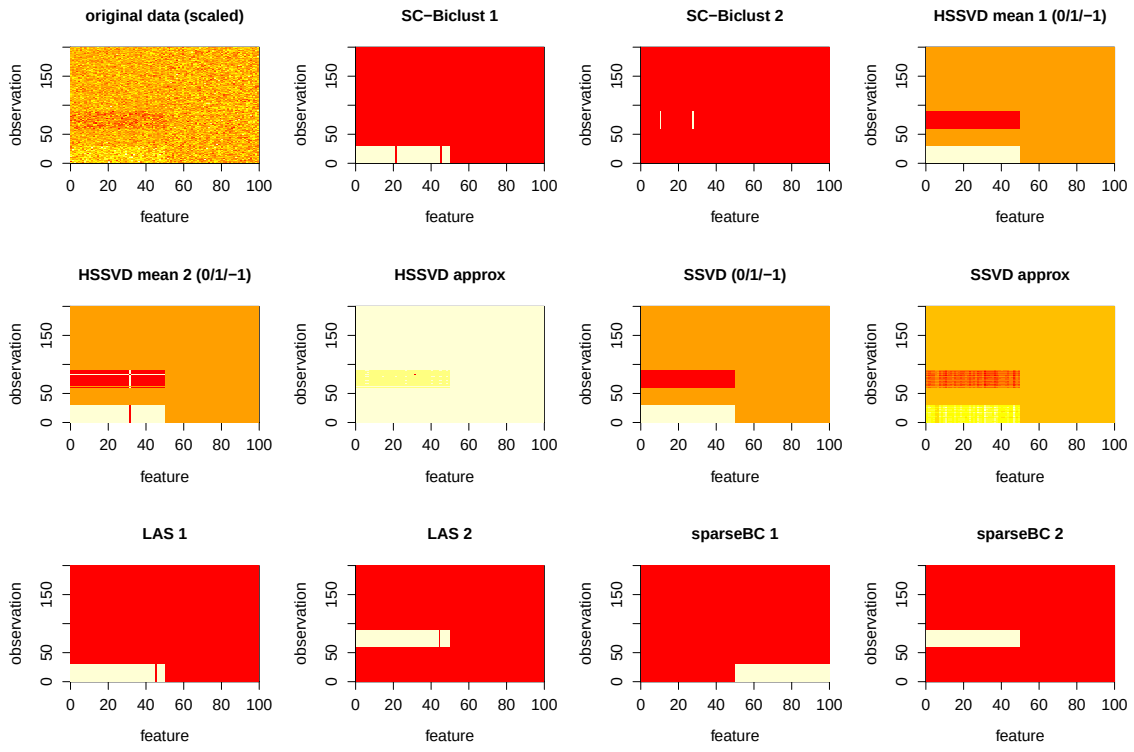

Figure 3.15: *Simulation 3 example: Symmetric Biclusters with no Overlap* This is an illustration of a single simulation from the third simulation scenario. The first panel shows a heat map of the (scaled) data. The two biclusters are in the bottom left corner of the data matrix; one is in red and the other is in yellow. The remaining panels show the first two biclusters identified by SCBiclust, HSSVD, SSVD, LAS, and sparse biclustering. The white regions correspond to the biclusters. For SSVD and HSSVD, both the 0/1/-1 indicator matrix layers and the overall approximation matrices are plotted.

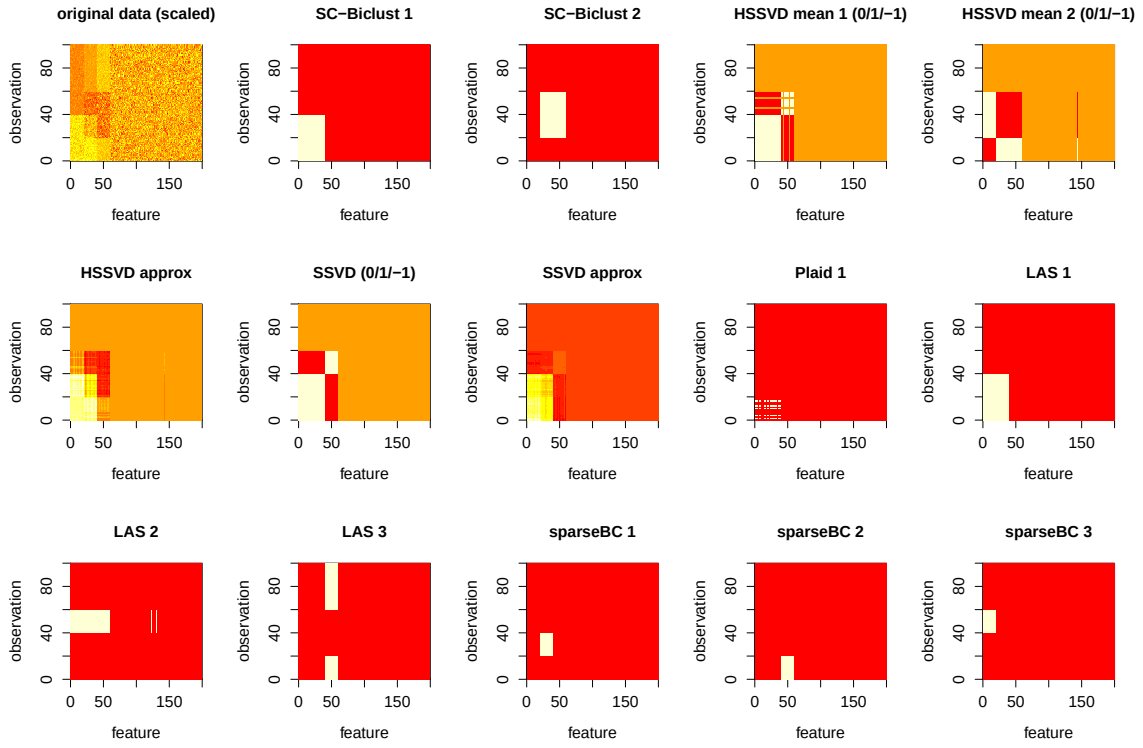

Figure 3.16: *Simulation 4 example: sequential biclusters with overlap.* This is an illustration of a single simulation from the fourth simulation scenario. The first panel shows a heat map of the (scaled) data. The two overlapping biclusters are in the bottom left corner of the data matrix; one is in red and the other is in yellow. The remaining panels show the first two biclusters identified by SCBiclust and HSSVD, the first bicluster identified by SSVD and Plaid, and the first three biclusters identified by LAS and sparse biclustering. The white regions correspond to the biclusters. For SSVD and HSSVD, both the 0/1/-1 indicator matrix layers and the overall approximation matrices are plotted.

3.3.1.4 Simulation 4: Sequential Biclusters with Overlap

We simulated data with two overlapping biclusters. Each data set was comprised of two layers, each of which was a 100×200 matrix with independent entries. The background data were $N(0, 0.5)$. The first layer contained a bicluster [1-40, 1-40] generated from $N(7, 2)$, and the second layer contained a bicluster [21-60, 21-60] generated from $N(-5, 3)$. The final data set was the sum of the two layers. Note that observations 21-40 and features 21-40 are contained in both biclusters. Reproducibility of the biclusters was not evaluated for this simulation scenario. Figure 3.16 illustrates the biclustering results from one of the simulations.

3.3.1.5 Simulation 5: Non-Spherical Biclusters

In this simulation study we provide an example of implementing SCBiclust with single linkage hierarchical clustering. The biclusters in this simulation are non-spherical and thus may not be identified by Euclidean distance based clustering approaches.

Each 1200×75 data set was simulated as follows. For $1 \leq j \leq 25$:

$$X_{i,2j} = -2I(i \leq 500) + 5\sin(\theta_i + \pi I(i > 500)) + \epsilon_i$$
$$X_{i,2j-1} = 5I(i \leq 500) + 5\cos(\theta_i + \pi I(i > 500)) + \epsilon_i$$

Here the ϵ_i's are iid $N(0, 0.2)$ and the θ_i's are iid Uniform$(0, \pi)$. For all $j > 50$, the X_{ij}'s are $N(0, 1)$.

Each simulated data set was partitioned to evaluate the reproducibility of the biclusters. Figure 3.17 illustrates the biclustering results from one of the simulations.

3.3.2 Comparison of Null Weights Methods in SCBiclust

We now repeat the above simulations but just compare SCBiclust implemented using beta-distribution-based null weights to SCBiclust implemented using null weights generated from clustering a unimodal null distribution. We will refer to these two methods as $SCBiclust_\beta$ and $SCBiclust_U$, respectively. For the $SCBiclust_\beta$ method it is recommended that the data be scaled before the biclustering algorithm is applied and for the $SCBiclust_U$ method it is recommended that the data only be centered. We used these general recommendations except for the data generated from the Cauchy distribution (Simulation 2). Since the feature variances were very large we scaled the data before implementing both methods.

The primary bicluster identification accuracy and reproducibility for simulations 1.1,1.2,2, and 5 are given in Table 3.10. We find that, in general, both methods give comparable results

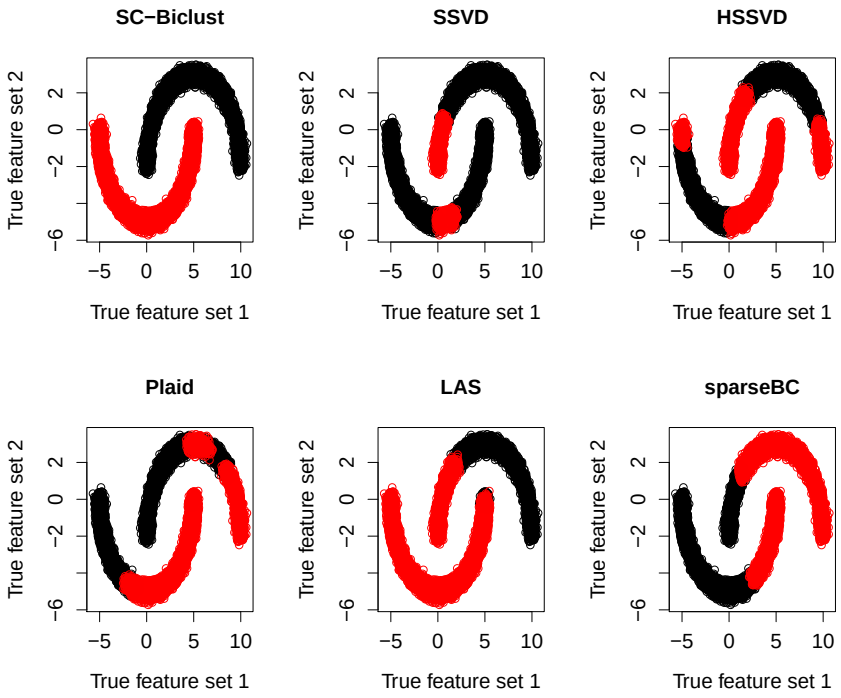

Figure 3.17: *Simulation 5 example: non-spherical biclusters.* Each panel shows a plot of the second feature versus the first feature for a single simulation from the fifth simulation scenario. Note that the data forms two non-spherical clusters. Each panel shows the result of applying a biclustering method (specifically SCBiclust, SSVD, HSSVD, Plaid, LAS, and sparse biclustering) to this data set. Observations that belong to the putative bicluster are labeled in red.

for simulations 1.1,1.2, and 2. In general, $SCBiclust_U$ had a lower feature false negative rate for identification accuracy but a higher feature false negative rate in the reproducibility analysis. $SCBiclust_U$ was both more accurate and more reproducible than $SCBiclust_\beta$ in Simulation 2 when the data entries were Cauchy distributed. In simulation 5 $SCBiclust_U$ had a higher feature false negative rate than $SCBiclust_\beta$. $SCBiclust_U$ may impose a more stringent criteria for feature selection.

The comparison of identification accuracy for simulation 3 and 4 are given in table 3.11. Both of these simulation scenarios included two biclusters. We find that both methods had comparable entry false positive and false negative rates, but $SCBiclust_U$ was more likely to detect the second cluster in simulation 3.

