CLUSTERING RARE EVENT FEATURES TO INCREASE STATISTICAL POWER

By

Robert Michael Sivley

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

In

Computer Science

May, 2013

Nashville, Tennessee

Approved:

Professor Douglas H. Fisher

Professor William S. Bush

Professor Tricia A. Thornton-Wells

To my parents, who always seem to know what I'm going to do before I do it.

(despite constantly asking me what it is that I do)

ACKNOWLEGEMENTS

While pursuing my Master's degree at Vanderbilt, several professors have assisted me through both mentorship and financial support. I would like to especially thank Dr. Tricia Thornton-Wells, who hired me as a work-study programmer during my unfunded first semester and brought me into her lab as a research assistant during my second. She was the first professor to take an interest in me, to mentor me, and is responsible for many of the opportunities I have had since coming to Vanderbilt.

I'd also like to thank Dr. William Bush, who pointed me in Tricia's direction when I first came to Vanderbilt, and brought me into his lab as a research assistant when he realized the terrible mistake he had made. He introduced me to the issue discussed in this thesis, coauthored a conference paper on the subject, mentored me throughout my thesis, and encouraged me to pursue a PhD in Biomedical Informatics.

Thank you to my advisor, Dr. Douglas Fisher, who taught me everything I know about artificial intelligence and machine learning, agreed to act as my thesis advisor despite a very busy schedule, helped brainstorm my original thesis topic, encouraged the transition to the current topic, and helped define the structure and focus of this thesis.

Finally, thank you to Alexandra Fish; a coauthor on the aforementioned conference paper, my source for all things biological, and *de facto* thesis editor. This thesis would not read nearly as well without her unwavering editorial support.

Financial support for my graduate study and this thesis were provided by the Department of Electrical Engineering and Computer Science, Dr. Tricia Thornton-Wells, and Dr. William Bush in part with development funds, NIH U01 HG004798 and its ARRA supplements, and the National Institute of Aging P30AG036445-01.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS ii

LIST OF TABLES vi

LIST OF FIGURES vii

Chapter

I. INTRODUCTION 1

Motivation 1
Binning and Collapsing 1
Clustering Binning 2
Chapters 3

II. REVIEW OF CORE CLUSTERING PARADIGMS 5

Introduction 5
Partitioning 6
k-Means 7
k-Medoids 8
k-Modes and *k*-Prototypes 9
Silhouettes 10
Hierarchical 10
Agglomerative and Divisive 11
Dissimilarity Matrices 11
Density 12
DBSCAN 13
OPTICS 13
Miscellaneous Clustering Paradigms 14
Constraints 14
Conceptual Clustering 15
Fuzzy and Probabilistic Clustering 15
Discussion 16

III. RVCLUST: AN R PACKAGE FOR RARE VARIANT CLUSTERING AND ANALYSIS 18

Introduction 18
Implementation 18
Initialization 18
Annotation 19
Cluster Analysis 19
Statistical Analysis 20

Software Design 20
Sample Run 21
Full Analysis 21
Annotation 21
Cluster Analysis 23
Statistical Analysis 23
Discussion 24

IV. KNOWLEDGE-CONSTRAINED K-MEDOIDS CLUSTERING OF REGULATORY RARE ALLELES FOR BURDEN TESTS 25

Preface 25
Introduction 25
Methods 27
Data 27
Domain Knowledge 28
Gene Selection 28
Cluster-based Analysis 29
Sliding Window Analysis 29
Determination of Significance 30
Visualization 30
Results 31
Gene Region Results 31
Bonferroni Correction 31
False Discovery Rate Correction 32
Discussion 34

V. DISCUSSION 35

Review 35
Improvements 36
Theoretical 36
Experimental 37
Additional 37
Future Directions 38
Conclusions 39

Appendix

A. RVCLUST SOFTWARE 40

REFERENCES 41

LIST OF TABLES

Table Page

1. Number of significant genomic regions detected using clustering and sliding window analysis with Bonferroni correction for multiple testing 32

2. Number of significant genomic regions detected using clustering a sliding window analysis with False Discovery Rare correction for multiple testing 32

LIST OF FIGURES

Table Page

1. Illustration of Voronoi diagrams 7

2. Demonstration and comparison of *k*-Means and *k*-Medoids 9

3. Illustration of hierarchical clustering results (dendrograms) 11

4. Sample density reachability generated by OPTICS 14

5. RVCLUST sample: full analysis with minimal output 22

6. RVCLUST sample: summary of *variants* after annotation and cluster analysis 22

7. RVCLUST sample: summary of *clusterinfo* after cluster analysis 23

8. RVCLUST sample: summary of *clusterinfo* after statistical analysis 24

9. Manhattan plot comparing sliding window analysis and cluster analysis results 33

CHAPTER I

INTRODUCTION

MOTIVATION

Rare genetic variants are thought to contribute to human disease, but statistically associating rare variants to diseases is difficult because of low statistical power. This issue does not stem from genetics, rather from statistical limitations to detect effects from Boolean features that rarely deviate from their usual value – *rare variants*. Many fields use Boolean features to indicate the occurrence of some event, indicating a deviation from some expected state or value. Association analysis tests for a correlation between the event and some observed variable – an *effect* – and the magnitude of that correlation – the *effect size*. The *significance* is a measure of confidence in the effect, representing the probability that an effect of that size or greater could have been detected by chance, given the sample size. Typically, a *significance threshold* is used to filter false positives from detected effects. When Boolean features have a low frequency of variation, the small number of occurrences reduces confidence that an effect is real, and detected effects are often dismissed as false positives. What is required for rare variant analysis is an increase in *statistical power*, which is the likelihood of detecting a real effect when one exists.

BINNING AND COLLAPSING

One approach to increasing statistical power for rare variants is to merge several features into a single, composite feature. This technique increases the overall frequency of variation, increasing the likelihood of detecting effects. However, because the features are merged, effects cannot be attributed to a single feature. Two methods for performing this merge are a *collapsing*

test, and a *burden test*. A collapsing test calculates the *disjunction* (*inclusive or*) of all features in the group, such that if any of the features are varied, the group is considered variant. Alternatively, a burden test represents the group by the total number of features in a group that show variation. The collapsing test assumes that if any variation exists in a group, there should be an effect on the observation, while burden testing assumes that the more variation in a group, the larger the effect on the observation. Furthermore, both approaches assume that all features in a group have the same direction of effect, meaning they all affect the observed variable in the same way. These assumptions highlight the need for variant groups to be meaningfully defined, improving the likelihood that the assumptions will be met. Binning is the process of grouping features for collapsing or burden testing. Because uninformed binning methods do not leverage domain knowledge, they may generate meaningless bins. One such method is sliding window analysis, a field-specific approach described in chapter 4. These methods might instead attempt to capture meaningful bins by generating large numbers of overlapping bins, representing a variety of feature combinations. This approach requires strict significance thresholds to compensate for multiple test correction, but a lack of independence between tests makes these thresholds overly conservative correction. Alternatively, cluster analysis uses similarity measures to meaningfully group features, improving the likelihood that assumptions are met, while avoiding additional statistical complications. Thus, cluster-based binning methods may offer increased statistical power over uninformed methods.

CLUSTER BINNING

This project developed a generalized workflow for clustering rare variants for the purpose of increasing statistical power. Given a set of objects, *cluster analysis* identifies the underlying structure by grouping similar objects and separating dissimilar objects. Applied to features, clustering can intelligently define feature bins using information about those features. Most

clustering algorithms also define disjoint clusters, such that objects belong to exactly one bin. In terms of feature binning, this results in drastically fewer, and statistically independent, bins to test. The framework does not restrict analysis to any particular clustering algorithm or statistical test, giving researchers the flexibility to customize the analysis to meet the requirements of their field, features, and preferences.

CHAPTERS

In chapter 2, I provide a review of the core clustering paradigms and dominant algorithms. The information should act as a guide for selecting the most appropriate clustering algorithm for a particular field or application. For clustering to be an effective alternative to uninformed binning, the clustering algorithm should reflect the feature characteristics and datatypes. Information that describes the features – *annotations* - is an important consideration when selecting a clustering algorithm, as not all clustering algorithms are applicable or ideal for certain datatypes. Additionally, uninformative or irrelevant annotations may misguide the cluster analysis, resulting in erroneous feature bins that do not inform statistical analysis.

In chapter 3, I introduce RVCLUST, which implements the rare variant clustering workflow as an R package. The package is implemented in R to facilitate easy integration into existing analyses. It is designed to be easily modifiable, offering an intuitive approach for integrating additional clustering algorithms and statistical tests. The distinct stages of rare variant cluster analysis are outlined conceptually and within the context of RVCLUST. Sample data is distributed with the package, and a demonstration of how to process that data is provided. The package itself is provided in Appendix A, and includes source code, documentation, and sample data that can be used to reproduce the analysis described in chapter 4.

In chapter 4, the RVCLUST framework is applied to the genetic analysis of rare variation in gene regulatory regions. Partitioning Around Medoids, a *k*-Medoids algorithm discussed in

chapter 2, is applied to the genomic position of rare genetic variants, with automated *k* selection. Variants are functionally annotated to constrain the clustering process with regard to domain knowledge. Lastly, the resultant clusters are tested for association with gene expression, and results are compared with an uninformed, near-exhaustive grouping method.

