An Efficient Algorithm for Clustering Genomic Data

A thesis submitted to the

Division of Research and Advanced Studies

of the University of Cincinnati

in partial fulfillment of the requirements for the degree of

Master of Science

in the Department of Electrical Engineering and Computing Systems

of the College of Engineering and Applied Science

2014

By

Xuan Zhou

Ph.D., University of Cincinnati, OH, 2013

Committee Members:

Jaroslaw Meller, Ph.D., Chair

Raj Krishna Bhatnagar, Ph.D.,

Yizong Cheng, Ph.D.

Abstract

In this thesis, we investigated an efficient framework for clustering analysis of gene expression profiles by discretizing continuous genomic data and adopting the 1D-jury approach for fast clustering that was previously used for protein model quality assessment. We demonstrated, through an empirical analysis of multiple data sets from independent studies, that the loss of information due to discretization of genomic data is limited. Patterns observed using the original data can largely be recovered from discretized expression profiles, while enabling efficient identification of genomic signatures and clustering of expression profiles. We further studied the application of 1D-Jury approach in reducing the dimensionality of genomic data. We demonstrated that discretization and 1D-Jury score projection efficiently reduced the dimensionality of feature space. More importantly, the proposed discretization-projection heuristic enhanced the discovery of cluster structure and patterns in the data. Therefore, the proposed discretization-projection method can be a valuable tool for the analysis of gene expression data.

Acknowledgement

I would like to thank my advisor, Dr. Jaroslaw Meller, for his generous help and insightful guidance. Dr. Meller suggested the overall scope of this thesis, and he also provided tremendous technical help which made this study possible.

I also want to thank Dr. Raj Krishna Bhatnagar and Dr. Yizong Cheng who were kind enough to agree to serve on my thesis committee and provided many valuable suggestions.

I want to thank Ms Julie Muenchen, Ms Vicki Baker, and Mr Anthony Seta for their administrative help.

Lastly and most importantly, I want to thank my dear parents. They were always there for me and I cannot say my thanks enough for their support and love.

Contents

Abstract i

Acknowledgement iii

Contents iv

Tables ix

Figures ix

Figure Legends x

Abbreviations xi

1 Introduction .... - 1 -

1.1 Clustering .... - 2 -

1.1.1 Definitions and Notations .... - 2 -

1.1.2 Components of Clustering .... - 2 -

1.1.3 The role of domain knowledge in clustering analysis .... - 8 -

1.1.4 Complexity of clustering algorithms .... - 8 -

1.1.5 Applications of Clustering Analysis .... - 9 -

1.2 Genomics and Sequencing.. - 9 -

1.3 References .... - 12 -

2 Empirical Analysis of the Effect of Discretization on Clustering Genomic Data... - 14 -

Abstract... - 15 -

Data.. - 16 -

Code .... - 17 -

Methods... - 18 -

Normalization functions.. - 18 -

Discretization functions .... - 18 -

Distance matrixes... - 19 -

Clustering functions .... - 19 -

Evaluating the effect of discretization on clustering analysis...................... - 19 -

Results.. - 20 -

Complete linkage hierarchical clustering is appropriate to cluster genomic data .... - 20 -

Discretization of the Simulation data set preserved cluster structure .... - 21 -

Discretization of the Iris data set preserved cluster structure .... - 22 -

Discretization of genomic data sets preserved cluster structures............... - 22 -

Conclusions .... - 25 -

References .... - 26 -

Figures .... - 27 -

Figure 2-1 Evaluation of different hierarchical clustering algorithms .... - 27 -

Figure 2-2 Discretization of the Simulation data set preserved its cluster structure .... - 28 -

Figure 2-3 Discretization of the Iris data set preserved its cluster structure - 29 -

Figure 2-4 Discretization of the Collins_PAM50 data set preserved its cluster structures.. - 30 -

Figure 2-5 Discretization of other genomic data sets preserved their cluster structures.. - 31 -

Figure legends .... - 32 -

Figure Legend 2-1 Evaluation of different hierarchical clustering algorithms on genomic data sets.. - 32 -

Figure Legend 2-2 Discretization of the Simulation data set preserved its cluster structure .... - 32 -

Figure Legend 2-3 Discretization of the Iris data set preserved its cluster structure .... - 33 -

Figure Legend 2-4 Discretization of the Collins_PAM50 data set preserved its cluster structures........ - 33 -

Figure Legend 2-5 Discretization of other genomic data sets preserved their cluster structures........ - 34 -

3 Projecting High-dimensional Genomic Data into Lower Dimensions Using the 1D-Jury Score Function in Linear Time .... - 36 -

Abstract........ - 37 -

Data........ - 38 -

Code .... - 38 -

Methods........ - 39 -

Gene program (GP) .... - 39 -

1D-Jury function for genomic data sets........ - 41 -

Results........ - 44 -

Dimension reduction of Collins_SmallPValue data set by 1D-Jury score dampened cluster details but still preserved overall cluster structure .... - 44 -