Table 3.8: *Comparison of identification accuracy (average of 100 simulations) and comparison of reproducibility (average of 100 simulations x 10 partitions) for simulations 1, 2, and 5*. OMR=object misclassification rate, FNR= false negative rate, FPR=false positive rate, FMR=feature misclassification rate.

Simulation 1.1: primary bicluster identification									
Algorithm	Valid biclusters	Time	Prediction accuracy			Reproducibility			
			OMR	Feature FNR	Feature FPR	OMR	Feature FNR	Feature FPR	FMR
SCBiclust	100	0.42 sec	0.000	0.150	0.002	0.110	0.180	0.041	0.140
SSVD	100	0.28 sec	0.250	0.000	0.390	0.015	0.012	0.012	0.024
HSSVD	100	1.25 min	0.180	0.000	0.320	0.110	0.320	0.008	0.130
Plaid	0	NA	NA	NA	NA	NA	NA	NA	NA
LAS	100	12.50 sec	0.140	0.002	0.380	0.061	0.150	0.023	0.190
Sparse Biclustering	100	0.85 sec	0.000	0.000	0.004	0.050	0.096	0.088	0.180
Simulation 1.2: primary bicluster identification with correlation between features									
Algorithm	Valid biclusters	Time	Prediction accuracy			Reproducibility			
			OMR	Feature FNR	Feature FPR	OMR	Feature FNR	Feature FPR	FMR
SCBiClust	100	0.45 sec	0.000	0.149	0.002	0.102	0.156	0.047	0.140
SSVD	100	0.69 sec	0.248	0.000	0.389	0.014	0.012	0.013	0.023
HSSVD	100	49.37 sec	0.184	0.001	0.317	0.109	0.316	0.010	0.122
Plaid	0	NA	NA	NA	NA	NA	NA	NA	NA
LAS	100	5.93 sec	0.140	0.000	0.385	0.172	0.183	0.006	0.225
SparseBC	100	0.96 sec	0.000	0.000	0.003	0.050	0.099	0.091	0.186
Simulation 2: departure from normality									
Algorithm	Valid biclusters	Time	Prediction accuracy			Reproducibility			
			OMR	Feature FNR	Feature FPR	OMR	Feature FNR	Feature FPR	FMR
SCBiclust	100	0.42 sec	0.180	0.085	0.050	0.290	0.120	0.093	0.190
SSVD	37	0.62 sec	0.180	0.430	0.072	0.080	0.370	0.041	0.140
HSSVD	100	1.27 min	0.400	0.070	0.530	0.160	0.210	0.240	0.300
Plaid	62	0.08 sec	0.280	0.330	0.130	0.370	0.290	0.150	0.190
LAS	100	1.27 min	0.200	0.017	0.270	0.048	0.210	0.010	0.190
Sparse Biclustering	13	0.99 sec	0.001	0.006	0.004	0.200	0.420	0.003	0.093
Simulation 5: non-spherical biclusters									
Algorithm	Valid biclusters	Time	Prediction accuracy			Reproducibility			
			OMR	Feature FNR	Feature FPR	OMR	Feature FNR	Feature FPR	FMR
SCBiclust	100	4.93 sec	0.058	0.000	0.000	0.073	0.000	0.000	0.000
SSVD	100	37.34 sec	0.410	0.000	0.000	0.011	0.000	0.000	0.000
HSSVD	100	2.36 min	0.450	0.000	0.000	0.270	0.001	0.000	0.001
Plaid	100	0.58 sec	0.290	0.500	0.000	0.770	0.340	0.170	0.320
LAS	100	41.91 sec	0.120	0.000	0.000	0.005	0.000	0.000	0.000
Sparse Biclustering	100	4.62 sec	0.470	0.500	0.000	0.250	0.000	0.000	0.000

Table 3.9: *Comparison of identification accuracy for simulations 3, and 4* (average of 100 simulations). VBCs= Valid Biclusters, FNR=False Negative Rate, FPR=False positive rate, SparseBC=Sparse Biclustering

Simulation 3: Two biclusters no overlap					
Algorithm	VBCs	Time	Identification	Entry FNR	Entry FPR
SCBiclust layer 1	100	0.44 sec	Biclust 1 61%, Biclust 2 39%	0.073	0.002
SCBiclust layer 2			Biclust 1 16%, Biclust 2 18%	0.099	0.007
SSVD	100	0.28 sec	Biclust 1+2 100%	0.002	0.004
HSSVD mean layer 1	84	52.80 sec	Biclust 1 3%, Biclust 2 2%, Biclust 1+2 79%	0.087	0.001
HSSVD mean layer 2			Biclust 1+2 66%	0.032	0.001
Plaid	0	NA	NA	NA	NA
LAS layer 1	100	3.06 sec	Biclust 1 56%, Biclust 2 44%	0.024	0.001
LAS layer 2			Biclust 1 44%, Biclust 2 56%	0.027	0.001
LAS layer 3			Biclust 1 18%, Biclust 2 5%	0.998	< 0.001
SparseBC layer 1	100	28.77 sec	Biclust 1 74%, Biclust 2 26%	0.629	0.155
SparseBC layer 2			Biclust 1 86%, Biclust 2 14%	0.749	0.176
Simulation 4: Sequential biclusters with overlap					
Algorithm	VBCs	Time	Identification	Entry FNR	Entry FPR
SCBiclust layer 1	100	0.79 sec	Bicluster 1 100%	0.000	0.000
SCBiclust layer 2			Bicluster 2 100%	0.000	0.000
SSVD	100	0.39 sec	Bicluster 1+2 100%	0.005	0.000
HSSVD mean layer 1	100	1.28 min	Bicluster 1 26%, Bicluster 1+2 74%	0.088	0.013
HSSVD mean layer 2			Bicluster 1+2 100%	< 0.001	< 0.001
Plaid	98	0.21 sec	Bicluster 1 98%	0.820	< 0.001
LAS layer 1	100	9.54 sec	Bicluster 1 100%	0.022	0.000
LAS layer 2			Bicluster 2 100%	0.500	0.022
LAS layer 3			Bicluster 1 100%	1.000	0.064
SparseBC layer 1	100	23.95 sec	Bicluster 1 85%, Bicluster 2 15%	0.920	0.043
SparseBC layer 2			Bicluster 1 67%, Bicluster 2 33%	0.870	0.026
SparseBC layer 3			Bicluster 1 75%, Bicluster 2 25%	0.910	0.071

3.3.2.1 Simulation Results

Table 3.8 shows the number of valid biclusters, average computing time, identification accuracy, and reproducibility results for simulations 1.1, 1.2, 2, and 5. Table 3.11 shows the number of valid biclusters, average computing time, and identification accuracy for simulations 3 and 4. Table 3.12 gives the number of biclusters identified by each method in simulations 1.1, 1.2, and 4.

SCBiclust performed very well in the first simulation scenario both with and without correlation among the features. No observations were misclassified across all 100 simulations and the feature false positive rate was very low. The feature false negative rate from $SCBiclust_U$ was markedly lower than the feature false negative rate from $SCBiclust_\beta$.

Table 3.10: *Comparison of identification accuracy (average of 100 simulations) and reproducibility (average of 100 simulations x 10 partitions) for SCBiclust methods for simulations 1.1,1.2, 2, and 5*. OMR=object misclassification rate, FNR= false negative rate, FPR=false positive rate, FMR=feature misclassification rate.

Simulation 1.1: primary bicluster identification							
Algorithm	Prediction accuracy			Reproducibility			
	OMR	Feature FNR	Feature FPR	OMR	Feature FNR	Feature FPR	FMR
$SCBiclust_\beta$	0	0.146	0.002	0.036	0.121	0.053	0.142
$SCBiclust_U$	0	0.015	0.002	0.043*	0.195*	0.014*	0.150*
Simulation 1.2: primary bicluster identification with correlation between features							
Algorithm	Prediction accuracy			Reproducibility			
	OMR	Feature FNR	Feature FPR	OMR	Feature FNR	Feature FPR	FMR
$SCBiclust_\beta$	0.000	0.145	0.002	0.018	0.094	0.064	0.139
$SCBiclust_U$	0.000	0.008	0.008	0.041	0.221‡	0.011‡	0.162‡
Simulation 2: departure from normality							
Algorithm	Prediction accuracy			Reproducibility			
	OMR	Feature FNR	Feature FPR	OMR	Feature FNR	Feature FPR	FMR
$SCBiclust_\beta$	0.183	0.085	0.050	0.242	0.317	0.026	0.131
$SCBiclust_U$	0.183	0.079	0.058	0.212	0.212	0.021	0.124
Simulation 5: non-spherical biclusters							
Algorithm	Prediction accuracy			Reproducibility			
	OMR	Feature FNR	Feature FPR	OMR	Feature FNR	Feature FPR	FMR
$SCBiclust_\beta$	0.058	0.000	0	0.073	0	0	0
$SCBiclust_U$	0.000	0.500	0	0.000	0	0	0

* $SCBiclust_U$ was unable to perform cross-validation for 3 data sets.
‡ $SCBiclust_U$ was unable to perform cross-validation for 6 data sets.

The reproducibility of the biclusters identified by SCBiclust was also good. The sparse biclustering method also produced good results, except for the relatively high feature misclassification rate in the reproducibility analysis. SSVD had the best reproducibility, but a large observation misclassification rate in the primary bicluster identification. Plaid did not select any features for this simulation scenario.

The results of the second simulation scenario were similar. Although the accuracy of $SCBiclust_\beta$ was lower when the assumption of normality was violated, it produced a noticeably lower error rate than competing methods (with the exception of Sparse Biclustering). $SCBiclust_U$ had much greater feature identification accuracy than $SCBiclust_\beta$. SCBiclust produced valid biclusters in all 100 simulations whereas SSVD, Plaid, and sparse biclustering frequently failed to identify valid biclusters. Sparse biclustering tended to produce good results when it identified biclusters in the data, but it failed to detect any

Table 3.11: *Comparison of identification accuracy for SCBiclust methods in simulations 3, and 4*(average of 100 simulations). BCs=Biclusters, FNR=False Negative Rate, FPR=False positive rate.

Simulation 3: Two biclusters no overlap				
Algorithm	Valid BCs	Identification	Entry FNR	Entry FPR
$SCBiclust_\beta$ layer 1	95	Biclust 1 56 %, Biclust 2 39%	0.034	0.002
$SCBiclust_\beta$ layer 2		Biclust 1 16 %, Biclust 2 18%	0.099	0.007
$SCBiclust_U$ layer 1	91	Biclust 1 57% Biclust 2 34 %	0.042	0.001
$SCBiclust_U$ layer 2		Biclust 34% Biclust 2 57%	0.051	0.011
Simulation 4: Sequential biclusters with overlap				
Algorithm	Valid BCs	Identification	Entry FNR	Entry FPR
$SCBiclust_\beta$ layer 1	100	Biclust 1 100%	0.000	0.000
$SCBiclust_\beta$ layer 2		Biclust 2 100%	0.000	0.000
$SCBiclust_U$ layer 1	100	Biclust 1 100%	0.003	0.000
$SCBiclust_U$ layer 2		Biclust 2 100%	0.000	0.000

biclusters in 87 of the 100 simulations. LAS identified valid biclusters in all 100 simulations with comparable observation misclassification rate to SCBiclust, but a much greater feature false positive rate.

$SCBiclust_U$ and LAS both had good performance in Simulation 3. They both identified the two biclusters that were present with good accuracy. $SCBiclust_\beta$ tended to only identify one bicluster in the data, but the identification of entries was fairly accurate. SSVD and HSSVD tended to identify bicluster 1+2 (combining the two biclusters into one). Plaid did not identify any valid biclusters for this simulation scenario. The sparse biclustering method split the data into biclusters with very poor accuracy as is evidenced by high false negative and false positive rates.

In the fourth simulation scenario, SCBiclust identified both biclusters with perfect accuracy in all the simulations. LAS also identified the first bicluster with high accuracy but it tended to not include all of the relevant features when identifying the second bicluster. SSVD and HSSVD tended to identify bicluster 1+2 (combining the two biclusters into one). The performance of Plaid was poor. The sparse biclustering method identified single biclusters with very high false negative rates.