CHAPTER II

REVIEW OF CORE CLUSTERING PARADIGMS

INTRODUCTION

Clustering is a type of unsupervised machine learning that attempts to identify underlying structure in unlabeled data by identifying groups – *clusters* – whose members are similar to other members within the same group, and dissimilar to members in other groups. The approach is unsupervised, meaning there is no labeled training data with which to determine accuracy. Rather, clusters are judged by their own measures of fitness and the effectiveness of their results. Clustering is ideal for data mining, where information about each object is known, but nothing is known about how those objects relate to each other. Clustering has been used for a variety of applications across fields, including the discovery of cancer taxonomies, identification of complex disease subtypes, and the grouping of like-minded shoppers for targeted marketing.

Clustering strategies differ in the type of data they accept, the manner in which clusters are determined, and the structure of those clusters. Algorithms may be designed for numerical, categorical, or spatial data, while some specialize in handling mixed datatypes. Considerations regarding the clustering approach and structure may include tolerance for outliers, detection of arbitrarily shaped clusters, and whether the algorithm requires the number of clusters to be specified *a priori*. The most appropriate clustering algorithm for an application should be determined by matching the requirements of the application to the characteristics of the algorithm. At their most general, the core clustering paradigms are partitioning, hierarchical, and density, though these categories are neither comprehensive nor mutually exclusive. Each approach is described in detail, with sections committed to each. An additional section on

miscellaneous clustering paradigms touches on overlapping, modified, and extended methods that are particularly relevant to clustering for scientific or statistical purposes.

The goal of this review is to familiarize the reader with the core clustering paradigms and their principal algorithms, where such specification is appropriate. Specialized algorithms and implementations may provide additional benefits and limitations, but those details are beyond the intent of this review. Andreopoulos et al. (2009) provides an excellent review of modifications and extensions to standard approaches.

PARTITIONING

The goal of clustering is to group similar objects and separate dissimilar objects. Partitioning methods define these groups by subdividing the space in which they exist, and clustering those objects within the same partition. The general algorithm described in this chapter was first proposed by Stuart Lloyd, but the first publication came from E. W. Forgy in 1965. As such the algorithm is called the Lloyd-Forgy method. Partitioning methods typically require the user to specify *k*, the number of clusters. To begin, *k* objects in the data are selected, either randomly or using *a priori* knowledge or calculations, as the initial *centroids* of the *k* clusters. Objects are then assigned to the nearest centroid to generate an initial set of *k* clusters. This approach is inspired by Voronoi diagrams, which subdivide a space around a set of *k* points, where each cell contains a single point and all areas of the space that are closer to that point than any other points, as illustrated in figure 1.

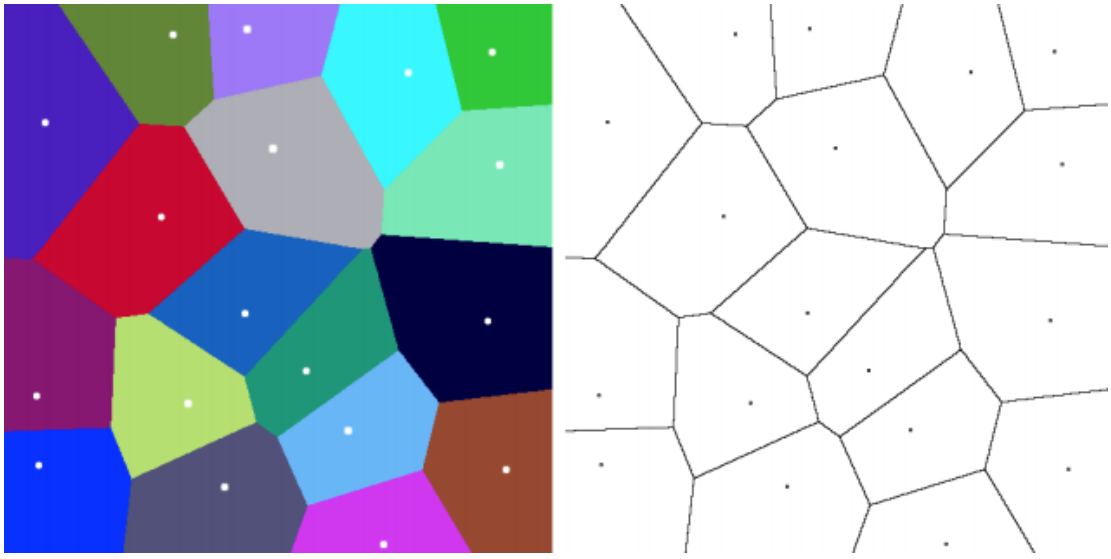

Figure 1: Voronoi diagrams partition a space according to the nearest point in that space. Each colored area is a cell that represents the space nearer to the centroid than any other point in the space. Image credit: Hoff III et al., 1999

Centroids are used to generate the Voronoi diagram, and all objects in the same cell represent one cluster. It is unlikely that the initial partitions are optimal; so new centroids are calculated *best represent* each cluster. What constitutes the *best representation* of a cluster differentiates the various partitioning algorithms, and is discussed in detail in those sections. Each iteration defines a set of *k clusters*, which refers to the current assignment of objects to *k* centroids. The iterative process attempts to refine these clusters until they overlap the *natural clusters*, the actual underlying structure that clustering aims to detect. A locally optimal solution is found when two consecutive iterations produce the same result, indicating convergence. The global optimum is then estimated by repeating the analysis with different initial centroids and determines the best clusters using some fitness measure. In the following sections, a variety of algorithms are presented that extend the partitioning paradigm.

k-Means

Developed by MacQueen (1967), *k*-Means defines the centroids of a cluster as the mean of all objects within that cluster. Figure 2 demonstrates how *k*-Means can identify natural clusters

even when centroid initialization is poor. The arithmetic mean is effective for numerical data, but is subject to distortion caused by outliers, and cannot be applied to data containing categorical values. Assigning the centroid to the arithmetic mean also decouples the centroid from actual values in the data, which allows it to drift into the empty space between a natural cluster and an outlier, or between two natural clusters, as can be seen in several iterations from figure 2.

k-Medoids

k-Medoids redefines a cluster centroid as the object most representative of the cluster. Specifically the medoid is the object within a cluster with the minimum total dissimilarity to other objects in that cluster (Kaufman & Rousseeuw 1990). The particular dissimilarity measure can vary, but is typically some form of distance function. Dissimilarity matrices are used extensively in hierarchical clustering, and are discussed in more detail in that section. Unlike an arithmetic mean, a medoid must be an actual object in the data, which improves outlier tolerance and prevents the centroid from entering the empty space between two natural clusters. In the worst case, an outlier will pull the medoid of a cluster to the object nearest the outlier. The effect is similar when two natural clusters assigned to the same centroid. The medoid must be an object in one of the natural clusters, and cannot drift between them like the arithmetic mean. This bias increases the likelihood that natural clusters will be assigned to more appropriate centroids in the following iteration. The differences between *k*-Means and *k*-Medoids can be seen in figure 2, where each identifies the natural clusters, but follow distinctly different paths.

Partitioning Around Medoids (PAM) (Kaufman & Rousseeuw 1987) is a popular implementation of *k*-Medoids, but does not scale well to large datasets. CLARA (Clustering Large Applications) (Kaufman & Rousseeuw 1990) is a modification to PAM that facilitates *k*-Medoids clustering of large datasets by using a random sample of the data to generate the centroids. Clusters are then defined for entire dataset by assigning objects to their nearest centroid. CLARANS (Clustering Large Applications based on Randomized Search) (Ng & Han

2002) is an extension of CLARA that resamples the dataset with each iteration of PAM. Selecting representative objects as centroids makes *k*-Medoids more appropriate for discrete values than *k*-Means, but does still require those values be numeric.

Figure 2: Comparison of *k*-Means and *k*-Medoids on identical data and initial centroids. Following different paths, each correctly identifies the three natural clusters. Note how the means glide between clusters, while medoids make distinct jumps.

k-Modes and *k*-Prototypes

k-Modes does not have the numeric restrictions of *k*-Means or *k*-Medoids. Centroids are chosen as the object in a cluster that appears most often. This is ideal for categorical data where similarity measures cannot be computed arithmetically. *k*-Prototypes is an extension of *k*-Modes that allows for the comparison of mixed datatypes using a dissimilarity measure that applies to both numerical and categorical data. (Huang 1998).

Silhouettes

All of the *k*-partitioning methods require users to specify the number of natural clusters *a priori*, which assumes domain knowledge that may not be available. Silhouettes are a method for determining how well an object was clustered, and maximization of the average silhouette width can be used to estimate the natural *k* for any partitioning method. The method first computes, for each object *i*, the average dissimilarity to all other objects in the same cluster, *a(i)*. It then defines an object's nearest neighboring cluster as the one that minimizes the average dissimilarity between all points in that cluster and itself, *b(i)*. The silhouette width, *s(i)*, is then calculated using the following formula (Rousseeuw 1987).

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

Because the denominator will always be greater than or equal to the absolute value of the numerator, the silhouette of an object will measure between *-1* and *1*. Positive values suggest correct cluster assignment, and negative values indicate the object is more similar to the neighboring cluster, suggesting poor cluster assignment. Silhouette widths near *0* represent uncertainty regarding the proper assignment of an object. Several silhouettes near *0* may indicate a poor selection of *k*. By repeating the analysis with different *k* values, the natural *k* is estimated as that which maximizes the average silhouette width.