Dimension reduction of TCGA_4_Cancers data set by 1D-Jury score enhanced cluster structure .... - 45 -

Conclusions .... - 47 -

References .... - 49 -

Figures... - 50 -

Figure 3-3 Dimension reduction of Collins_SmallPValue data set by 1D-Jury score dampened cluster details but still preserved overall cluster structure..... - 50 -

Figure 3-4 Dimension reduction of TCGA_4_Cancers data set by 1D-Jury score enhanced cluster structure .... - 51 -

Figure Legends .... - 52 -

Figure Legend 3-3 Dimension reduction of Collins_SmallPValue data set by 1D-Jury score dampened cluster details but still preserved overall cluster structure .... - 52 -

Figure Legend 3-4 Dimension reduction of TCGA_4_Cancers data set by 1D-Jury score enhanced cluster structure... - 52 -

4 Summaries and Perspectives .... - 54 -

Figure 4-1 Feature extraction on the discretized-projected TCGA_4_Cancers data set .... - 57 -

Figure Legend 4-1 Feature extraction on the discretized-projected TCGA_4_Cancers data set .... - 57 -

Tables

Table 1-1-1 Complexity of Clustering Algorithms... - 8 -

Table 3-1 Lists of gene programs defined pathways.. - 39 -

Figures

Figure 2-1 Evaluation of different hierarchical clustering algorithms......................... - 27 -

Figure 2-2 Discretization of the Simulation data set preserved its cluster structure . - 28 -

Figure 2-3 Discretization of the Iris data set preserved its cluster structure............. - 29 -

Figure 2-4 Discretization of the Collins_PAM50 data set preserved its cluster structures - 30 -

Figure 2-5 Discretization of other genomic data sets preserved their cluster structures . - 31 -

Figure 3-1 Schematic demonstration of an efficient 1D-Jury score algorithm .... - 42 -

Figure 3-2 Schematic illustration of 1D-Jury Score for an entire genomic data set.... - 43 -

Figure 3-3 Dimension reduction of Collins_SmallPValue data set by 1D-Jury score dampened cluster details but still preserved overall cluster structure .... - 50 -

Figure 3-4 Dimension reduction of TCGA_4_Cancers data set by 1D-Jury score enhanced cluster structure .... - 51 -

Figure 4-1 Feature extraction on the discretized-projected TCGA_4_Cancers data set- 57 -

Figure Legends

Figure Legend 2-1 Evaluation of different hierarchical clustering algorithms on genomic data sets .... - 32 -

Figure Legend 2-2 Discretization of the Simulation data set preserved its cluster structure .... - 32 -

Figure Legend 2-3 Discretization of the Iris data set preserved its cluster structure . - 33 -

Figure Legend 2-4 Discretization of the Collins_PAM50 data set preserved its cluster structures... - 33 -

Figure Legend 2-5 Discretization of other genomic data sets preserved their cluster structures.. - 34 -

Figure Legend 3-1 Schematic demonstration of an efficient 1D-Jury score algorithm. - 42 -

Figure Legend 3-2 Schematic illustration of 1D-Jury Score for an entire genomic data set .... - 43 -

Figure Legend 3-3 Dimension reduction of Collins_SmallPValue data set by 1D-Jury score dampened cluster details but still preserved overall cluster structure..... - 52 -

Figure Legend 3-4 Dimension reduction of TCGA_4_Cancers data set by 1D-Jury score enhanced cluster structure .... - 52 -

Figure Legend 4-1 Feature extraction on the discretized-projected TCGA_4_Cancers data set.. - 57 -

Abbreviations

PCA	principle component analysis
ICA	independent component analysis
MDS	multidimensional scaling
LCC	library of congress classification
SNP	single nucleotide polymorphisms

NGS	next generation sequencing
RPKM	reads per kilobase per million
FPKM	fragments per kilobase of exon per million fragments mapped
BRCA	breast carcinoma

1 Introduction

1.1 Clustering

Clustering is a common unsupervised machine learning task, which involves separating a finite set of objects (e.g., biological samples defined by gene expression vectors) into non-overlapping or overlapping groups based on some similarity or dissimilarity measures. The main difference between clustering and classification, another major type of machine learning technique, is that the class labels are no used during clustering. The goal of clustering is to put samples that are "similar" to each other into the same group while keeping "dissimilar" samples apart.