$SCBiclust_\beta$ had a much lower proportion of misclassified observations in the fifth simulation scenario and excellent reproducibility. This is not surprising, since the other biclustering methods assume that the biclusters are spherical, and this assumption is violated

for this simulation. $SCBiclust_U$ had a much higher feature false negative rate indicating that it may be too restrictive at selecting features. These results illustrate that $SCBiclust_\beta$ can be used to identify biclusters in situations where existing methods will fail.

In terms of computing time, $SCBiclust_\beta$ was one of the fastest methods with only SSVD and Plaid having comparable running times across all simulations. Note that the time recorded does not take into consideration the computation time for the cluster significance step in $SCBiclust_\beta$. Also, these simulations use HSSVD without pre-specified rank, which increases the computing time of this method.

In simulation 1.1, $SCBiclust_\beta$ with SigClust correctly determined that 4 biclusters were present in the data in 40% of the simulations, but incorrectly identified a 5th bicluster in 60% of simulations. HSSVD mean correctly identified 4 biclusters in 54% of simulations. LAS and sparse biclustering consistently identified too many biclusters. Plaid was not included since it did not return any valid results for this simulation scenario. $SCBiclust_\beta$ correctly identified 4 biclusters in only 8% of the repetitions of Simulation 1.2, but identified 5 biclusters 48% of the time and 6 biclusters 43% of the time. HSSVD mean correctly identified 4 biclusters in 43% of the simulations. The remaining methods had comparable performance as in simulation 1.1. In simulation 4, SCBiclust correctly determined that 2 biclusters were present in the data in 99% of the simulations. HSSVD mean correctly determined that 2 biclusters were present in 69% of simulations, and Plaid determined that 2 biclusters were present in only 24% of simulations. Again, LAS and sparse biclustering always overestimated the number of biclusters. SSVD was not included in this comparison since it always returns a single layer.

Table 3.12: *Stopping rule comparison for simulations 1.1, 1.2, and 4* (average of 100 simulations). Maximum number of biclusters was set to 7

	Simulation 1.1: 4 biclusters are present in the data
Algorithm	Number of biclusters identified (%)
SCBiclust	4 (40%) 5 (60%)
HSSVD mean	2 (10%) 3 (32%) 4 (54%) 5 (4%)
HSSVD var	2 (100%)
LAS	6 (100%)
Sparse Biclustering	6 (99%)
	Simulation 1.2: 4 biclusters are present in the data
Algorithm	Number of biclusters identified (%)
SCBiclust	4 (8%) 5 (48%) 6 (43%)
HSSVD mean	3 (46%) 4 (43%) 5 (11%)
HSSVD var	2 (100%)
LAS	6 (100%)
Sparse Biclustering	6 (100%)
	Simulation 4: 2 biclusters are present in the data
Algorithm	Number of biclusters identified (%)
SCBiclust	2 (99%) 3 (1%)
HSSVD mean	2 (69%) 3 (31%)
HSSVD var	2 (100%)
Plaid	1 (40%) 2 (24%) 3 (14%) 4 (9%) 5(2%)
LAS	6 (100%)
Sparse Biclustering	4 (100%)

3.4 Real Data Application

3.4.1 Analysis of OPPERA data

We next look for biclusters in data collected in the Orofacial Pain Prospective Evaluation and Risk Assessment (OPPERA) study. OPPERA is a prospective cohort study on Temporomandibular Disorders (TMD), which are a set of painful conditions that affect the jaw muscles and/or the jaw joint. Both TMD-free participants and chronic TMD patients were enrolled in the study. Each study participant completed a quarterly questionnaire, and participants who showed signs of first-onset TMD returned to the clinic for a formal examination. The median follow up period was 2.8 years. The data set contains 185 chronic TMD patients and 3258 initially TMD-free individuals, 260 of whom developed TMD by the end of the study. Among the TMD-free individuals, 521 did not complete any follow up questionnaires and were excluded from the analysis. The remaining 2737 were used for

survival analysis where development of first-onset TMD is the event of interest. For a more detailed description of the OPPERA study, see Slade et al. (2011) and Bair et al. (2013).

Three sets of possible risk factors for TMD were measured in OPPERA: autonomic measurements such as blood pressure and heart rate (44 total variables), psychosocial measurements such as depression and anxiety (39 total variables), and quantitative sensory testing (QST) measurements (33 total variables) that evaluate participants' sensitivity to experimental pain. See Fillingim et al. (2011), Greenspan et al. (2011), and Maixner et al. (2011) for more detailed descriptions of these variables.

We implemented the $SCBiclust_\beta$ algorithm with a maximum of 3 biclusters and no cluster significance step. For bicluster identification we included both chronic TMD cases and initially TMD-free individuals. Since individuals who are currently TMD free may exhibit similar characteristics to chronic TMD cases, including chronic TMD cases in the bicluster identification step may produce biclusters of currently TMD-free individuals that exhibit a higher risk of developing first-onset TMD.

The first bicluster contained 30 measures of autonomic function, the second bicluster contained 29 measures of psychological distress, and the third bicluster contained 6 measures of pain sensitivity (Table 3.13). It is interesting to note that the biclusters detected in this paper using unsupervised methods showed a similar structure to the clusters Bair et al. (2016) identified using semi-supervised methods. Specifically, biclusters were determined by patterns in both psychological distress and pain sensitivity which were identified as important contributers to cluster membership in Bair et al. (2016). There was no overlap in the features selected in the three biclusters, but there was overlap in the observations selected for the biclusters. Thus, the biclusters identified by $SCBiclust_\beta$ were consistent with the known structure of the data set.

Plaid and LAS also were able to identify biclusters consisting of features from just one set of possible risk factors for TMD (Table 3.13). The biclusters identified by the other methods did not correspond to the three different types of measurements known to exist in

this data set. SSVD and HSSVD mean both returned non-informative biclusters containing all of the observations. The two HSSVD variance clusters included over 3300 observations and more than 100 features.

Examining the observations identified in the biclusters, we find that all but one of the observations in the first, second, and third Plaid biclusters were also in the first $SCBiclust_\beta$ bicluster. On the contrary, all of the observations in the first LAS bicluster were not found in the first $SCBiclust_\beta$ bicluster, but all except one of the observations in the third LAS bicluster were contained in the first $SCBiclust_\beta$ bicluster. We did not find any patterns between the second LAS bicluster and the $SCBiclust_\beta$ biclusters.

Table 3.13: *OPPERA: comparison of different biclustering algorithms*

Algorithm (computing time)	Layer	Bicluster composition	
		# obs. (case; non-case)	# features (Auto; Psy; QST)
SCBiclust (2.59 min)	Layer 1	1561 (110; 1451)	30 (30; 0; 0)
	Layer 2	998 (89; 909)	29 (0; 29; 0)
	Layer 3	1619 (118; 1501)	6 (0; 0; 6)
SSVD (21.28 sec)	Layer 1	3443 (185; 3258)	98 (44; 22; 32)
HSSVD (12.47 min)	Mean 1	3443 (185; 3258)	115 (44; 39; 32)
	Mean 2	3443 (185; 3258)	116 (44; 39; 33)
	Var 1	3378 (184; 3194)	109 (44; 36; 29)
	Var 2	3408 (185; 3223)	111 (44; 36; 31)
Plaid (14.08 sec)	Layer 1	68 (6; 62)	23 (23; 0; 0)
	Layer 2	6 (1; 5)	21 (21; 0; 0)
	Layer 3	23 (2; 21)	21 (7; 14; 0)
LAS (14.66 min)	Layer 1	817 (33; 784)	24 (24; 0; 0)
	Layer 2	638 (73; 565)	43 (0; 23; 20)
	Layer 3	945 (78; 867)	24 (24; 0; 0)

Membership in the biclusters identified by each method was evaluated as a possible risk factor for both chronic TMD and first-onset TMD. (Subjects with chronic TMD were excluded from the analysis for first-onset TMD.) Since SSVD and HSSVD mean included all of the observations in the biclusters they will be excluded from the analysis. The results from the Chi-squared test of association between each bicluster and chronic TMD are shown in Table 3.14. The results from the Cox proportional hazards model predicting incident TMD from bicluster membership are also shown in Table 3.14. Kaplan-Meier plots for first-onset TMD for selected biclusters are provided in Figure 3.18. All three

Table 3.14: *OPPERA: association between biclusters and chronic and first-onset TMD*. LRS=Log-rank Statistic

Association between biclusters and chronic TMD						
Algorithm	Bicluster 1		Bicluster 2		Bicluster 3	
	χ^2 (df=1)	p-value	χ^2 (df=1)	p-value	χ^2 (df=1)	p-value
SCBiclust	15.13	1.0×10^{-4}	33.75	6.3×10^{-9}	21.34	3.8×10^{-6}
HSSVD var	1.23	2.7×10^{-1}	1.08	3.0×10^{-1}	NA	NA
Plaid	1.01	3.2×10^{-1}	0.10	7.5×10^{-1}	0.06	8.1×10^{-1}
LAS	3.41	6.5×10^{-2}	55.27	1.1×10^{-13}	20.49	6.0×10^{-6}
Association between biclusters and first-onset TMD						
Algorithm	Bicluster 1		Bicluster 2		Bicluster 3	
	LRS (df)	p value	LRS (df)	p value	LRS (df)	p value
SCBiclust	2.72 (df=1)	9.9×10^{-2}	41.01 (df=1)	1.5×10^{-10}	3.95 (df=1)	4.7×10^{-2}
HSSVD var	0.40 (df=1)	5.3×10^{-1}	0.26 (df=1)	6.1×10^{-1}	NA	NA
Plaid	2.87 (df=1)	9.0×10^{-2}	0.42 (df=1)	5.2×10^{-1}	0.07 (df=1)	8.0×10^{-1}
LAS	0.50 (df=1)	4.8×10^{-1}	31.18 (df=1)	2.4×10^{-8}	1.71 (df=1)	1.9×10^{-1}

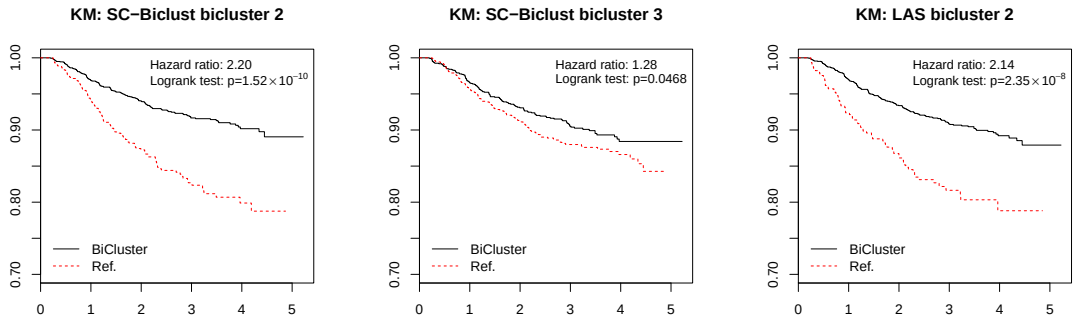

Figure 3.18: *OPPERA Kaplan-Meier plots.* The Kaplan-Meier plots showing the association between first-onset TMD and the biclusters identified by SCBiclust (layer 2 and 3) and LAS (layer 2).

biclusters identified by SCBiclust were associated with chronic TMD. The second bicluster was also strongly associated with first-onset TMD and the third bicluster showed moderate association. The second and third biclusters identified by LAS were associated with chronic TMD, and the second bicluster was also associated with first-onset TMD. The remaining biclusters identified by HSSVD, Plaid, and LAS were associated with neither chronic TMD nor first-onset TMD. The sparse biclustering algorithm failed to detect any biclusters.

3.4.2 Analysis of a Gene Expression Data Set

The data set used in this section contains gene expression measurements on 4751 genes from tissue samples from a total number of 78 breast cancer patients. The time to metastases of each subject is also available. See van't Veer et al. (2002) for a more detailed description of this data set.