HIERARCHICAL

Rather than partitioning objects into a set of disjoint, independent clusters, hierarchical clustering creates a binary tree containing clusters of clusters. Hierarchical clustering results are typically depicted as dendrograms, where each node represents a cluster of objects, as illustrated

in figure 3. Leaf nodes contain a cluster of one object, each parent node is a cluster of all objects in its child clusters, and the root is a cluster of all objects (Johnson 1967).

Agglomerative and Divisive

Algorithms for building this tree fall into two categories: agglomerative and divisive. Agglomerative algorithms begin with all objects as members of their own cluster, and iteratively merge clusters up to the root. Divisive algorithms work in reverse, starting with a cluster of all objects, and iteratively dividing those clusters until all clusters contain a single object. The decision on how to divide parent clusters is typically made by applying another clustering algorithm. Because hierarchical clustering algorithms produce binary trees, partitioning algorithms with *k=2* are a common choice.

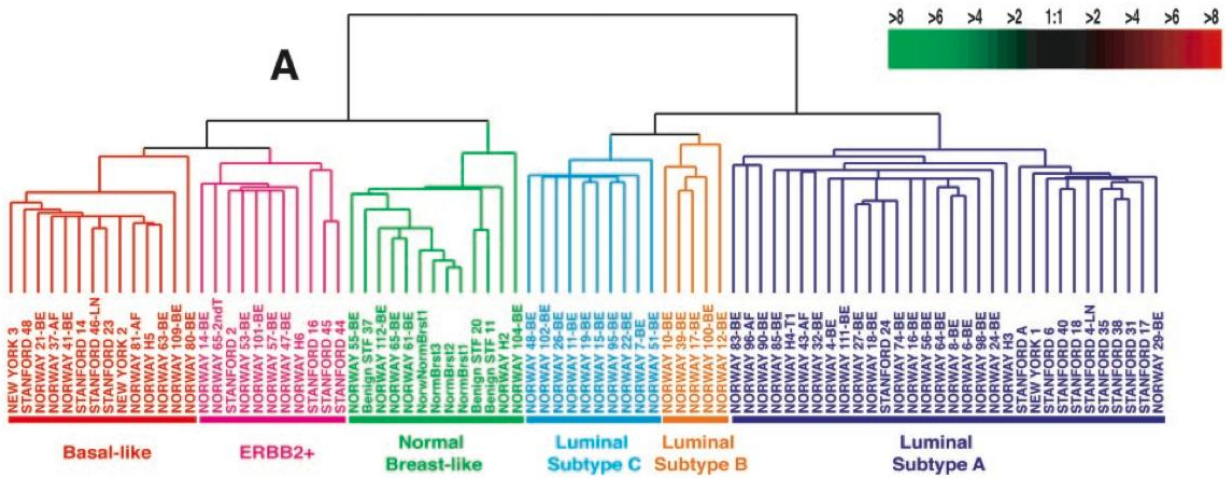

Figure 3: Dendrogram showing subdivisions of breast cancer tumors found by hierarchically clustering gene expression data. Colors highlight the tight clusters interpreted to be tumor subtypes. Image credit: Sørlie et al., 2001

Dissimilarity Matrices

Rather than clustering objects directly, hierarchical clustering algorithms typically use dissimilarity matrices, which contain the dissimilarity of each object in the data to every other object. The dissimilarity is calculated using a dissimilarity function. This is typically a distance

measure, though any metric may be used, as the purpose of the matrix is only to order the objects by dissimilarity (Johnson 1967).

Agglomerative clustering utilizes this dissimilarity matrix by identifying the two least dissimilar (most similar) objects in the set and grouping them to form a cluster. Dissimilarities for the two objects are removed from the matrix and replaced by dissimilarity measures for the cluster. Cluster dissimilarities are typically calculated using either the *single linkage* or *complete linkage* methods. In each, objects are treated as clusters of one object. S*ingle linkage* defines the dissimilarity between two clusters as the *minimum* pairwise dissimilarity between the member objects. With this approach, it is possible for two seemingly distinct clusters to merge because they have a single pair of similar objects. Alternatively, *complete linkage* selects the *maximum* pairwise dissimilarity. While single linkage is likely to generate large, sparse clusters, complete linkage typically produces small, dense clusters. A third alternative defines cluster dissimilarity as the mean pairwise dissimilarity (Sokal & Michaner 1958).

DENSITY

The clustering algorithms discussed to this point have been searching for clusters with low within-cluster dissimilarity and high between-cluster dissimilarity. Density clustering is an alternative approach that defines clusters by separating high-density regions from low-density regions, or noise. This is particularly appropriate for spatial data. Unlike partitioning methods, density clustering tolerates noise by not requiring that all objects be assigned to clusters. Furthermore, because cluster assignment is not driven by an object's distance from a centroid, density-based algorithms can detect arbitrarily-shaped clusters.

DBSCAN

A fundamental algorithm for density-based clustering, DBSCAN conceptualizes density clusters as consisting of core objects and border objects. Users specify a neighborhood radius and a minimum number of neighbors, where core objects are those that have at least the minimum number of neighbors within the given radius (neighborhood). Core objects make up the interior of a density cluster. Border objects do not meet minimum density requirements, but are directly density-reachable from a core object, meaning they exist within the neighborhood of a core object. The algorithm scans through unclassified objects, and locates those satisfying the density requirements. When objects are reclassified as core objects, the cluster is expanded recursively through its neighbors until all objects belonging to that cluster have been identified (Ester et al., 1996). Objects not meeting density requirements, and not qualifying as border objects, are classified as noise. User specification of the minimum density is DBSCAN's major limitation, expecting the user to know the cluster density *a priori*. DBSCAN also struggles with clusters of varying density, as the density threshold may be too strict to detect low-density clusters, or too lenient to separate high-density clusters.

OPTICS

OPTICS is an extension of DBSCAN that improves performance when densities vary between clusters. OPTICS takes the same input as DBSCAN, but orders objects by their minimum density-reachability. This ordered set of objects is used to generate a reachability-plot, from which clusters can be extracted by identifying reachability valleys (Ankerst et al., 1999). Figure 4 shows a density-reachability plot that contains four large clusters, one of which contains three nested clusters of higher density, yet another of which contains two nested clusters of even higher density. This complexity would go undetected in DBSCAN, either discarding the lower density clusters or merging the nested clusters into their parent clusters.

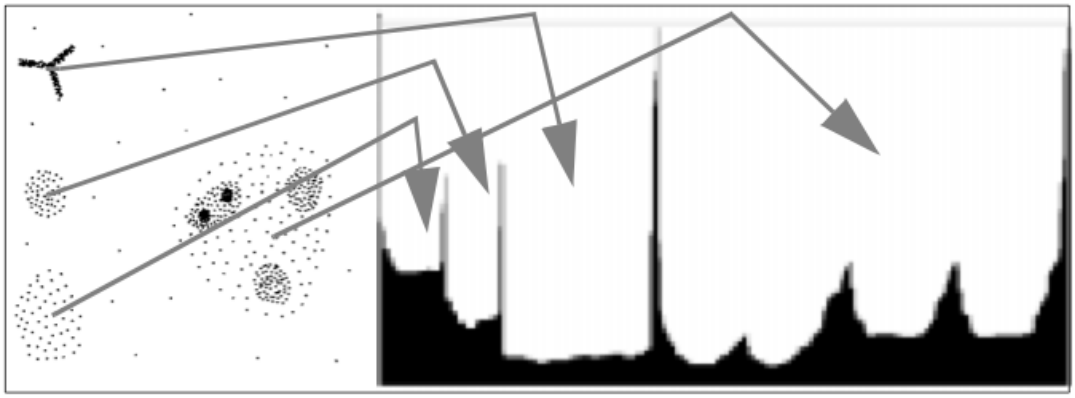

Figure 4: OPTICS uses reachability plots to identify valleys of varying density. Hierarchical density nesting can be seen as valleys within valleys. Deeper valleys indicate higher density, while high peaks indicate noise. Image credit: Ankerst et al., 1999.

MISCELLANEOUS CLUSTERING PARADIGMS

Clustering is a vast field of study, and a comprehensive review of all possible methods is beyond the scope of this chapter. However, there are some concepts that are of particular interest to the topic of this thesis that deserve mention, if only briefly. Some of these concepts incorporate properties of multiple paradigms, while others act as extensions or modifications to existing approaches. Fisher (2002) provides a more extensive review of the topics discussed below.

Constraints

Clustering is, in principle, the assignment of objects to groups without prior knowledge of group membership or classification. This does not consider that relationships between some objects might be known. Constraints can play an important role in situations where there is some understanding of which objects should be clustered together and which objects should not. Constrained Partitioning (COP) is a semi-supervised clustering technique for applying these constraints (Wagstaff et al., 2000)(Wagstaff et al., 2001). In COP, constraints are represented by a list of *must-link* and *cannot-link* relationships where objects that are known to cluster together are specified in the must-link list, and objects that are known to cluster separately are specified in the

cannot-link list. During partitioning, objects are assigned to their nearest cluster centroid such that none of the must-link or cannot-link constraints are violated. COP enforces hard constraints, so if a solution cannot be found such that the constraints are satisfied, the cluster analysis fails. Partial Closure *k*-Means (PCKMeans) is an alternative that applies soft constraints, which penalize constraint violations rather than prohibiting them. PCKMeans never fails to return a clustering solution (Zhang et al., 2008), making it preferable for data in which object relationships are suspected, but not known.