The first biclusters identified by $SCBiclust$ and LAS algorithms both contain exactly the same 16 observations, but the $SCBiclust_\beta$ bicluster has 8 features whereas the LAS bicluster has 1421 features (Table 3.15). ($SCBiclust_U$ found 3 of the same features as $SCBiclust_\beta$ and one additional feature). The primary bicluster identified by the sparse biclustering method contains 60 observations and 553 features. The HSSVD method identified 8 mean bicluster layers and 3 variance bicluster layers, for which we will only study the primary mean layer. The Plaid method failed to identify any biclusters within the data set, and the SSVD method and the HSSVD variance identification did not produce valid biclusters.

We tested the null hypothesis of no association between each putative bicluster and metastases using log rank tests. Table 3.15 show the associations between metastases and the biclusters identified by SCBiclust, HSSVD (mean layer only), LAS, and sparse biclustering. A Kaplan-Meier plot is provided in Figure 3.19. The putative biclusters identified by SCBiclust, LAS, and sparse biclustering were associated with time to metastases, but the putative bicluster identified by HSSVD mean was not.

Table 3.15: *Gene expression: Comparison of biclustering and survival analysis results.*

Algorithm	Obs.	Feature	Score (log-rank) test	
			Statistic (df)	p value
$SCBiclust_\beta$	16	8	11.11 (df=1)	0.0009
HSSVD mean	75	1046	0.42 (df=1)	0.5150
LAS	16	1421	11.11 (df=1)	0.0009
Sparse Biclustering	60	553	10.20 (df=1)	0.0014

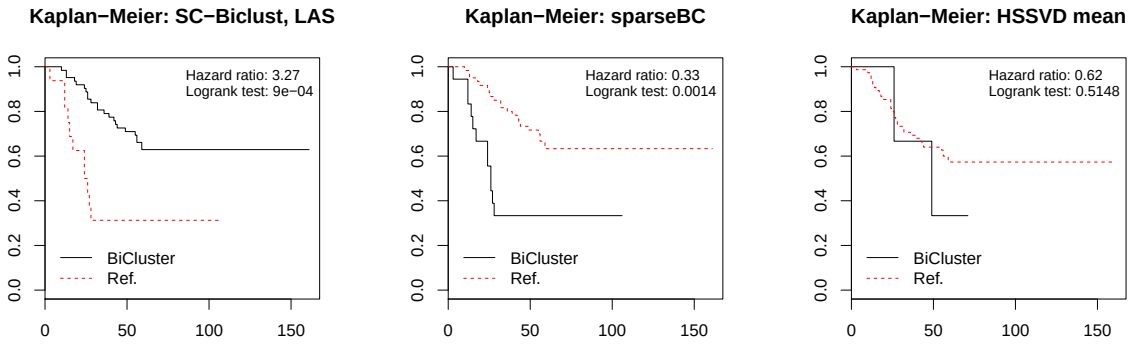

Figure 3.19: *Breast cancer gene expression Kaplan-Meier plot.* The Kaplan-Meier plots showing the association between time to metastases (months) and the biclusters identified by SCBiclust, LAS, and sparse biclustering, and HSSVD mean.

3.5 Discussion

Biclustering is an unsupervised learning method that can be useful for uncovering underlying data patterns in HDLSS data. In addition to identifying clusters of observations, features responsible for the clusters are also identified. Uncovering the features responsible for clustering may be especially important if one wishes to group additional data into pre-identified clusters. In this paper we have proposed a biclustering method which extends sparse clustering (Witten and Tibshirani, 2010) to also identify distinguishing features. The method does not place any distributional constraints on the data or clusters and can be used to identify both mean-based biclusters and more complex structures identified through hierarchical clustering.

In simulation studies and real data analysis we illustrate that the proposed method compares favorably with existing methods. SCBiclust tends to correctly identify biclusters

with high feature and observation accuracy. Also, unlike some biclustering methods such as Plaid (Lazzeroni and Owen, 2002) and sparse biclustering (Tan and Witten, 2014), the proposed method does not hinge upon the assumption that biclusters have the same mean. We have shown in the hierarchical clustering example given in Simulation 5 that the method can be adapted to incorporate other methods for identifying clusters. All that is required for SCBiclust is a function which increases as a measure of the distance between the biclusters and the remaining observations grows, and a method for maximizing this function with respect to the observations.

We have proposed two ways for generating null weights for SCBiclust. $SCBiclust_\beta$ makes use of distributional assumptions about feature weights assuming that the between cluster sum of squares for features are uncorrelated. The other method, $SCBiclust_U$, generates unimodal null data and determines features weights from clustering this null data. We find that both methods generally produce comparable results. $SCBiclust_U$ may be mores suited to find biclusters in non-normal settings, but may be more restrictive about feature selection. Using the weights produced by $SCBiclust_\beta$ greatly reduces the computation time. If possible we recommend using both methods and comparing the results to identify biclusters in a data set.

Finally, SCBiclust can be modified to incorporate any cluster significance testing method to be used as a stopping criteria for biclusters identification. Evaluating the number of clusters, or biclusters, present in the data is an ongoing field of research so having a method that can be flexible to advances in research is important. Currently we iteratively employ the SigClust algorithm (Liu et al., 2008) to test the significance of each putative bicluster. We chose this method for the present paper because of its accuracy in many situations and its relatively short computation time. In our simulation studies we found that the stopping criteria used by SCBiclust was generally more accurate than the methods used by the other biclustering methods, but it may identify slightly more biclusters than are

present in the data. A future area of research includes modifying the criteria for generating feature weights to also include a non-parametric test for cluster significance.

In this paper we have shown that SCBiclust performs well in terms of biclusters identification and reproducibility in both simulation and real data. SCBiclust is able to identify both biclusters that differ from the rest of the data in terms of feature means and other complex structures that can be identified through hierarchical clustering. Future work includes extending the method to identify biclusters that differ based on feature variance, perhaps by extending the method to SVD based approaches. An additional avenue of future research includes identifying network based biclusters such that observations in the biclusters are more correlated than observations outside of the biclusters. These future advances could further extend the application of SCBiclust in identifying subgroups and distinguishing features in more complicated HDLSS data.

CHAPTER 4: PERMUTATION ASSOCIATION TESTING BETWEEN A SECONDARY PHENOTYPE AND GENETIC MARKERS IN A CASE-CONTROL STUDY

4.1 Introduction

Case-control studies are a common study design to evaluate the association between genetic risk factors and a disease outcome. The case-control design can be used to efficiently estimate the odds ratio between a genetic factor and a disease (Prentice and Pyke, 1979) without the large sample size and follow-up time required for a cohort study. Due to the expense of collecting data in a genome wide association study (GWAS) the case-control design is an especially attractive study option. To further maximize the investment, researchers often collect as much information as possible, such as secondary phenotypes, which may be associated with the primary disease outcome. Since the case-control study is not a random sample from the population, naïvely using unadjusted regression models to evaluate the association between a risk factor (such as a SNP or other genetic marker) and a secondary phenotype will produce biased results (Richardson et al., 2007, Monsees et al., 2009, and Lin and Zeng, 2009).

Several statistical methods have been developed for analyzing the association between putative risk factors and secondary outcomes in a case-control study. The inverse probability weighting (IPW) method (Monsees et al., 2009 and Richardson et al., 2007) effectively down-weights cases in the study so that the weight of cases in the analysis is comparable to the proportion of cases in the general population. Since this effectively decreases the sample size, the sandwich estimator must be used to find the correct variance of the resulting regression coefficients (Monsees et al., 2009) which may result in low power. Lin and Zeng (2009) take a different approach and use likelihood functions to account for the case-

control sampling. Also, both the IPW and maximum likelihood estimation approaches do not specifically consider the situation when the secondary and primary phenotypes are correlated. To account for this possibility, He et al. (2012) use Gaussian copulas to model the primary and secondary phenotypes within a case-control sampling scheme. This approach requires that the secondary phenotypes have a distribution belonging to the exponential family. Wang and Shete (2009) use a parametric bootstrap approach to estimate the odds ratio between genetic variates and secondary phenotypes. However, the parametric bootstrap likely suffers from the same issues as conventional parametric models. Also, it is unclear that it can be generalized to situations where either the secondary phenotype or the covariate of interest is non-binary. Since parametric regression models can produce unstable parameter estimates when applied to secondary phenotypes that are strongly associated with case status there is a need for a nonparametric, or semi-parametric, association test that is robust to situations where the assumptions of parametric regression models are violated.

In this study, we propose a permutation-based IPW method to analyze the association between genetic markers and secondary phenotypes in a case-control study where the secondary outcome is correlated with case status. The proposed method has the advantage of being completely non-parametric when covariates are not included in the model and semi-parametric otherwise. Thus this method will produce valid p-values even when parametric model assumptions are violated. We also show that the proposed method has comparable power and running time to conventional IPW regression. In addition to producing permutation p-values we also propose a semi-parametric bootstrap method to estimate the association between the genetic factor and secondary phenotype and standard errors for the parameter estimate. We compare the power and type-I error rates of the proposed method to the conventional IPW method in extensive simulation studies. We also use the proposed method to analyze the association between SNPs and secondary phenotypes collected in the Orofacial Pain: Prospective Evaluation and Risk Assessment (OPPERA) case-control study.

4.2 Methods

Suppose genetic markers and secondary phenotypic data has been collected within a case-control study. Let n_1 and n_0 represent the total number of study individuals with and without the disease of interest, respectively. Let p represent the prevalence of disease in the total population. Let D_i denote the disease status for individual i, with D_i = 1 for cases and D_i = 0 for controls. Y_i represents the secondary phenotype. X_{ij} represents the number of copies of the minor allele for SNP $j = 1, \ldots, N$.

4.2.1 P-value Calculation

We wish to test the association between a given SNP and the secondary phenotype. Specifically, our null and alternative hypothesis can be represented as:

H_0: There is no association between X_j and Y.

H_a: There is an association between X_j and Y.

The motivation for the proposed method is the following: If a given SNP is associated with a secondary phenotype, we would expect that the vector of minor allele counts for the SNP, X_j, would be highly correlated with the secondary phenotype, Y. If Y were permuted to form a new vector, Y^*, this association would be eliminated. We would expect that the (absolute) correlation between X_j and Y would be greater than the correlation between X_j and Y^*. Thus, we may test the null hypothesis by comparing the correlation between X_j and Y to the correlations between X_j and a series of Y^*s, where each Y^* is a permutation of Y.

The procedure can be summarized as follows:

1. To account for the over-representation of disease cases in the data assign a weight of $w_0 = 1$ to controls and a weight of $w_1 = \frac{p*n_0}{(1-p)*n_1}$ to the cases.

2. Calculate the weighted correlation between X_j and Y. Denoted this correlation as R_j.

3. Permute Y B times to produce $Y_1^*, Y_2^*, \ldots, Y_B^*$

4. For each Y_b^*, $b = 1, \ldots, B$, calculate the weighted correlation between each X_k, $k = 1, \ldots, N$, and Y_b^*. Denote this correlation as R_{jb}^*

5. The p-value for the test of the null hypothesis is then given by

$$p_j = \frac{1}{NB} \sum_{k=1}^{N} \sum_{b=1}^{B} I\left(|R_{kb}^*| \geq |R_j|\right) \tag{4.38}$$

This procedure is valid even if the secondary phenotype Y is dichotomous, since the squared correlation between X_j and Y is proportional to the Armitage trend χ^2 statistic in this case (Price et al., 2006). Note that this test assumes that the distribution of the R_{kb}^*'s does not depend on k. In practice, this assumption is unlikely to be perfectly satisfied. However, this results in enormous computational savings. Under this assumption, B=25 provides a sufficient number of permutations to demonstrate that a SNP is associated with a secondary phenotype at a Bonferroni-corrected threshold for genome-wide significance. Without this assumption, millions of permutations would be required for each SNP, which is intractable computationally.