Conceptual Clustering

The development of constraint-based clustering was influenced by an earlier paradigm, conceptual clustering, which produces clusters that can be explained with conceptual or logical descriptions (Michalski 1980). Rather than defining constraints in terms of object relationships, logic-based conceptual clustering is constrained to clusters that can be expressed as logical descriptions. Similar to the hard constraints imposed by COP, conceptual clustering rejects a grouping if a suitable logical description cannot be found. Scientific applications may find this approach particularly useful for interpreting or explaining clustering results within the context of some scientific domain. A conceptual explanation of why particular objects cluster together may also inspire new hypotheses regarding object relationships and interactions.

Fuzzy and Probabilistic Clustering

Cluster assignment discussed so far has been either strict or hierarchical, such that objects belong absolutely to a cluster or hierarchy of clusters. Fuzzy clustering replaces the concept of absolute membership with degree of membership. Objects in a fuzzy clustering algorithm belong to all clusters *to some degree*. In real applications, objects rarely fall exclusively into one category or another, even within the same classification system. Fuzzy clustering allows objects with legitimately ambiguous membership to be accurately represented in the cluster analysis. The

fuzzy *c*-Means algorithm is a partitioning algorithm similar to *k*-Means, that follows the standard partitioning algorithm of determining cluster membership and adjusting the cluster centroid until convergence is reached. However, rather than assigning each object to a single cluster, each object is assigned a degree of membership in every cluster. Cluster centroids are then recalculated by taking the mean of all objects, weighted according to their membership in the cluster (Bezdek et al., 1984). Another algorithm, FANNY (Kaufman & Rouseeuw 1990), is a fuzzy implementation of the *k*-Medoids algorithm, PAM. Strict clusters can be derived from fuzzy clusters by assigning objects to the cluster in which they hold the greatest membership, which is how FANNY computes cluster silhouettes. Kaufman and Rousseeuw make the suggestion that very fuzzy objects (objects with near equal membership in multiple clusters) could be discarded before creating the hard partition to improve the silhouette of the solution.

An alternative to fuzzy clustering is probabilistic clustering, examples of which include the online hierarchical algorithm COBWEB (Fisher 1987) and its descendants (Fisher, 1996). The Bayesian probabilities utilized in AUTOCLASS (Cheeseman 1988) offers an interesting contrast to fuzzy clusters. While fuzzy clustering methods calculate for each object a degree of membership in every cluster, Bayesian methods calculate the probability that an object holds full membership in a single cluster.

DISCUSSION

Clustering is foundational to the workflow presented in chapter 1 for rare variant clustering. Rather than limiting researchers to a particular algorithm, the framework offers flexibility in algorithm selection. This chapter is intended as a general review of the major clustering paradigms, and not as a comprehensive review of the field. As such, some important clustering concepts and algorithms have been excluded. The information presented in this chapter

should provide researchers with a solid foundation from which to make informed decisions on the algorithm most appropriate for their application.

CHAPTER III

RVCLUST: AN R PACKAGE FOR RARE VARIANT CLUSTERING AND ANALYSIS

INTRODUCTION

Chapter 1 proposed that feature annotation and clustering were improvements over uninformed binning methods, and introduced a flexible workflow that included the identification, annotation, clustering, and statistical analysis of rare variants. RVCLUST (Rare Variant Clustering) is an R package designed to support all stages of this analysis. Initially developed for the analysis of rare genetic variants, RVCLUST approaches rare variation as a statistics problem with machine learning solutions. The utilities provided are applicable to any field studying the effects of rare variants. This chapter provides an overview of how RVCLUST implements each stage of the rare variant clustering workflow, followed by a sample analysis demonstrating its functionality. The data and annotations used in this demonstration are distributed with RVCLUST, which is included as appendix A.

IMPLEMENTATION

Initialization

The required data format for RVCLUST input is PEDMAP, an industry standard for genetic analysis. This format organizes information into two files: the PED file, which contains feature vectors and observation values, and the MAP file, which contains information about the features themselves. This format is ideal for rare variant analysis, so researchers in other fields should not see it as a functional limitation. Initialization of the *rvclustobject* involves loading this

data into memory, recoding each feature according to its expected and unexpected values, and calculating the frequency of the unexpected value. If the frequency is considered common (greater than 5%) then the feature is filtered to reduce the set to rare variants. Additional specifications may be specified during initialization to customize the analysis, including the option of burden testing, supplying covariate files, and setting minimum fitness thresholds for clusters to be statistical tested. Genetic annotations can also be requested at this stage, but the functionality is still in development and only relevant for genetic analysis.

Annotation

Feature annotation is an important aspect of informed clustering, and can be applied at two stages of analysis. When users initialize the *rvclustobject*, the primary feature annotation is read from the MAP file. In genetics, this file is used to map genetic variants to their genomic position. The annotation is required, but unrestricted in regards to what information is supplied. Genetics analyses can additionally employ the *annotate* feature of RVCLUST, however the functionality is still in development. Currently, this feature only provides chromatin state annotations for a single cell line, functionality necessary for reproducing the sample analysis included later in this chapter.

Cluster Analysis

A variety of clustering algorithms are available in R, and RVCLUST incorporates these into analysis through interface functions. These interfaces require an *rvclustobject* as input and return an *rvclustobject* as output, providing a standardized experience for users. Two interfaces are distributed with RVCLUST, but users are encouraged to create their own, which may be submitted for official inclusion in the distribution package. The default interfaces provide access to *pamk*, an implementation of *k*-Medoids with silhouette-based *k* selection available from the *fpc* package. Additionally, this interface can enforce hard constraints on chromatin state annotations

for genetic analysis. The second interface provides access to a novel clustering algorithm, *rvcluster*. This algorithm is a divisive hierarchical algorithm, using *pam* with *k=2* to divide clusters at each stage. *rvcluster* monitors the variation frequency of collapsed clusters and terminates when the frequency exceeds some threshold. The clusters returned from *rvcluster* will contain "trash clusters," where the division of a parent cluster improved the frequency of one child at the expense of another. Minimum fitness thresholds are important to use in conjunction with this approach to ensure that only fit clusters are statistically tested.

Statistical Analysis

Chapter 1 described two methods for merging the features in a bin for statistical analysis: collapsing tests and burden tests. Unless otherwise specified, RVCLUST uses a collapsing test to generate features from clusters. The disjunction of all features in a cluster is calculated for each observation in the data. With this method, if any binned feature shows variation, the bin is coded as varied. The alternative is burden testing, specified during the initialization stage. If burden testing is selected, a bin is represented by the count of all variations in the bin. Similar to the cluster stage, statistical tests are provided through *rvclustobject* interfaces. RVCLUST can apply any statistical test for which an interface is available. Linear regression, provided through *stats::lm*, is the default analysis, and the only interface distributed with RVCLUST. Additional tests can be incorporated into RVCLUST by creating an *rvclustobject* interface.

Software Design

RVCLUST is object oriented and distributed as an R package. Initialization defines an *rvclustobject*, which contains all necessary data and configuration parameters to perform each step of the analysis. RVCLUST is built on a set of interfaces, which accept the *rvclustobject* as input, perform the expected analysis, and return the updated *rvclustobject*. This workflow allows users to heavily modify the functionality of RVCLUST, while simplifying modifications to the

development of individual interfaces. Chapter 4 demonstrates the RVCLUST framework in the context of genetic analysis, associating rare genetic variants in gene regulatory regions to gene expression. A sample of that analysis is included as a demonstration of RVCLUST in the following section.

SAMPLE RUN

Full Analysis

Figure 1 demonstrates the simplicity of running the four stages of analysis in RVCLUST. PLINK is used to identify the unexpected value of all features, and also calculates the frequency of that value. This data is distributed with RVCLUST, and can be accessed by instantiating *rvclustobject* with NA parameters and *CHROMATIN* annotation.

rvobj <- rvclustobject(NA,NA,annotations=c("CHROMATIN"))

While figure 1 is a practical example of what a user would see during analysis, it does not describe the effects of each operation with any detail. The following sections provide a more in depth explanation of the four stages of analysis.

Annotation

The analysis described in chapter 4 constrains the cluster analysis by chromatin state, a functional annotation for gene regulatory regions. This annotation is required for the sample analysis is distributed with RVCLUST. Figure 2 shows a summary of the *variants* data frame after annotation, which appends the *CHROMATIN* column. *SNP*, *CHR*, and *POS* are field-specific columns, where *POS* is the primary annotation, genomic position. *MA* and *MAF* are abbreviations for the genetics terms *minor allele* and *minor allele frequency*, which code for a feature's unexpected value and frequency. The *CLUSTERID* column is described in more detail in the next section.

```
> rvobj <- rvclustobject('~','ENSG00000128699',annotations=c("CHROMATIN"))

@----------------------------------------------------------@
|        PLINK!       |     v1.07      |   10/Aug/2009     |
|----------------------------------------------------------|
|  (C) 2009 Shaun Purcell, GNU General Public License, v2  |
|----------------------------------------------------------|
|  For documentation, citation & bug-report instructions:  |
|        http://pngu.mgh.harvard.edu/purcell/plink/        |
@----------------------------------------------------------@

Skipping web check... [ --noweb ]
Writing this text to log file [ plink.log ]
Analysis started: Wed Mar  6 16:41:21 2013

Options in effect:
        --noweb
        --file /home/sivleyrm/ENSG00000128699
        --freq
        --allow-no-sex

12647 (of 12647) markers to be included from [ /home/sivleyrm/ENSG00000128699.map ]
Warning, found 149 individuals with ambiguous sex codes
Writing list of these individuals to [ plink.nosex ]
149 individuals read from [ /home/sivleyrm/ENSG00000128699.ped ]
149 individuals with nonmissing phenotypes
Assuming a quantitative trait
Missing phenotype value is -9
0 males, 0 females, and 149 of unspecified sex
Before frequency and genotyping pruning, there are 12647 SNPs
149 founders and 0 non-founders found
Writing allele frequencies (founders-only) to [ plink.frq ]

Analysis finished: Wed Mar  6 16:41:22 2013

> rvobj <- annotate(rvobj)
> rvobj <- pamk(rvobj)
> rvobj <- lm(rvobj)
```

Figure 5: Sample analysis using RVCLUST to load, annotate, cluster, and statistically test rare variants for association with the gene expression of ENSG000000128699. The use of RVCLUST interfaces simplifies the analysis.

```
> rvobj <- annotate(rvobj)
> summary(rvobj$variants)
     SNP             MA             MAF                CHR          POS
 Length:5080        0:   0   Min.   :0.003356   Min.   :2   Min.   :190135113
 Class :character   A:1336   1st Qu.:0.003356   1st Qu.:2   1st Qu.:190356093
 Mode  :character   C:1207   Median :0.010070   Median :2   Median :190632247
                    G:1173   Mean   :0.012797   Mean   :2   Mean   :190635759
                    T:1364   3rd Qu.:0.020130   3rd Qu.:2   3rd Qu.:190904560
                             Max.   :0.046980   Max.   :2   Max.   :191148763
   CHROMATIN       CLUSTERID
 Min.   :1.000   Min.   : 1.00
 1st Qu.:3.000   1st Qu.:25.00
 Median :5.000   Median :39.00
 Mean   :4.138   Mean   :32.98
 3rd Qu.:5.000   3rd Qu.:40.00
 Max.   :5.000   Max.   :40.00
```

Figure 6: Summary of the *variants* data frame after annotating the *rvclustobject* with chromatin state.

Cluster Analysis

Clustering is performed using the *rvclust::pamk* interface to *fpc::pamk* to the *rvclustobject*. This interface affects both the *variants* and *clusterinfo* data frames, both elements of the *rvclustobject*. In figure 2, the *CLUSTERID* column is generated by *pamk,* and indicates cluster assignment. The *clusterinfo* data frame is then populated with information about those clusters, as shown in figure 3. The information in this data frame is used to filter clusters that do not meet fitness thresholds, run statistical analysis, and define boundaries for replication studies.

```
> rvobj <- pamk(rvobj)
> summary(rvobj$clusterinfo)
   CLUSTERID      CHROMATIN              CHR         MIN.BP
 Min.   : 1.00   Length:40          Min.   :2   Min.   :190135113
 1st Qu.:10.75   Class :character   1st Qu.:2   1st Qu.:190425800
 Median :20.50   Mode  :character   Median :2   Median :190584869
 Mean   :20.50                      Mean   :2   Mean   :190625634
 3rd Qu.:30.25                      3rd Qu.:2   3rd Qu.:190857898
 Max.   :40.00                      Max.   :2   Max.   :191142326
     MAX.BP               SIZE
 Min.   :190176342   Min.   :   1.00
 1st Qu.:190444386   1st Qu.:   3.00
 Median :190622213   Median :   8.00
 Mean   :190670607   Mean   : 127.00
 3rd Qu.:190889227   3rd Qu.:  23.25
 Max.   :191148763   Max.   :1839.00
```

Figure 7: Summary of the *clusterinfo* data frame after *pamk* clustering. Forty clusters were generated, varying in size from 1 to 1839 rare variants. Strict constraints ensure that bins are chromatin state homogenous.

Statistical Analysis

Statistical analysis is the final stage of the rare variant clustering workflow. The *rvclustobject*, which now contains clustered *variant* and populated *clusterinfo* data frames, is passed to the linear regression interface. This interface collapses the clusters using whichever method was specified during initialization, produces a linear regression model for each bin using *stats::lm*. The statistical results are then appended to the *clusterinfo* data frame, as shown in figure 4.

```
> rvobj <- lm(rvobj)
> summary(rvobj$clusterinfo)
   CLUSTERID      CHROMATIN              CHR         MIN.BP
 Min.   : 1.00   Length:40          Min.   :2   Min.   :190135113
 1st Qu.:10.75   Class :character   1st Qu.:2   1st Qu.:190425800
 Median :20.50   Mode  :character   Median :2   Median :190584869
 Mean   :20.50                      Mean   :2   Mean   :190625634
 3rd Qu.:30.25                      3rd Qu.:2   3rd Qu.:190857898
 Max.   :40.00                      Max.   :2   Max.   :191142326

     MAX.BP               SIZE              PVALUE            EFFECT
 Min.   :190176342   Min.   :   1.00   Min.   :0.00000   Min.   :0.0000041
 1st Qu.:190444386   1st Qu.:   3.00   1st Qu.:0.06023   1st Qu.:0.0007680
 Median :190622213   Median :   8.00   Median :0.59383   Median :0.0021875
 Mean   :190670607   Mean   : 127.00   Mean   :0.47755   Mean   :0.0353969
 3rd Qu.:190889227   3rd Qu.:  23.25   3rd Qu.:0.74300   3rd Qu.:0.0267557
 Max.   :191148763   Max.   :1839.00   Max.   :0.98051   Max.   :0.5000000
                                       NA's   :1
   R.SQUARED         ADJ.R.SQUARED
 Min.   :0.0000000   Min.   :-0.006799
 1st Qu.:0.0006294   1st Qu.:-0.006030
 Median :0.0018313   Median :-0.004600
 Mean   :0.0228969   Mean   : 0.016420
 3rd Qu.:0.0229234   3rd Qu.: 0.016277
 Max.   :0.2296135   Max.   : 0.224373
```

Figure 8: Summary of the *clusterinfo* data frame after association testing using linear regression. The *PVALUE* summary indicates that some bins were found to be significant.

DISCUSSION

The framework provided by RVCLUST is a flexible, cluster-based approach to rare variant analysis. Binning methods sacrifice resolution to increase the frequency of rare events in Boolean features, which in turn improves statistical power. Clustering can intelligently define feature bins using domain knowledge, resulting in drastically fewer, and statistically independent, bins compared to uninformed binning methods. RVCLUST provides a set of interfaces for all stages of analysis, and allows users to easily develop additional interfaces to customize their analysis. The package was designed with this in mind, and encourages crowd sourcing to expand the set of available interfaces. The project is open source and available at *github.com/bushlab/rvclust*, where new interfaces can be submitted for general distribution. The package itself is included as appendix A, and more detailed technical information may be found in the package documentation.

CHAPTER IV

KNOWLEDGE-CONSTRAINED K-MEDOIDS CLUSTERING OF REGULATORY RARE ALLELES FOR BURDEN TESTS

PREFACE

The entirety of this chapter, following this Preface, is a reformatted, published conference paper, of which I am the first author (Sivley, Fish, & Bush 2013), used with the permission of my coauthors and allowed by the copyright license. My contribution to this work includes all methods development and implementation of cluster-based analysis. This includes the decision to explore cluster-based analysis, research and selection of the clustering method, interface development for clustering and statistics packages, including the extension of those packages to enforce biological constraints, and statistical testing of the resultant clusters. This study is included to demonstrate the practical application of RVCLUST in statistical analysis.

INTRODUCTION

Numerous studies have been published illustrating the association of commonly occurring genetic variants to traits of interest in humans (Hindorff et al., 2009), and to changes in gene expression (Veyrieras et al., 2008). Recent technological advances in sequencing technology have enabled the study of rare variation – single base-pair changes in DNA that occur at less than 5% frequency in a population (Durbin et al., 2010). Typical genetic association studies rely on linear or logistic regression models to contrast the phenotype of interest across genotype categories based on a single variant (i.e. AA [25%], Aa [50%], and aa [25%]). Statistical power for these studies is directly related to the frequencies of these genotype categories, and lower

frequency variants often have extremely low power to detect associations using these methods because most individuals in the study do not have the rare variant (i.e. AA [98%], Aa [1.8%], and aa [0.2%]).

Multiple methods have been proposed to address the issues of statistical power (Bansal et al., 2010), all of which rely on grouping rare variants together either by biological function or physical proximity in the genome. The vast majority of these statistical methods provide users with the flexibility to specify the genomic region they wish to use for grouping variants together. In practice, variants are typically collapsed within gene regions under the hypothesis that a variants influence disease by changing coding DNA that impacts protein function in some way. However, recent publications by the ENCODE project have shown that the vast majority of previously identified genetic associations are non-coding and regulatory in nature (Schaub et al., 2012).