Now suppose one wishes to evaluate the association between an allele count X_j and a secondary phenotype Y after controlling for covariates $Z = Z_1, \ldots, Z_K$, such as demographic covariates or eigenvectors corresponding to race or ancestry. The above procedure can be modified as follows:

1. Perform a weighted regression of X_j on Z and find the vector $X_j^{'}$ of residuals from the resulting model.

2. Similarly, perform a weighted regression of Y on Z and find the vector $Y^{'}$ of residuals from the resulting model.

3. Apply the permutation test procedure above using $X_j^{'}$ and $Y^{'}$ in place of X_j and Y, respectively.

Note that this procedure requires one to regress X_j on the covariates for each SNP. Thus, a naïve application of this procedure would require the computation of N regression models, which would be computationally expensive. The required computing time can be significantly reduced, however, by noting that each regression model has exactly the same covariates. The only difference between these regression models is the outcome variable X_j. Suppose we are performing a weighted regression of X_j on Z. Let W be a diagonal matrix of the weights. Then the regression coefficients are given by $(Z^T W Z)^{-1} Z^T W X_j$, and estimated values of X_j are therefore given by $Z(Z^T W Z)^{-1} Z^T W X_j$. Let $H = Z(Z^T W Z)^{-1} Z^T W$. (This H is commonly known as the hat matrix.) Note that H does not on X_j. Thus, by calculating and storing the hat matrix H, one may calculate the residuals of the regression model to predict X_j based on Z by calculating X_j-HX_j, which requires only a single matrix multiplication rather than recomputing the entire regression model for each SNP. This approach is likely to substantially reduce the computation time needed for the procedure.

Now suppose that some minor allele counts are missing at random for some of the individuals in the study. We will show how using the Cholesky decomposition of $Z^T W Z$ can speed the computation of H. Note, we can determine L, the lower triangular matrix with real and positive diagonal entries which solves the expression $Z^T W Z = L^T L$. Then we have that $(Z^T W Z)^{-1} = L^{-1}(L^T)^{-1}$. When we have missingness in X_j we will need to recompute H. Let Z^* represent the matrix of covariates with the individuals who have missing values of X_j removed, W^* represent the matrix of weights with the individuals who have missing values of X_j removed, z represent the matrix of covariates for individuals who have missing values of X_j, and w represent the matrix of weights for individuals who have missing values of X_j. Then $H = Z^*(Z^{*T} W^* Z^*)^{-1} Z^{*T} W^* = Z^*(Z^T W Z - z^T w z)^{-1} Z^{*T} W^*$. We can compute the down-dated Cholesky factor, U which solves $U^T U = Z^T W Z - z^T w z$ in a single step. Then we have that $H = Z^* U^{-1}(U^T)^{-1} Z^{*T} W^*$. Down-dating the Cholesky factor takes far less time than recomputing $(Z^{*T} W^* Z^*)^{-1}$ for each regression model.

4.2.2 Strength of Association

The methodology described in Section 4.2.1 can be used to perform a nonparametric test (or semiparameteric test when covariates are included in the model) of the null hypothesis of no association between a given SNP and a secondary phenotype. However, this methodology only produces a p-value. It does not give a measure of the strength of the association or associated confidence intervals. We now present a simple way to estimate the association between X_j and Y.

Note that for a simple ordinary least squares regression model the estimate of the slope between X_j and Y is simply $r\frac{\text{sd}_Y}{\text{sd}_{X_j}}$ where r is the correlation between X_j and Y, sd_Y is the standard deviation of Y, and sd_{X_j} is the standard deviation of X_j. Thus for the case when no covariates are included in the model the association between X_j and Y can be estimated as

$$\beta_{X_jY} = r^* \frac{\text{sd}_{Y^*}}{\text{sd}_{X_j^*}} \tag{4.39}$$

where r^* is the weighted correlation between X_j and Y, Y^* is the weighted version of Y, and X_j^* is the weighted version of X_j.

If covariates are included in the model then residuals from the weighted regression of X_j on Z and residuals from the weighted regression of Y on Z can be used in place of X and Y, respectively. Note the hat matrix trick described previously can be used to reduce computation time.

For estimating the variance of the coefficient estimates we propose using a bootstrapping method (Efron, 1979):

1. For each SNP, generate B bootstrap samples of size n (where n is the number of participants in the study) by sampling from the data (X_j, w, Z, and Y) with replacement.

2. If no covariates, Z, are present then calculate $\beta_{jb} = r_b^* \frac{\text{sd}_{Y_b^*}}{\text{sd}_{X_{jb}^*}}$ For $b = 1, \ldots, B$ where r_b^* is the weighted correlation between Y_b and X_{jb}, $Y_b^* = Y_b \times w_b$, and $X_b^* = X_b \times w_b$.

3. If covariates are present in the model then perform the following procedure:

   - Perform a weighted regression of X_{jb} on Z_b and find the vector $X_{jb}^{'}$ of residuals from the resulting model.
   - Similarly, perform a weighted regression of Y_b on Z_b and find the vector $Y_b^{'}$ of residuals from the resulting model.
   - Use X_{jb}' and Y_b' in place of X_{bj} and Y_b, respectively in step 2 of the procedure.

4. Estimate the standard error of the coefficient estimate as

$$s = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\beta_{jb} - \overline{\beta_j})^2} \tag{4.40}$$

   Where $\overline{\beta_j} = \sum_{b=1}^{B} \beta_{jb}$

In this paper, B=100 simulations were used to calculate confidence intervals via the normal approximation. However, if one wishes to calculate confidence intervals using the percentile or BCa methods B=1000 or greater may be necessary.

4.3 Simulation Studies

In this section a simulation study is conducted to compare the proposed permutation method to the IPW method for assessing the association between SNPs and an secondary phenotype within a case-control study. The p-values and standard errors are assessed for both the conventional IPW regression and our permutation-based IPW methods. All simulation studies included 10,000 SNPs. In the first two simulations no associations were present between the simulated SNPs and secondary phenotypes and type-I error rates are compared.

To match the real data setting 5% of the SNPs were assumed to be missing. The p-value for conventional IPW regression was calculated using the "geepack" R package as described in Monsees et al. (2009) with the weighting specified for a retrospective case-control study (Wang and Shete, 2009). Specifically controls were given a weight of $w_0 = 1$ and cases were given a weight of $w_1 = n_0 p/(n_1(1-p))$ where p represents the disease prevalence in the population. Unless otherwise noted, the prevalence of the disease was assumed to be 5% in the general population.

In the last simulation, the SNP and secondary phenotype are associated and the power of both methods is assessed. The power was compared for varying levels of disease prevalence, minor allele frequency, and varying strengths of association between case status and SNP, case status and secondary phenotype, and SNP and secondary phenotype.

4.3.1 Simple Type-I Error Study

First a data set with 100 subjects and 10,000 SNPs was simulated. The first 50 subjects were designated as cases, and the remaining subjects were designated as controls. The secondary outcome variable was assumed to be an integer between 0 and 10. For cases, the secondary outcome was generated under a uniform distribution on the integers between 1 and 10. For each control, the secondary outcome variable was defined to be 0 with probability 0.9 and an integer generated uniformly between 1 and 10 with probability 0.1. The minor allele frequency (MAF) of each SNPs was randomly generated from a uniform distribution on (0.05, 0.1). After generating the MAF, the number of minor alleles for each subject at each SNP was generated under a binomial distribution with two trials and success probability equal to the MAF. No covariates were included in this simulation.

The Q-Q plots of the observed p-values versus the expected (uniform) distribution of the p-values for conventional IPW regression and our proposed method are shown in Figure 4.20 . Lines were added to each plot to show the expected distribution of the p-values as well as the significance thresholds corresponding to false discovery rates (FDRs) (Benjamini and

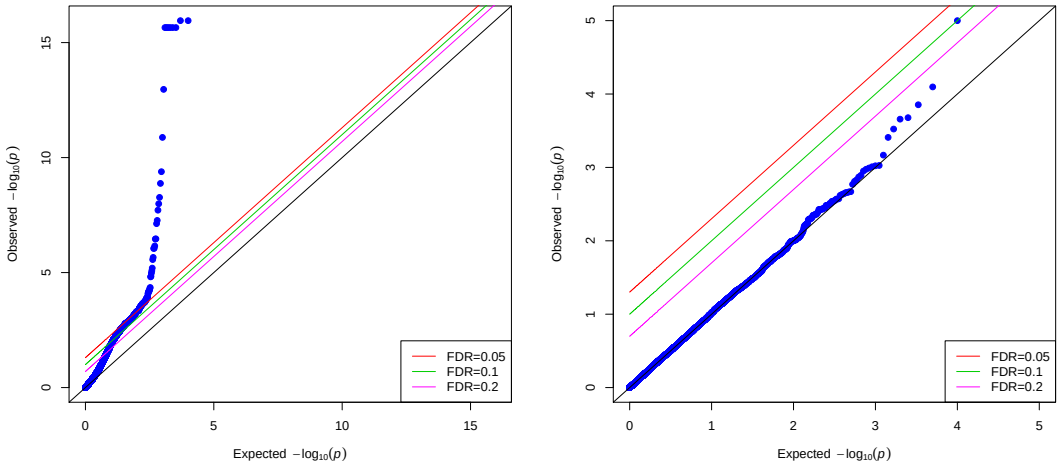

Figure 4.20: *Q-Q plot of the p-values produced by conventional IPW regression (left) and our proposed semi-parametric test (right) from the simulated data set described in section 4.3.1.*

Hochberg, 1995) of 0.2, 0.1, and 0.05, respectively. If a point (corresponding to a SNP) on the graph is above the line corresponding to an FDR of 0.05, that implies that the estimated false discovery rate would be less than 0.05 if that SNP (and all other SNPs above the line) were called significant.

We see in Figure 4.20 that conventional IPW regression produces inflated p-values in this simulation scenario. The genomic inflation factor lambda was equal to 2.24 (Table 4.16). This is the result of the fact that nearly all of the variance in the outcome occurs among cases. Cases were down-weighted as part of the IPW procedure, causing the associated regression coefficients to have high variance. In contrast, our nonparametric IPW regression method produced only one p-value at the FDR cutoffs of 0.1 (Figure 4.20). Our method also had a reasonable genomic inflation factor of 1.01 (Table 4.16, suggesting that our method does not suffer from this shortcoming.

The average estimated standard error produced by the conventional IPW method was smaller than the value produced by the bootstrap method (0.48 versus 0.97, Table 4.16). However, the standard error produced by the bootstrap method was larger than the conventional estimate for only 3,951 of the 10,000 SNPs. The 95% confidence intervals produced

by conventional IPW regression contained 0 (the true coefficient) for 8,234 SNPs. This is significantly less than 95%, indicating that these confidence intervals tend to be anticonservative. The bootstrap confidence intervals contained 0 for 9,193 SNPs, indicating that our proposed bootstrap method provides a more reasonable estimate of the standard error in this situation.