Currently, non-genic approaches to group rare variants include a simple sliding window approach (Lawrence et al., 2010) or collapsing variants within regions defined by experimental data, such as the ENCODE annotations. Sliding window approaches require millions of statistical tests which are highly correlated. The large number of tests makes determining the false positive or false discovery rate of the analysis challenging. Collapsing variants within putative regulatory regions may produce windows that are too small to capture variants to provide a powerful test. This approach also assumes that the genomic locations of regulatory regions are well-defined – an unlikely assumption for many Chromatin Immuno-Precipiation (ChIP) experiments (Mendenhall et al., 2012). Therefore, new methods for defining non-genic windows for statistical analysis are needed.

In this work, we apply k-medoids clustering to leverage both physical proximity and biological function with the goal of defining groups of rare variants for statistical analysis. We use a single source of putative biological function – a prediction of genome function based on chromatin state – and refine groupings using physical proximity in the genome. We apply this

clustering method to generate rare variant groupings and evaluate the impact of these grouped variants on gene expression traits. Results from our clustering-based approach are compared with a traditional sliding window approach.

METHODS

Data

Publically available datasets with phased haplotype information and whole-genome gene expression data on 1000 Genomes samples were used (Durbin et al., 2010). There were 149 independent, multi-ethnic individuals, consisting of 32 CEPH (CEU) and 37 Yoruba (YRI) parental samples, and 41 Chinese (CHB) and 39 Japanese (JPT) unrelated individuals. Phased haplotype data was obtained from the imputation reference panels for MaCH software (1000G Phase 1 version 3 MaCH panels) and was based upon 1000 Genomes Phase 1 integrated genotype calls and included singleton variants (Li et al., 2010). For gene expression data, we accessed normalized gene expression data from (Veyrieras et al., 2008) (available online: http://eqtnminer.sourceforge.net/), which was generated using Illumina human whole-genome expression arrays (WG-6 version 1) on lymphoblastoid cell lines from each of the 149 individuals. Expression data was first normalized by quantile normalization within replicates, and then was median normalized across individuals. Additionally, we applied Gaussian quantile normalization for the test genes within each population, in order to account for population differences in gene expression. This normalization was congruent with the original normalization performed in Veyrieras et al., (2008). For each of the selected genes, we extracted genotypes in the *cis*-regulatory region (500KB upstream of the transcriptional start site and 500KB downstream of the transcriptional end site).

Domain Knowledge

We used classification results from a published study of chromatin marks [9] to guide our cluster analysis. This study used ChIP data to identify methylation and acetylation modifications to histone proteins throughout the genome for nine cell lines. These patterns form the *histone code* (Rando et al., 2012), and were classified using a multivariate Hidden Markov Model into 15 states, which we loosely grouped into promoter, enhancer, insulator, and transcribed regions. Because our analysis was focused exclusively on gene expression in lymphoblastoid cell lines, we used chromatin state classifications generated for the GM12878 lymphoblastoid cell line. This data is available via the ENCODE project website through the UCSC genome browser (http://genome.ucsc.edu/ENCODE/). By guiding our cluster analysis with this data, we hypothesize that genetic variation within similar chromatin states should be grouped together.

Gene Selection

To compare the two methods across a variety of different regulatory architectures, four genes were selected from a group of genes previously identified as having collections of rare variants functioning as cis-eQTLs, based upon a genome-wide collapsing analysis (unpublished data). Each gene selected represents a potentially unique regulatory architecture, based upon the functional annotation of rare variants which were within the significant regions. Rare variants within significant regions could be identified as disrupting a transcription factor binding site (*ORMDL1*), being present in a ChIP peak (*NUDT22*), or having no functional annotation whatsoever (*FAM154B*). A potential confounder to this study is the presence of common eQTLs in significant regions. A compilation of known common eQTLs was used to determine that none of the above genes had a common eQTL in the previously identified significant regions. To interrogate the effects of common eQTLs on the analysis, *DYPSL4* was also selected, which

contained three common eQTLs in the previously identified significant region in addition to rare variants affecting transcription factor binding sites.

Cluster-based Analysis

Constrained Partitioning (COP) is a method by which partial knowledge can be introduced into a clustering algorithm, making it a semi-supervised method. Constraints allow for otherwise uninformed clustering methods to include background knowledge of a particular domain. Typically, COP is provided with a list of must-link constraints and cannot-link constraints, which dictate which observations must and cannot be placed in the same cluster.

In our implementation, we allow for an initial classification of chromatin state SNPs surrounding a gene. This classification acts as a must-link constraint for all observations in a class, and a cannot-link constraint for all observations of differing classes. We then apply Partitioning Around Medoids (PAM) to subdivide these SNPs according to their base position. PAM divides the data into k clusters, where k is specified *a priori* (Kaufman & Rousseeuw 1987). To choose an optimal k, we ran PAM multiple times with increasing k and select k such that it maximizes with average silhouette width of the resultant clusters. The choice of k is made for each initial classification and the original classes do not need to be partitioned into the same number of clusters.

With our rare variants clustered, we then performed a rare variant burden test, which collapses the data into a single variable, indicating the number of rare variants within that cluster. For each cluster, linear regression was used to determine the significance of association between the clustered rare variants and gene expression. This implementation was done entirely in R.

Sliding Window Analysis

A rare variant burden test with sliding windows was performed on the test genes. For each gene, the region tested consisted of 500KB both up and downstream, in addition to the gene

itself. In this region, a 5KB sliding window was used, such that each SNP served as the start point for a window. All rare variants in this 5KB region were used to determine the burden of rare variants. Only windows with at least one rare variant detected were included in analysis. For each window, a linear regression was performed between the number of rare variants present within a region for each individual and the gene expression level. This is slightly different from the analysis used to select the genes, in which individuals were placed into a binary category of either having a rare variant or not – a *collapsing* test (Li et al., 2008).

Determination of Significance

The best practice for the statistical analysis of sliding windows is a current topic of debate. To place these results in the context of standard genetic analysis guidelines, both a Bonferroni correction and a False Discovery Rate (FDR) analysis were performed (Storey et al., 2003). Each gene was analyzed independently in both the Bonferroni and FDR (FDR = 0.05) analyses. In the Bonferroni correction analysis, the number of clusters present in each gene is used to set the gene-specific significance threshold for cluster data. For the sliding window analysis, the number of windows set the gene-specific significance threshold. After being identified as significant, all overlapping windows were merged to form a significant 'signal' in the sliding window analysis.

Visualization

We visualized the results from both the sliding window and cluster analyses in a single plot using the R package ggplot2 (Wickham 2009). For the sliding window analysis, the midpoint chromosome position of each 5KB window is plotted relative to the $-\log_{10}$ of the regression p-value to generate a *Manhattan* plot. We used loess to fit a smooth curve to these data points using the stat_smooth function with a span parameter of 0.2. Results from the cluster analysis are shown as horizontal bars (to illustrate the span of the cluster) plotted relative to the $-\log_{10}$ of the

regression p-value, color coded by chromatin state. Note that some clusters are too small to be seen on these plots.

RESULTS

Gene Region Results

Visual comparisons of sliding window and cluster analysis approaches are provided in figure 1. *ORMDL1* best illustrates the potential of this method. A highly significant effect is seen from an enhancer cluster which overlaps with the strongest effect from the sliding window analysis. *NUDT22* also shows a strong effect of a large enhancer cluster which spans the best sliding window effect. For both these genes the clustering results correlate well with the loess curves, capturing the 'shape' of the regional effect. The cluster analysis shows less utility for *DYPSL4*, a gene with complex common eQTL effects, and *FAM154B,* a gene with no obvious regulatory mechanisms. For these genes, the method clustered together distant variants within insulator elements creating single clusters containing variants at great distances; these clusters do not reflect the domain knowledge well. We plan to refine the algorithm to include additional constraints limiting the physical distance separating rare variants within potential clusters.

Bonferroni Correction

The summary of significant genomic regions with a Bonferroni corrected analysis is presented in Table 1. Similar numbers of significant genomic regions are returned by both the sliding window and cluster analysis. In both methods, *DYPSL4* failed to result in significant results. In the case of *ORMDL1*, both clustering and sliding window analysis each resulted in one unique significant region which was not overlapping. All other significant regions overlapped with a region identified in the other test. In *NUDT22*, all significant signals identified by sliding window analysis overlapped with significant clusters. Cluster analysis additionally resulted in two

unique significant regions. None of the significant regions identified in *FAM154B* overlapped between the sliding window analysis and the cluster analysis.

GENE	Bonferroni Threshold for Cluster Analysis	Number of Significant Clusters	Bonferroni Threshold for Sliding Window Analysis	Number of Significant Windows from Sliding Window Analysis
ORMDL1	0.001250	6 of 40	3.95476×10^{-6}	604 of 12,643
NUDT22	0.001351	5 of 37	4.64857×10^{-6}	26 of 10,756
DYPSL4	0.001282	0 of 39	3.16476×10^{-6}	0 of 15,799
FAM154B	0.001351	3 of 37	6.38162×10^{-6}	32 of 7,835

Table 1: Number of significant genomic regions detected using both clustering and sliding window analysis with a Bonferroni correction for multiple testing.

False Discovery Rate Correction

The significant genomic regions with a FDR (FDR = 0.05) corrected analysis are presented in Table 2. All the regions identified as significant with the Bonferroni correction were identified with the FDR correction as well. One unique cluster was identified with FDR analysis in both *ORMDL1* and *NUDT22*. A dramatic increase was observed in the number of signals identified as significant in the sliding window analysis. For *ORMDL1*, *NUDT22*, and *FAM154B*, all significant clusters overlapped with regions identified as being significant by sliding window analysis. In the case of *DYPSL4*, clustering failed to identify any significant regions, whereas sliding window analysis identified two genomic regions as significant. Sliding window analysis identified a total of 28 unique genomic regions as significant in these genes.

GENE	Threshold for Cluster Analysis FDR = 0.05	Number of Significant Clusters	FDR = 0.05 Threshold for Sliding Window Analysis	Number of Significant Windows from Sliding Window Analysis
ORMDL1	0.001346812	7 of 40	0.007989149	2021 of 12,643
NUDT22	0.006583255	6 of 37	0.007619227	1126 of 10,756
DYPSL4	NA*	0 of 39	0.000434797	126 of 15,799
FAM154B	0.001213077	3 of 37	0.006232502	628of 7,835

Table 2: Number of significant genomic regions detected using both clustering and sliding window analysis with an FDR=0.05 correction for multiple testing. *There are no p-values < 0.05, making it impossible to calculate the FDR = 0.05 threshold.

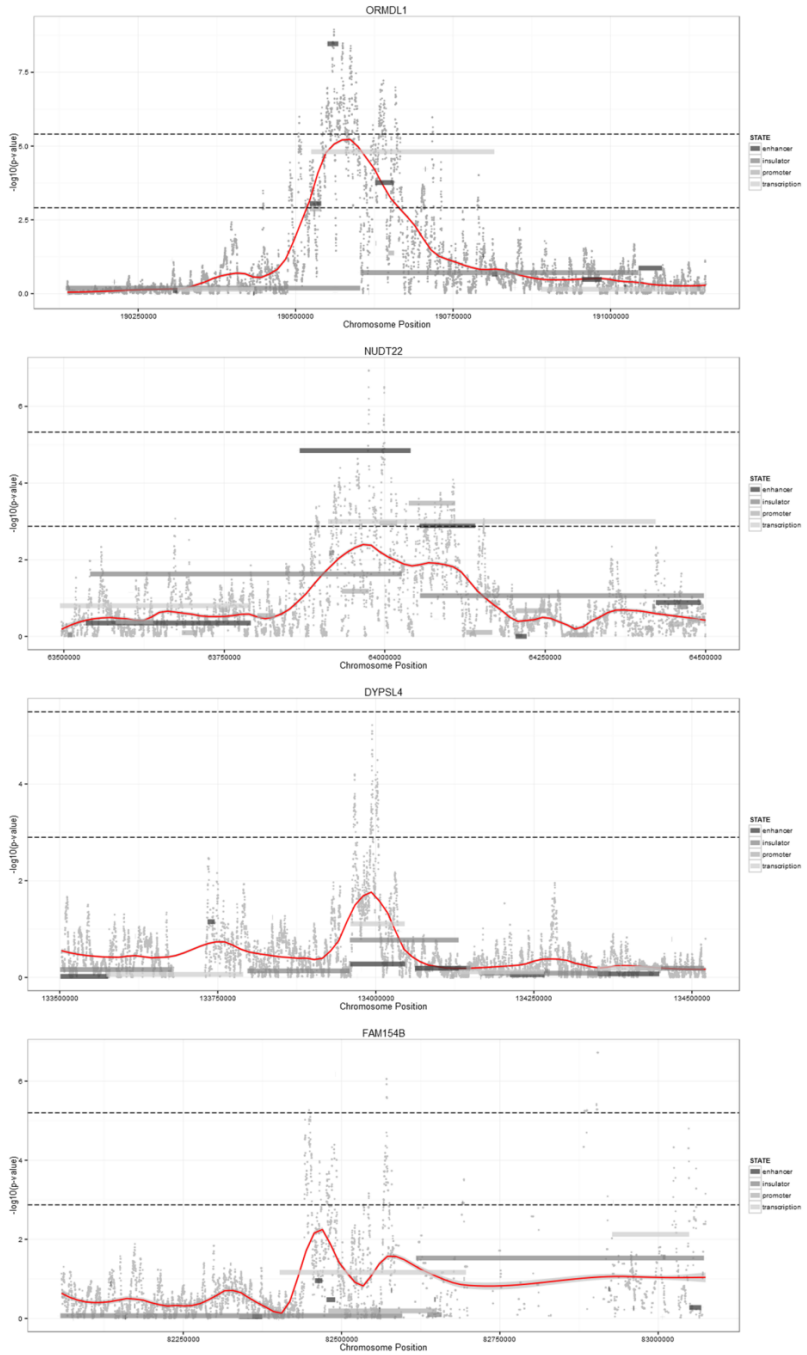

Figure 9: Manhattan plot of window midpoints (points), variant clusters (bars) by significance with loess fit (red line, loess span = 0.2) of window midpoint by significance

DISCUSSION

Our results indicate that informed clustering of rare variants using regulatory annotations can dramatically reduce the number of statistical tests, reducing the multiple testing burden for rare variant analysis, thus increasing overall power. Obviously, this approach will perform best when the underlying assumption of the method holds true; that influential variants fall within regulatory regions, as illustrated in the *ORMDL1* gene.

A great strength of this approach is that the clustering is independent of statistical analysis, and can be coupled with various methods, such as the Sequence Kernel Association Test (SKAT) (Wu et al., 2011) or KBAC (Liu et al., 2010). Because the method is unsupervised, there are no over-fitting concerns in the association analysis, and standard statistical assumptions of these tests are not violated. The cluster method could also be informed by statistical power calculations of the coupled association test (or other testing assumptions), allowing clusters of rare variants to be optimized to improve the overall power of the analysis. Finally, in this study we have used chromatin state data to guide cluster formation, however numerous other genomic annotations could be applied simultaneously to intelligently design functional clusters of rare variants. As ENCODE and other projects continue to expand our understanding of gene regulation, methods that can leverage this data for analysis will become ever more important.

CHAPTER V

CONCLUSIONS

REVIEW

The study of rare genetic variation, as well as any field studying the effects of rare events, is hindered by the statistical complications caused by that infrequency. In most feasibly sized datasets, the number of occurrences of a rare event is insufficient to define a trend, or alternative distribution of expected outcomes when the rare event has occurred. Without this trend, we cannot detect effects with statistical confidence, and so our results typically fail to meet significance thresholds.

Currently, different fields address the problem in their own way, such as the sliding window approach in genetics. The concept of binning features and merging their events to increase frequency is not conceptually tethered to a particular field, but most implementations of the idea are. Many of these methods also introduce additional statistical issues that complicate the interpretation and analysis of their results. Uninformed binning methods also fail to meet some assumptions, the most important of which is that the features in a bin have the same direction of effect on the observed variable.

Cluster analysis was proposed as an alternative binning strategy that would incorporate feature information to guide the clustering algorithm. Bins defined using cluster analysis have a quantifiable justification for why certain features were binned together. When the information used to cluster the features is relevant, or predictive, clustering increases the likelihood that the direction of effect assumption is met.

The rare variant clustering workflow was presented as a flexible guideline for rare variant analysis that used clustering algorithms to bin features. This workflow did not restrict users to any

particular clustering algorithm, collapsing method, or statistical analysis, ensuring that researchers in any field could use it to guide their analysis. The clustering algorithm selected can play a crucial role in determining the content and structure of the resultant bins, so a review of core clustering paradigms was included. To support the workflow, I developed RVCLUST, an R package implementation that facilitates all steps of the workflow by providing a collection of interfaces to common algorithms and statistical tests. Allowing users to specify the clustering algorithm, collapsing method, and statistical analysis, retains the flexibility of the conceptual workflow. If the desired functionality is not available, RVCLUST is designed such that additional interfaces can be added with minimal effort.

The RVCLUST workflow was applied to a genetics application studying rare genetic variants in gene regulatory regions, and their effects on gene expression. The genetic variants were annotated with chromatin state, clustered around genomic position using *k*-Medoids, and constrained to chromatin state-homogenous clusters. Sliding window analysis was applied to the same data for comparison. The rare variant clustering workflow identified the same significant effects in the same regions, but reduced the number of bins, and by extension the number of statistical tests, by two orders of magnitude.

IMPROVEMENTS

Theoretical

Cluster analysis as a binning method provides all of the statistical advantages gained through binning, without incurring the additional statistical complications introduced by other methods. The frequency of rare events is increased by merging multiple features into a single, composite feature representing the events of all features in the bin. By extension, the increased frequency allows for the detection of significant effects that could not otherwise be detected with confidence. Cluster analysis produces disjoint bins, eliminating any statistical dependency

concerns. Additionally, many clustering algorithms allow for the specification of how many clusters are generated, which can be useful when the statistical penalty of multiple test correction is of particular importance.

Experimental

When compared with a sliding window analysis, the rare variant clustering workflow identified the same significant effects, but did so with drastically fewer bins, none of which shared rare variants. Additionally, interpretation of the results is aided by cluster analysis defining bins using field-specific information. The composition of a bin, in terms of the field-specific similarities the features share, can form the basis of why those features affect the observed variable, and why they affect it in the way that they do. Also included in that analysis is a gene with no suspected regulatory regions, which conflicted with our annotations, which were regulatory predictions. The cluster analysis failed to define any bins with a significant effect on the expression of that gene, demonstrating that the effectiveness of cluster analysis is dependent on relevant and predictive annotations.