4.3.2 Type-I Error Study with Population Stratification

An additional simulation was conducted to assess the type-I error rate of the permutation IPW method. This simulated data set consisted of 1,000 subjects (with 500 controls and 500 cases) and 10,000 SNPs. Secondary phenotypes were generated as in the simulation described in section 4.3.1. It was assumed that two subpopulations were present in the data with 300 of the controls and 200 of the cases belonging to population 1, and the remaining study participants in population 2. For the first 100 SNPs, the ancestral allele frequencies, f_{ij}, were chosen to be $f_{1j} = 0.2$ for population 1 and $f_{2j} = 0.8$ for population 2, $j = 1, \ldots, 100$, $i = 1, 2$, to model the fact that a small number of SNPs are expected to have larger differences between the two populations. For the remaining SNPs, first h_j was generated for each SNP under a uniform distribution on (0.1, 0.9). The allele frequencies for subpopulation $i = 1, 2$ at SNP $j = 101, \ldots, 10,000$ were generated using a Balding-Nichols model (Balding and Nichols, 1995):

$$f_{ij} \sim Beta\left(\frac{h_j(1 - F_{ST})}{F_{ST}}, \frac{(1 - h_j)(1 - F_{ST})}{F_{ST}}\right) \tag{4.41}$$

Where $F_{ST} = 0.01$ represents Wright's coefficient which is a measure of shared ancestry (Balding and Nichols, 1995). Price et al. (2006) choose this value for F_{ST} since it is close to observed values for divergent European populations (Cavalli-Sforza et al., 1994 and Nicholson et al., 2002). For each SNP, the count of minor alleles for each subject was generated under a binomial distribution with two trials and probability of success f_{ij}. To

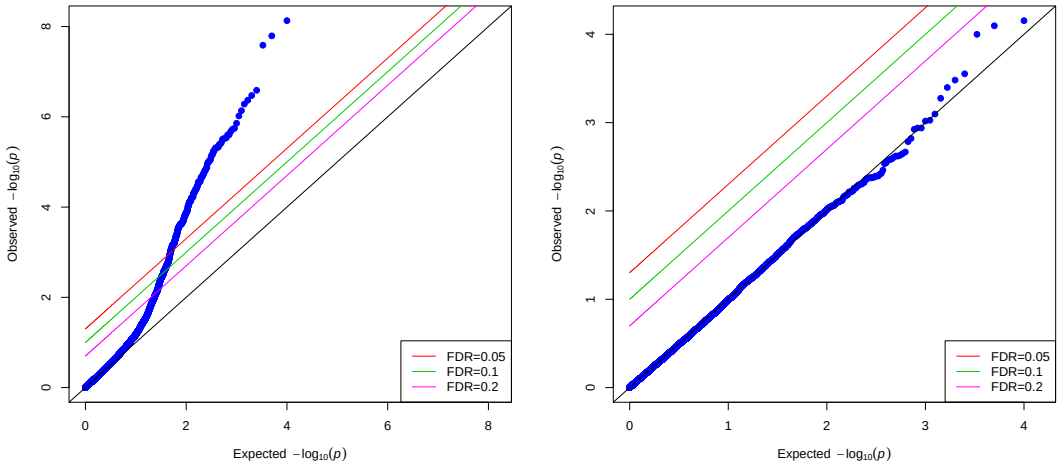

Figure 4.21: *Q-Q plot of the p-values produced by conventional IPW regression (left) and our proposed semi-parametric test (right) from the simulated data set described in section 4.3.2.*

control for population stratification, the value of the first 10 eigenvectors of the data matrix were included as covariates in the regression models.

Figure 4.21 gives the Q-Q plots for the observed p-values versus the expected (uniform) distribution of the p-values for conventional IPW regression and our proposed method. The p-value produced by conventional IPW regression are much smaller than would be expected (Lambda=1.70, Table 4.16) whereas the p-values for the proposed method have an appropriate distribution (Lambda=0.99, Table 4.16).

The average estimated standard error produced by the bootstrap method was slightly smaller than the value produced by the conventional IPW method (0.19 versus 0.23, Table 4.16). Similarly, the standard error produced by the bootstrap method was larger than the conventional estimate for only 461 of the 10,000 SNPs. The 95% confidence intervals produced by conventional IPW regression contained 0 (the true coefficient) for 9,237 SNPs whereas the bootstrap confidence intervals contained 0 for 9,351 SNPs (Table 4.16).

Table 4.16: *Comparison of conventional IPW vs. proposed permuted p-values and bootstrapped standard errors for simulations where there was no association between genetic factors and secondary phenotype*. "Lambda" represents the genomic inflation factor with the optimal value being 1. "Ave SE" represents the average standard error. "CI with 0" represents the number of confidence intervals (out of 10,000 SNPS), containing zero.

Value	Simulation 4.3.1		Simulation 4.3.2	
	Conventional IPW	Proposed IPW	Conventional IPW	Proposed IPW
Lambda	2.24	1.01	1.70	0.99
Ave. SE	0.48	0.97	0.23	0.19
CI with 0	8234	9154	9237	9351

4.3.3 Power Simulation

To assess the power of both the conventional IPW and proposed permutation method simulated data sets were generated where the secondary phenotype was associated with the genetic factor of interest. Each data set consisted of 2,000 individuals with 10,000 measured SNPs and was similar to the simulation proposed by Monsees et al. (2009).

Diallelic genotypes were simulated assuming Hardy-Weinberg equilibrium with minor allele frequency taking values of 0.05, 0.075, and 0.1. Let X_i represent the number of minor alleles carried by individual i. The continuous secondary phenotype, Y_i was modeled as $Y_i = \beta_{XY} X_i + \epsilon_i$ where $\epsilon_i \sim N(0, 1)$. The variance of Y explained by X, r^2_{XY}, took values of approximately 0.005 and 0.01. The β_{XY}'s were chosen to be positive values that were consistent with producing appropriate r^2_{XY} values.

Disease status was modeled as $\text{logit}\{P(D_i = 1|Y_i, X_i)\} = \beta_0 + \beta_{DX} X_i + \beta_{DY} Y_i$ with $\beta_{DX} = \{0, \log(1.7)/2, \log(1.7)\}$ and $\beta_{DY} = \{0, \log(2)/2, \log(2)\}$. β_0 was chosen to ensure a disease prevalence, p, of 0.01, 0.05, or 0.10. The population was assumed to be of size (3000/p). 1000 disease free cases and 1000 controls were sampled from the population and the data from the remaining individuals was discarded.

Both conventional IPW and the permutation-based method were used to assess the association between X and Y. The null hypothesis of no association between X and Y was rejected if the associated p-value was less than 0.05. 10,000 simulations were conducted for

each choice of minor allele frequency, $r^2_{XY}, \beta_{DX}, \beta_{DX}$, and p. Power was reported as the number of p-values less than 0.05 divided by 10,000.

The confidence intervals, bias in parameter estimation, and standard errors produced by the conventional IPW method and the proposed bootstrap method were compared for 1,000 simulations for each choice of minor allele frequency, r^2_{XY},β_{DX},β_{DY} when disease prevalence of 0.10.

Tables 4.17, 4.18, and 4.19 show the estimated power of conventional IPW regression and the proposed permutation method. We find that the power is comparable for all choices of minor allele r^2_{XY}, $\beta_{XY}, \beta_{DX}, \beta_{DX}$, and p.

Examining tables 4.20-4.22 we find that both the conventional IPW and the bootstrap method had comparable bias in parameter estimation and comparable average standard error. The confidence intervals produced by both methods had approximately 95% coverage.

Table 4.17: *Power (Percentage of simulations which gave a p-value less than 0.05) for conventional IPW method (IPW) versus the proposed permutation method (Perm).* 10,000 simulations were run for each combination of parameters. The prevalence of disease was set to be 0.01. MAF=minor allele frequency.

	$r^2_{XY} = .005$						$r^2_{XY} = .01$					
	MAF=0.05		MAF=0.075		MAF=0.1		MAF=0.05		MAF=0.075		MAF=0.1	
	IPW	Perm	IPW	Perm	IPW	Perm	IPW	Perm	IPW	Perm	IPW	Perm
1. β_{DY}= 0												
β_{DX}=0	0.6214	0.6196	0.6187	0.6088	0.6293	0.6226	0.8927	0.8937	0.8921	0.8923	0.8839	0.8827
β_{DX}=log(1.7)/2	0.6254	0.6179	0.6257	0.6273	0.6208	0.6151	0.8947	0.8920	0.8928	0.8906	0.8939	0.8960
β_{DX}=(1.7)	0.6224	0.6111	0.6281	0.6174	0.6183	0.6121	0.8953	0.8913	0.8971	0.8928	0.8962	0.8948
2. β_{DY}= log(2)/2												
β_{DX}=0	0.6225	0.6162	0.6206	0.6139	0.6250	0.6218	0.8973	0.8944	0.8968	0.8956	0.8915	0.8938
β_{DX}=log(1.7)/2	0.6225	0.6116	0.6145	0.6089	0.6269	0.6208	0.8948	0.8926	0.8928	0.8883	0.8964	0.8951
β_{DX}=log(1.7)	0.6238	0.6172	0.6228	0.6178	0.6333	0.6277	0.8891	0.8868	0.8983	0.8925	0.8969	0.8930
3. β_{DY}= log(2)												
β_{DX}=0	0.6172	0.6114	0.6231	0.6166	0.6234	0.6213	0.8918	0.8900	0.8924	0.8913	0.8935	0.8901
β_{DX}=log(1.7)/2	0.6263	0.6157	0.6232	0.6188	0.6213	0.6162	0.8975	0.8974	0.896	0.8931	0.8942	0.8913
β_{DX}=log(1.7)	0.6310	0.6153	0.6222	0.6151	0.6221	0.6165	0.8933	0.8909	0.8974	0.8941	0.8928	0.8908

Table 4.18: *Power (Percentage of simulations which gave a p-value less than 0.05) for conventional IPW method (IPW) versus the proposed permutation method (Perm).* 10,000 simulations were run for each combination of parameters. The prevalence of disease was set to be 0.05. MAF=minor allele frequency.

	$r^2_{XY} = .005$						$r^2_{XY} = .01$					
	MAF=0.05		MAF=0.075		MAF=0.1		MAF=0.05		MAF=0.075		MAF=0.1	
	IPW	Perm	IPW	Perm	IPW	Perm	IPW	Perm	IPW	Perm	IPW	Perm
1. β_{DY}= 0												
β_{DX}=0	0.6566	0.6513	0.6595	0.657	0.6577	0.6568	0.9137	0.9122	0.9065	0.9052	0.9132	0.9127
β_{DX}=log(1.7)/2	0.6587	0.6455	0.6621	0.6609	0.6512	0.6479	0.913	0.9094	0.9180	0.9167	0.9223	0.9188
β_{DX}=log(1.7)	0.6695	0.6546	0.6755	0.6666	0.6684	0.6606	0.9194	0.9098	0.9245	0.9200	0.9209	0.9149
2. β_{DY}= log(2)/2												
β_{DX}=0	0.6629	0.6546	0.6710	0.6625	0.6618	0.6596	0.9198	0.9167	0.9209	0.9172	0.9176	0.9182
β_{DX}=log(1.7)/2	0.6724	0.6608	0.6671	0.6574	0.6604	0.6522	0.9183	0.9142	0.9202	0.9166	0.9218	0.9184
β_{DX}=log(1.7)	0.673	0.6563	0.6770	0.6614	0.6702	0.6599	0.9216	0.9151	0.9267	0.9210	0.9221	0.9184
3. β_{DY}= log(2)												
β_{DX}=0	0.6705	0.6611	0.6627	0.6573	0.6611	0.6565	0.9218	0.9194	0.9154	0.9140	0.9264	0.9228
β_{DX}=log(1.7)/2	0.6828	0.6694	0.6712	0.6633	0.6826	0.6688	0.9282	0.9231	0.9261	0.9225	0.9306	0.9280
β_{DX}=log(1.7)	0.6849	0.6607	0.6940	0.6710	0.6774	0.6626	0.9337	0.9268	0.9324	0.9249	0.9318	0.9236

Table 4.19: *Power (Percentage of simulations which gave a p-value less than 0.05) for conventional IPW method (IPW) versus the proposed permutation method (Perm).* 10,000 simulations were run for each combination of parameters. The prevalence of disease was set to be 0.10. MAF=minor allele frequency.

	$r^2_{XY} = .005$						$r^2_{XY} = .01$					
	MAF=0.05		MAF=0.075		MAF=0.1		MAF=0.05		MAF=0.075		MAF=0.1	
	IPW	Perm	IPW	Perm	IPW	Perm	IPW	Perm	IPW	Perm	IPW	Perm
1. β_{DY}= 0												
β_{DX}=0	0.7047	0.6972	0.7008	0.699	0.6955	0.6914	0.9388	0.9383	0.9365	0.936	0.9335	0.9345
β_{DX}=log(1.7)/2	0.7076	0.6916	0.7123	0.702	0.7191	0.7085	0.9438	0.9412	0.9389	0.9365	0.9429	0.9393
β_{DX}=log(1.7)	0.7256	0.7062	0.7231	0.7002	0.7161	0.6951	0.9484	0.943	0.9478	0.9394	0.9468	0.9419
2. β_{DY}= log(2)/2												
β_{DX}=0	0.7084	0.7065	0.7013	0.6997	0.6981	0.696	0.9407	0.9409	0.9398	0.9386	0.9432	0.9408
β_{DX}=log(1.7)/2	0.7164	0.6966	0.7132	0.7005	0.7233	0.7087	0.9485	0.9453	0.9477	0.9445	0.9446	0.9434
β_{DX}=log(1.7)	0.7279	0.7089	0.7316	0.7149	0.7345	0.7135	0.9539	0.947	0.9532	0.9465	0.9509	0.9444
3. β_{DY}= log(2)												
β_{DX}=0	0.7226	0.7193	0.7232	0.716	0.7186	0.7113	0.9479	0.9455	0.9488	0.9458	0.9494	0.9465
β_{DX}=log(1.7)/2	0.7289	0.7127	0.7339	0.7143	0.7329	0.7182	0.9594	0.9554	0.9544	0.9515	0.9523	0.9479
β_{DX}=log(1.7)	0.7522	0.7230	0.7563	0.7321	0.7483	0.7273	0.9618	0.9548	0.9599	0.9545	0.9597	0.9564

Table 4.20: *Comparison of confidence interval coverage for conventional IPW versus the bootstrapped method.* 1,000 simulations were run for each combination of parameters. The minor allele frequency=0.05 and the disease prevalence was set to 10%. Bias=average absolute difference between parameter estimate and true β_{XY}. ASE=Average Standard Error. Coverage=Number of 95% confidence intervals containing the true β_{XY}.