Additional

The incorporation of clustering comes with additional benefits not captured by a simple comparison with existing methods. Cluster analysis is not a particular binning method designed to solve a single problem, but an entire field of study that has been developing over decades. In addition to generating informed bins through a variety of algorithms, cluster analysis also provides methods for measuring the fitness of a cluster, determining how well a particular objects fits into its cluster, and strategies for determining how many clusters to define in disjoint approaches. Rather than a field-specific solution, rare variant clustering workflow is supported by decades of work in an established and well-developed field.

FUTURE DIRECTIONS

Development of the rare variant clustering workflow was inspired by genetics and applied to genetics, but it is designed for general use. Future work would apply the existing framework to other fields that study rare events. Possible applications might include the study of rare news events and how they affect stock prices, or uncommon features in a house and how they affect housing prices.

The goal of any binning method is to increase the frequency of rare events to improve the likelihood of detecting significant effects. This likelihood has been previously described as statistical power. If a clustering algorithm were to consider statistical power during the clustering process, it might be possible to optimize which features are binned to detect effects without including unnecessary features that might not be as informative as others. It might also be interesting to incorporate the frequency of an event during the clustering process, perhaps suggesting events that occur with similar frequencies are more likely to have similar effect sizes.

The major cost of any binning method is the ability to attribute significant effects to a single event. However, with informed cluster analysis, this limitation might discover predictive features not otherwise considered. While cluster analysis does not label its output, users can often abstract object similarities into a descriptive label. It may be the case that in those situations where feature binning is most effective, the feature definitions are too strict, and do not capture the truly predictive aspect. When these overly specific features are grouped together, perhaps an examination of the composite feature would reveal that it implicates a more comprehensive domain concept as the actual predictive feature.

The RVCLUST software was designed encourage constant development. Continued efforts to extend the functionality of RVCLUST to include more clustering algorithms, novel collapsing methods, and more options for statistical analysis will increase its relevance and

effectiveness in studying the effects of rare events. The project is designed around an open source model, and user contributions are encouraged to better capture the needs of the community.

CONCLUSIONS

The rare variant clustering workflow is an effective approach to binning rare variants to detect effects. The incorporation of cluster analysis provides the same benefits as field-specific binning methods, while remaining field-independent and incurring none of the statistical complications. Cluster analysis provides more control over the number of bins generated and the methods by which they are generated. The disjoint bins generated by most clustering algorithms offer cleaner and more compelling statistics, supported by domain-specific justification for bin definitions. Additionally, clustering metrics like object similarity and cluster fitness can be leveraged to customize analysis and improve results. RVCLUST provides all of these benefits in a simplified software package, which is both easy to use and easy to modify, ensuring its utility in whichever field it is applied

Appendix A

RVCLUST SOFTWARE

The RVCLUST R package, as described in chapter three, is submitted as appendix A. The complete repository, including source code, documentation, and sample data, is attached. The most up to date revision can also be found at *github.com/bushlab/rvclust*.

REFERENCES

Andreopoulos, B., An, A., Wang, X., & Schroeder, M. (2009). A roadmap of clustering algorithms: finding a match for a biomedical application. *Briefings in Bioinformatics*, 10(3), 297-314.

Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2), 49-60.

Bansal, V., Libiger, O., Torkamani, A., Schork, N.J. (2010). Statistical analysis strategies for association studies involving rare variants. *Nature reviews. Genetics*. 11, 773–85.

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2), 191-203.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., Freeman, D. (1988). Autoclass: a Bayesian classification system. In *Proceedings of the Fifth International Conference on Machine Learning (Ann Arbor, MI)*, pp. 54-64.

Durbin, R.M., Altshuler, D.L., Abecasis, G.R., Bentley, D.R., Chakravarti, A., et al. (2010). A map of human genome variation from population-scale sequencing. *Nature*. 467, 1061–1073.

E.W. Forgy (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics* 21, 768–769.

Ernst, J., Kheradpour, P., Mikkelsen, T.S., Shoresh, N., Ward, L.D., Epstein, C.B., Zhang, X., Wang, L., Issner, R., Coyne, M., Ku, M., Durham, T., Kellis, M., Bernstein, B.E. (2011). Mapping and analysis of chromatin state dynamics in nine human cell types. *Nature*. 473, 43–9.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, Vol. 1996, 226-231. AAAI Press.

Fisher, D. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine Learning* 2: 139-172.

Fisher, D. 1996. Iterative Optimization and Simplification of Hierarchical Clusterings. *Journal of Artificial Intelligence Research*. 4: 147-148

Fisher, D. (2002). Conceptual Clustering. In W. Klosgen and J. Zytkow (eds.), *Handbook of Data Mining and Knowledge Discovery, Oxford University Press*, 388--396, Chapter 16.5.2.

Hindorff, L.A., Sethupathy, P., Junkins, H.A., Ramos, E.M., Mehta, J.P., Collins, F.S., Manolio, T.A. (2009). Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. In *Proceedings of the National Academy of Sciences of the United States of America* 106, 9362–9367.

Hoff III, K. E., Keyser, J., Lin, M., Manocha, D., & Culver, T. (1999). Fast computation of generalized Voronoi diagrams using graphics hardware. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 277-286. ACM Press/Addison-Wesley Publishing Co.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3), 283-304.

Johnson, S. C. (1967). Hierarchical clustering schemes. Psychometrika, 32(3), 241-254.

Lawrence, R., Day-Williams, A.G., Elliott, K.S., Morris, A.P., Zeggini, E. (2010). CCRaVAT and QuTie-enabling analysis of rare variants in large-scale case control and quantitative trait association studies. *BMC bioinformatics*. 11, 527.

Li, B., Leal, S.M. (2008). Methods for detecting associations with rare variants for common diseases: application to analysis of sequence data. *American journal of human genetics*. 83, 311–21.

Li, Y., Willer, C.J., Ding, J., Scheet, P., Abecasis, G.R.: MaCH (2010). Using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genetic epidemiology*. 34, 816–34.

Liu, D.J., Leal, S.M. (2010). A novel adaptive method for the analysis of next-generation sequencing data to detect complex trait associations with rare variants due to gene main effects and interactions. *PLoS genetics*. 6, e1001156.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* Vol. 1, No. 281-297, p. 14.

Mendenhall, E.M., Bernstein, B.E. (2012). DNA-protein interactions in high definition. *Genome biology*. 13, 139.

Michalski, R. 1980. Knowledge acquisition through conceptual clustering: a theoretical framework and algorithm for partitioning data into conjunctive concepts. *International Journal of Policy Analysis and Information Systems* 4: 219-243.

Ng, R. T., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 14(5), 1003-1016.

Rando, O.J. (2012). Combinatorial complexity in chromatin structure and function: revisiting the histone code. *Current opinion in genetics & development*. 22, 148–55.

Rousseeuw, P.J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*. 20: 53-65.

Rousseeuw, P. J., & Kaufman, L. (1987). Clustering by Means of Medoids. *Statistical data analysis based on the L1-norm and related methods*, 405.

Rousseeuw, P. J., & Kaufman, L. (1990). Finding groups in data: An introduction to cluster analysis. *John, John Wiley & Sons*.

Schaub, M.A., Boyle, A.P., Kundaje, A., Batzoglou, S., Snyder, M. (2012). Linking disease associations with regulatory information in the human genome. *Genome research*. 22, 1748–59.

Sivley, R. Michael, Fish, Alexandra E., Bush, William S. (2013). Knowledge-constrained k-Medoids Clusters of Regulatory Rare Alleles for Burden Tests. *Lecture Notes in Computer Science, vol 7833* (In Press).

Sokal, R. R., & Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *Univ. Kans. Sci. Bull*., 38, 1409-1438.

Sørlie, T., Perou, C. M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., ... & Børresen-Dale, A. L. (2001). Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. In *Proceedings of the National Academy of Sciences*, 98(19), 10869-10874.

Storey, J.D., Tibshirani, R. (2003). Statistical significance for genomewide studies. In *Proceedings of the National Academy of Sciences of the United States of America*. 100, 9440–5.

Veyrieras, J.-B., Kudaravalli, S., Kim, S.Y., Dermitzakis, E.T., Gilad, Y., Stephens, M., Pritchard, J.K. (2008). High-resolution mapping of expression-QTLs yields insight into human gene regulation. *PLoS genetics*. 4, e1000214.

Wagstaff, K., & Cardie, C. (2000). Clustering with instance-level constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 103–1110.

Wagstaff, K., Cardie, C., Rogers, S., & Schroedl, S. (2001). Constrained K-means Clustering with Background Knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 577-584.

Wickham, H. (2009). ggplot2: elegant graphics for data analysis. *Springer New York.*

Wu, M.C., Lee, S., Cai, T., Li, Y., Boehnke, M., Lin, X. (2011). Rare-variant association testing for sequencing data with the sequence kernel association test. *American journal of human genetics*. 89, 82–93.

Zhang, S., & Wong, H. S. (2008). Partial closure-based constrained clustering with order ranking. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 1-4. IEEE.