	$r^2_{XY} = .005$						$r^2_{XY} = .01$					
	Bias		ASE		Coverage		Bias		ASE		Coverage	
	IPW	Boot	IPW	Boot	IPW	Boot	IPW	Boot	IPW	Boot	IPW	Boot
1. β_{DY}= 0												
β_{DX}=0	0.074	0.073	0.091	0.091	941	944	0.074	0.074	0.091	0.091	946	944
β_{DX}=log(1.7)/2	0.074	0.075	0.090	0.091	941	938	0.072	0.073	0.090	0.092	944	940
β_{DX}=log(1.7)	0.074	0.072	0.093	0.091	945	945	0.073	0.072	0.092	0.091	937	938
2. β_{DY}= log(2)/2												
β_{DX}=0	0.074	0.073	0.090	0.090	946	942	0.072	0.072	0.090	0.090	943	945
β_{DX}=log(1.7)/2	0.071	0.073	0.089	0.091	954	953	0.074	0.075	0.089	0.090	944	945
β_{DX}=log(1.7)	0.076	0.075	0.091	0.09	938	929	0.075	0.074	0.092	0.090	941	938
3. β_{DY}= log(2)												
β_{DX}=0	0.073	0.072	0.089	0.089	954	946	0.073	0.073	0.089	0.089	950	943
β_{DX}=log(1.7)/2	0.070	0.070	0.088	0.089	943	944	0.070	0.072	0.087	0.089	944	940
β_{DX}=log(1.7)	0.073	0.071	0.091	0.089	942	949	0.070	0.069	0.090	0.088	955	946

4.3.4 Simulation Summary

In sections 4.3.1 and 4.3.2 we examined the Type-I error rates of both the permutation based and conventional IPW methods. We found that the conventional IPW method produced p-values much smaller than would be expected. This is the result of the conventional IPW method's inability to handle the situation when the secondary phenotype is associated with disease status. We also found that the confidence intervals produced using the conventional IPW method tend to be anticonservative. In contrast, we found the p-values produced by the permutation-based IPW method tend to have an appropriate distribution and the confidence intervals produced by the bootstrapping method tend to be more conservative.

It should be noted that the permutation method results in substantial decrease in computation time. For the simulation described in section 4.3.1 the standard IPW method took 139.5 seconds and the permutation method took 35.5 seconds. The conventional method took 649.5 seconds for the simulation described in section 4.3.2 whereas the permutation method only took 58.7 seconds.

Table 4.21: *Comparison of confidence interval coverage for conventional IPW versus the bootstrapped method. 1,000 simulations were run for each combination of parameters.* The minor allele frequency=0.075 and the disease prevalence was set to 10%. Bias=average absolute difference between parameter estimate and true β_{XY}. ASE=Average Standard Error. Coverage=Number of 95% confidence intervals containing the true β_{XY}.

	$r^2_{XY} = .005$						$r^2_{XY} = .01$					
	Bias		ASE		Coverage		Bias		ASE		Coverage	
	IPW	Boot	IPW	Boot	IPW	Boot	IPW	Boot	IPW	Boot	IPW	Boot
1. β_{XY}=0												
β_{DX}=0	0.062	0.061	0.076	0.075	944	936	0.060	0.059	0.076	0.075	944	952
β_{DX}=log(1.7)/2	0.061	0.061	0.075	0.075	948	942	0.062	0.062	0.075	0.075	948	945
β_{DX}=log(1.7)	0.060	0.059	0.077	0.074	951	948	0.063	0.061	0.076	0.074	939	937
2. β_{XY}=log(2)/2												
β_{DX}=0	0.059	0.059	0.075	0.074	949	944	0.059	0.058	0.075	0.074	957	953
β_{DX}=log(1.7)/2	0.059	0.060	0.074	0.075	950	941	0.057	0.058	0.074	0.075	952	946
β_{DX}=log(1.7)	0.061	0.059	0.076	0.074	960	956	0.060	0.059	0.076	0.074	945	940
3. β_{XY}=log(2)												
β_{DX}=0	0.061	0.059	0.074	0.073	946	943	0.061	0.060	0.074	0.072	953	947
β_{DX}=log(1.7)/2	0.060	0.060	0.073	0.072	934	929	0.057	0.057	0.072	0.072	954	948
β_{DX}=log(1.7)	0.058	0.057	0.075	0.073	949	950	0.060	0.058	0.075	0.072	947	937

The bootstrapping method for estimating the standard error of the IPW estimates produced more reliable estimates than the conventional method. For the simulation described in section 4.3.1 the standard errors were larger on average indicating the true uncertainty in estimating the parameter. For the simulation described in 4.3.2 the standard errors were smaller than those produced by the IPW, while still having optimal coverage.

In section 4.3.3 we studied the power of the permutation-based IPW method in comparison to the standard method. We found that both methods had comparable power for various choices of minor allele frequency, r^2_{XY}, β_{DX},β_{DY} and p. The bootstrap method also had comparable performance to the conventional method in parameter estimation and confidence interval coverage.

Table 4.22: *Comparison of confidence interval coverage for conventional IPW versus the bootstrapped method. 1,000 simulations were run for each combination of parameters.* The minor allele frequency=0.1 and the disease prevalence was set to 10%. Bias=average absolute difference between parameter estimate and true β_{XY}. ASE=Average Standard Error. Coverage=Number of 95% confidence intervals containing the true β_{XY}.

	$r^2_{XY} = .005$						$r^2_{XY} = .01$					
	Bias		ASE		Coverage		Bias		ASE		Coverage	
	IPW	Boot	IPW	Boot	IPW	Boot	IPW	Boot	IPW	Boot	IPW	Boot
1. β_{XY}=0												
β_{DX}=0	0.054	0.053	0.067	0.065	946	945	0.052	0.051	0.067	0.065	952	945
β_{DX}=log(1.7)/2	0.053	0.052	0.066	0.065	940	934	0.053	0.052	0.066	0.065	960	955
β_{DX}=log(1.7)	0.053	0.051	0.067	0.065	958	948	0.056	0.054	0.067	0.065	946	941
2. β_{XY}=log(2)/2												
β_{DX}=0	0.051	0.050	0.066	0.065	946	940	0.053	0.052	0.066	0.065	953	943
β_{DX}=log(1.7)/2	0.052	0.052	0.065	0.065	953	950	0.051	0.051	0.065	0.065	941	939
β_{DX}=log(1.7)	0.055	0.053	0.067	0.064	937	931	0.056	0.054	0.067	0.064	945	946
3. β_{XY}=log(2)												
β_{DX}=0	0.050	0.049	0.065	0.063	961	951	0.053	0.051	0.065	0.063	952	939
β_{DX}=log(1.7)/2	0.051	0.050	0.064	0.063	950	950	0.050	0.050	0.064	0.064	954	957
β_{DX}=log(1.7)	0.050	0.048	0.066	0.063	952	947	0.053	0.051	0.066	0.063	952	941

4.4 Application to the OPPERA Study

We illustrate the utility of the permutation-based IPW method using real data from the Orofacial Pain: Prospective Evaluation and Risk Assessment (OPPERA) Study. The OPPERA Study is a large-scale multi-center prospective cohort study with a case control arm. The goal of the study was to identify the genetic, psychosocial and clinical factors associated with the development of temporomandibular disorder (TMD) (Slade et al., 2011). Information was collected via questionnaires, clinical examinations, and blood samples for study participants. Genotyping was performed using the Omni2.5 Bead Chip Illumina Platform, which provided minor allele counts for 2,567,845 SNPs. Genotypes are available for 999 TMD cases and 2,031 TMD-free controls.

Two secondary phenotypes are examined presently: “pain free opening”, and ”characteristic pain intensity. “Pain free opening” is the vertical range of pain free mandibular motion measured in millimeters. “Characteristic pain intensity” is a composite score on a 1-100 scale representing a participant’s current facial pain intensity, average facial pain in the past six months, and greatest facial pain experienced during that time.

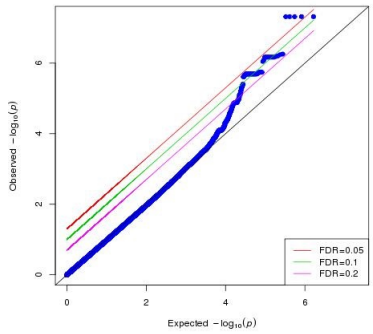

Figure 4.22: *Q-Q plot of the p-values produced by our proposed nonparametric test versus the expected (uniform) distribution for testing the association between each SNP and "pain free opening" in the OPPERA Study.*

For each analysis participants with non-missing secondary phenotype and all SNPs with a MAF greater than 0.02 were included. Principal component analysis was performed on the SNP matrix, and the first 6 components were included in the model as covariates. Gender and dummy variables for OPPERA study site were also included as covariates. The permutation-based IPW procedure was applied to calculate p-values for each SNP as described in Section 4.2.1. The p-values were compared to the p-values expected under a uniform distribution using Q-Q plots.

The analysis revealed several potential SNPs associated with the secondary phenotypes of interest. The QQ plot for "pain free opening" revealed several SNPs which exceeded the FDR=0.05 cutoff (Figure 4.22). The analysis for "characteristic pain intensity" revealed several SNPs with an association that did not quite reach the stringent FDR=0.05 cutoff (Figure 4.23).

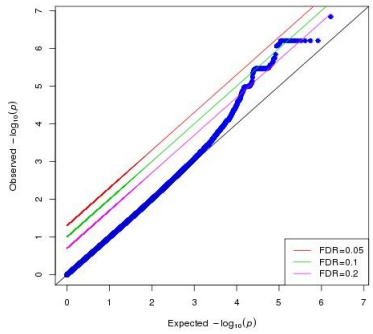

Figure 4.23: *Q-Q plot of the p-values produced by our proposed nonparametric test versus the expected (uniform) distribution for testing the association between each SNP and "characteristic pain intensity" in the OPPERA Study.*

4.5 Discussion

In genome wide association studies it is often of interest to analyze the association between a secondary phenotype and genetic factors. Inverse-probability-of-sampling-weighted (IPW) regression can be a useful tool to identify such associations, but it can produce unreliable estimates when there is a strong association between the secondary phenotype and disease status. In this study we have proposed a novel semi-parametric method for testing the association between genetic factors and a secondary phenotype within a case-control study. We also implement a bootstrap method to produce standard errors and confidence intervals for estimates of the association between the genetic factors and secondary phenotype. Through simulation studies and an application to the OPPERA study we have shown that the the proposed permutation based method retains the power of the conventional IPW method while having more appropriate control of the type I error rate.

In the simulation studies we explored the scenario where the secondary phenotype was equal to 0 for the majority of controls, and the genetic factors were not associated with the secondary phenotype or disease status. Simulation 4.3.1 did not include covariates in the analysis and simulation 4.3.2 included 10 eigenvectors of the data matrix to control for population stratification. We found that conventional IPW regression performed poorly with inflated type-I errors. The permutation-based method, on the other hand, produced p-values

very close to the expected uniform distribution. Also the confidence intervals produced by the bootstrapping method had better coverage than the conventional IPW method. When associations were present between the simulated SNPs and the secondary phenotype (section 4.3.3) we found that the permutation-based method had comparable power to conventional IPW regression analysis.

Since the proposed method showed promising results in the simulation studies it was used to assess the association between SNPs and secondary phenotypes in the OPPERA study. These secondary phenotypes are strongly associated with TMD status, with pain-free opening having a lower value and characteristic pain intensity having a higher value for study cases. The permutation-based method produced moderate p-values for the majority of SNPs but did reveal several SNPs that may be associated with clinical orofacial pain.

The proposed permutation method is easy to implement in common statistical software. When the number of genetic markers is large, fewer permutations are required resulting in fast computing times. This makes this method an ideal tool for assessing the strength of association between secondary phenotypes and genetic factors in GWAS when there is a strong correlation between secondary phenotype and disease status.

This study illustrates that the proposed method is a powerful and accurate method for detecting genetic factors associated with secondary phenotypes, but there are several avenues for future research to improve the method. Even though the confidence intervals produced by the bootstrapping method had better coverage than the conventional IPW confidence intervals, the coverage was less than the optimal 95% level. Additional research is needed to improve coverage rates and decrease computational times. Also the proposed method assumes that the distribution of correlations between the permuted phenotype and SNP, R^*_{jb}'s is unrelated to the j. This assumption may be violated in certain scenarios and future work is needed to expand the method to be used more broadly while still having a short computation time.

CONCLUSION

Non-parametric machine learning methods are widely used in many scientific areas due to their ability to discover important data structures. These methods have also been especially important for discovering underlying patters in HDLSS genetics data. In this dissertation we present two new advances in machine learning methodology, a non-parametric cluster significance test and a method for biclustering. We also explore the commonly occurring problem of analyzing the association between an secondary phenotype and genetic factors within a case-control study. These tools advance the field by providing the ability to assess clustering in a data set, identify distinguishing features responsible for clusters, and identify genetic factors associated with secondary phenotypes in case-control studies.

In the first part of the dissertation we develop the Unimodal Non-Parametric Cluster (UNPaC) test. This method can be used to assess the significance of clusters and estimate the number of clusters present in the data. UNPaC is compared to several existing methods in simulation studies and has comparable power, type-I accuracy, and, in addition, can be used in a wider way of settings than existing methods. Applications to the OPPERA study and cancer microarray data illustrate the utility of UNPaC.

We next delve into the topic of biclustering and further develop the sparse clustering method of Witten and Tibshirani (2010). By employing a cluster significance step we retain the ability to identify distinguishing features, but do not produce spurious biclusters even when the features are correlated. We compare the proposed method. SCBiclust, to existing biclustering methods and find it compares favorably in terms of accuracy and reproducibility.

In the last chapter of the dissertation, we step outside the domain of unsupervised learning and focus on developing a semi-parametric test for the association between secondary

phenotypes and genetic factors within a case-control study. This permutation-based test avoids placing distributional constraints on the secondary phenotype and can be computed quickly in standard statistical software. The proposed method has improved type-I accuracy compared to the IPW method (Monsees et al., 2009 and Richardson et al., 2007) and comparable power.

BIBLIOGRAPHY

Ahmed, M. and Walther, G. (2012). Investigating the multimodality of multivariate data with principal curves. *Computational Statistics and Data Analysis*, 56(12):4462–4469.

Bair, E., Brownstein, N. C., Ohrbach, R., Greenspan, J. D., Dubner, R., Fillingim, R. B., Maixner, W., Smith, S. B., Diatchenko, L., Gonzalez, Y., et al. (2013). Study protocol, sample characteristics, and loss to follow-up: The oppera prospective cohort study. *The Journal of Pain*, 14(12):T2–T19.

Bair, E., Gaynor, S., Slade, G., Ohrbach, R., Fillingim, R., Greenspan, J., Dubner, R., Smith, S., Diatchenko, L., and Maixner, W. (2016). Identification of clusters of individuals relevant to temporomandibular disorders and other chronic pain conditions: The oppera study. *PLOS BIOLOGY*, 157(6):1266–1278.

Bair, E. and Tibshirani, R. (2004). Semi-supervised methods to predict patient survival from gene expression data. *PLOS BIOLOGY*, 2(4):511–522.

Balabdaoui, F., Rufibach, K., and JA, W. (2009). Limit distribution theory for maximum likelihood estimation of a log-concave density. *The Annals of Statistics*, 37(3):1299–1331.

Balding, D. and Nichols, R. (1995). A method for quantifying differentiation between populations at multi–allelic loci and its implications for investigating identity and paternity. *Genetica*, 96(1):3–12.

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

Birge, L. (1997). Estimation of unimodal densities without smoothness assumptions. *The Annals of Statistics*, 25(3):970–981.

Calinski, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3(5):1–27.

Cavalli-Sforza, L., Menozzi, P., and Piazza, A. (1994). *The History and Geography of Human Genes*. Princeton Univ. Press, New Jersey.

Chen, G., Sullivan, P. F., and Kosorok, M. R. (2013). Biclustering with heterogeneous variance. *Proceedings of the National Academy of Sciences*, 110(30):12253–12258.

Cheng, Y. and Ray, S. (2014). Multivariate modality inference using gaussian kernel. *Open Journal of Statistics*, 4(5):419–434.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26.

Efron, B. and Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman & Hall/CRC, Boca Raton, FL.

Fang, Y. and Wang, J. (2012). Selection of the number of clusters via the bootstrap method. *Computational Statistics and Data Analysis*, 56:468–477.

Fillingim, R., Ohrbach, R., Greenspan, J., Knott, C., Dubner, R., Bair, E., Baraian, C., Slade, G., and Maixner, W. (2011). Potential psychosocial risk factors for chronic tmd: Descriptive data and empirically identified domains from the oppera case-control study. *The Journal of Pain*, 12(11):T46–T60.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Greenspan, J., Slade, G., Bair, E., Dubner, R. Roger B. Fillingim, R., Ohrbach, R., Knott, C., Mulkey, F., Rothwell, R., and William Maixner, W. (2011). Pain sensitivity risk factors for chronic tmd: Descriptive data and empirically identified domains from the oppera case control study. *The Journal of Pain*, 12(11):T61–T74.

Hall, P., Marron, J., and Neeman, A. (2005). Geometric representation of high dimension, low sample size data. journal of the royal statistical society. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):427–444.

Hastie, T. and Stuetzle, W. (1989). Principal curves. *J. Amer. Statist. Assoc.*, 84:502–516.

He, J., Li, H., Edmondson, A. C., Rader, D. J., and Li, M. (2012). A gaussian copula approach for the analysis of secondary phenotypes in case-control genetic association studies. *Biostatistics*, 13(3):497–508.

Hennig, C. (2014). How many bee species? a case study in determining the number of clusters. In Spiliopoulou, M., Schmidt-Thieme, L., and Janning, R., editors, *Data Analysis, Machine Learning and Knowledge Discovery*, pages 41–49. Springer.

Huang, H., Liu, Y., Yuan, M., and Marron, J. S. (2015). Statistical significance of clustering using soft thresholding. *Journal of Computational and Graphical Statistics*, 24(4):975–993.

Kapp, A. and Tibshirani, R. (2006). Are clusters found in one dataset present in another dataset? *Biostatistics*, 8(1):9–31.

Lazzeroni, L. and Owen, A. (2002). Plaid models for gene expression data. *Statistica sinica*, 12(1):61–86.

Lee, M., Shen, H., Huang, J. Z., and Marron, J. S. (2010). Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095.

Li, J., Ray, S., and Lindsay, B. (2007). A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, 8:1687–723.

Lin, D. Y. and Zeng, D. (2009). Proper analysis of secondary phenotype data in case-control association studies. *Genet Epidemiol.*, 33(3):256–265.

Liu, Y., Hayes, D. N., Nobel, A., and Marron, J. S. (2008). Statistical significance of clustering for high-dimension, low–sample size data. *Journal of the American Statistical Association*, 103(483).

Maitra, R., Melnykov, V., and Lahiri, S. (2012). Bootstrapping for significance of compact clusters in multidimensional datasets. *Journal of the American Statistical Association*, 107(497):378–392.

Maixner, W., Greenspan, J. D., Dubner, R., Bair, E., Mulkey, F., Miller, V., Knott, C., Slade, G. D., Ohrbach, R., Diatchenko, L., et al. (2011). Potential autonomic risk factors for chronic tmd: descriptive data and empirically identified domains from the oppera case-control study. *The Journal of Pain*, 12(11):T75–T91.

Monsees, G., Tamimi, R., and Kraft, P. (2009). Genomewide association scans for secondary traits using casecontrol samples. *Genet Epidemiol*, 33(8):717–728.

Nicholson, G., Smith, A., Jnsson, F., Gstafsson, O., Kri Stefnsson, K., and Peter Donnelly, P. (2002). Assessing population differentiation and isolation from single-nucleotide polymorphism data. *J. R. Statist. Soc. (B)*, 64(4):695–715.

Prentice, R. L. and Pyke, R. (1979). Logistic disease incidence models and case-control studies. *Biometrika*, 66(3):403–411.

Price, A., Patterson, N., Plenge, R., Weinblatt, M., Shadick, N., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nat Genet.*, 38(8):904–909.

Richardson, D. B., Rzehak, P., Klenk, J., and K., W. S. (2007). Analyses of case-control data for additional outcomes. *Epidemiology*, 18(4):441–445.

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math*, 20:53–65.

Sager, T. (1978). Estimation of a multivariate mode. *The Annals of Statistics*, 6(4):802–812.

Schiffman, E., Ohrbach, R., Truelove, E., Look, J., Anderson, G., Goulet, J., List, T., Svensson, P., Gonzalez, Y., Lobbezoo, F., Michelotti, A., Brooks, S., Ceusters, W., Drangsholt, M., Ettlin, D., Gaul, C., Goldberg, L., Haythornthwaite, J., Hollender, L., Jensen, R., John, M., De Laat, A., de Leeuw, R., Maixner, W., van der Meulen, M., Murray, G., Nixdorf, D., Palla, S., Petersson, A., Pionchon, P., Smith, B., Visscher, C., Zakrzewska, J., and Dworkin, S. (2014). Diagnostic criteria for temporomandibular disorders (dc/tmd) for clinical and research applications: Recommendations of the international rdc/tmd consortium network* and orofacial pain special interest group. *Journal of Oral & Facial Pain and Headache*, 28(1):6–27.

Scott, D. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley, New York, NY.

Shabalin, A. A., Weigman, V. J., Perou, C. M., and Nobel, A. B. (2009). Finding large average submatrices in high dimensional data. *The Annals of Applied Statistics*, pages 985–1012.

Silverman, B. (1981). Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society*, 43(1):97–99.

Slade, G., Bair, E., By, K., Mulkey, F., Baraian, C., Rothwell, R., Reynolds, M., Miller, V., Gonzalez, Y., Gordon, S., Ribeiro-Dasilva, M., Lim, P., Greenspan, J., Dubner, R., Fillingim, R., Diatchenko, L., Maixner, W., Dampier, D., Knott, C., and Ohrbach, R. (2011). Study methods, recruitment, sociodemographic findings, and demographic representativeness in the oppera study. *The Journal of Pain*, 12(11):T12–T26.

Tan, K. and Witten, D. (2014). Sparse biclustering of transposable data. *Journal of Computational and Graphical Statistics*, 23(4):985–1008.

Tarpey, T. and Flury, B. (1996). Multivariate density estimation: Theory, practice, and visualization. *Statist. Sci.*, 11(3):229–243.

Tibshirani, R. and Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Turner, H., Bailey, T., and Krzanowski, W. (2003). Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational Statistics and Data Analysis*, 48(2):235–254.

van't Veer, L. J., Dai, H., Van De Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Shreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536.

Wang, J. and Shete, S. (2009). Estimation of odds ratios of genetic variants for the secondary phenotypes associated with primary diseases. *Genet Epidemiol.*, 35(3):190–200.

Ward, J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

Witten, D. M. and Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490).

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.