1.1.1 Definitions and Notations

- A pattern **X** is a vector representing a sample to be clustered: **X** = $(x^{(1)}, \ldots, x^{(d)})^T$.
- A feature $x^{(i)}$ is a scalar component of a pattern **X.** It can be numerical (continuous, discrete or interval) or categorical (nominal or ordinal).
- A pattern set $\mathscr{X}$ = $\{\mathbf{X_1}, \ldots, \mathbf{X_n}\}$ is a matrix representation of n samples to be clustered.
- A distance function D: $\mathbf{X_i}, \mathbf{X_j} \rightarrow \mathscr{R}$ maps two patterns from the feature space into a real number. The output of this distance function is inversely correlated with the similarity between sample i and sample j.

1.1.2 Components of Clustering

Numerous clustering algorithms have been developed. Generally speaking, a clustering algorithm consists of the following components: pattern representation, distance function definition, clustering method, clusters validation, results interpretation and knowledge generation [1, 2].

1.1.2.1 Pattern representation

A sample can be represented by limitless features and not all features are relevant to a particular clustering task, which makes it neither unfeasible nor undesirable to use all the features as input for clustering. A complication is that when the dimensionality of feature space is high enough, all patterns tend to be equally distant from others irrespective of the similarity measure used[3]. Therefore, it is necessary to only select the most informative features as inputs for further steps of clustering. This is usually achieved by feature selection or feature extraction. Feature selection is the process of selecting the most informative subset of original features for further clustering, whereas feature extraction also involves additional transformations of the input features into new features. Usually both feature selection and feature extraction are performed on the pattern set to acquire a proper feature set for clustering analysis. By selecting critical and independent features, feature selection and extraction enhance the overall clustering outcome. In addition, since feature selection and extraction lead to reduced dimensionality, these operations will also reduce the computational complexity of clustering. Meanwhile, the reduced dimensionality is also advantageous in generating models that are easier to interpret.

Typical feature selection/extraction strategies include principle component analysis (PCA), independent component analysis (ICA), multidimensional scaling (MDS). Both PCA and ICA are linear transformation: PCA is proper for normal distributions; ICA is more suitable for non-normal distribution of the data. MDS, on the other hand, is a non-linear projection function which maps a high dimension matrix into a low dimension structure while trying to maintain most information[2]. It is noteworthy that feature selection and extraction will inevitably result in loss of information, which sometimes distorts the clusters. Therefore, comprehensive validation and benchmarking are usually advised.

1.1.2.2 Distance function definition

Distance functions define how two patterns from the feature space are "dissimilar" from each other, which is the starting point of most clustering algorithms. There are many distance functions proposed and a careful selection of the distance function is one of the key components of clustering procedures.

Perhaps the best known distance function is the Euclidean distance $d(X_i, X_j) = \|X_i - X_j\|_2$. The Euclidean distance has an intuitive interpretation for distances between a pair of points in geometric space. The Euclidean distance is a special case (p = 2) of the Minkowski distance $d(X_i, X_j) = \|X_i - X_j\|_p$. Another widely used Minkowski distance is the Manhattan distance, which is a p = 1 Minkowski distance. A drawback of directly using Minkowski distance is that features on large scales tend to mask the ones on smaller scales and therefore proper scaling /normalization is usually required during the pattern representation step. Other common used distance metrics include the squared Mahalanobis distance $d(X_i, X_j) = (X_i - X_j)^T \Sigma^{-1}(X_i - X_j)$, which is not susceptible to difference in feature scales. Cosine similarity $S(X_i, X_j) = \frac{X_i^T X_j}{\|X_i\|\|X_j\|}$ measures the similarity between two patterns, which is inversely correlated with the distance between the patterns.

1.1.2.3 Clustering methods

A large number of clustering methods have been developed and a comprehensive review of these methods is out of the scope of this thesis. Generally speaking, there are two types of clustering algorithms: hierarchical clustering and partition-based clustering methods.

The output of a hierarchical clustering is a dendrogram, the leaves of which are clusters with individual samples per cluster and the root of which is a cluster consisting of all the samples altogether. An internal node of the dendrogram joins clusters most similar to each other and height of the node corresponds to the distances between its child clusters. Therefore, to partition of the pattern set $\mathscr{X}$ into k disjoint clusters, one can cut the dendrogram at different heights until k clusters are separated. Depending on how to define distances between clusters, hierarchical clustering has several versions. For example, the distance in single-linkage is defined as the minimum of all pairs of patterns from the two clusters; the distance in complete-linkage is defined as the maximum of all pairs of patterns from the two clusters. Other commonly used clustering methods include Ward's minimum variance method, centroid linkage clustering, and average linkage cluster[4].

Instead of generating a dendrogram containing all patterns in the pattern set, partition-based clustering algorithms generate discrete clusters, overlapping (fuzzy clustering) or not (hard clustering). Computationally speaking, partition-based clustering algorithms are usually more efficient compared with hierarchical clustering, and thus they might be more suitable for clustering large data sets. One major downside of partition-based clustering algorithms is that they are usually very sensitive to the number of output clusters, and multiple runs are usually required to generate an optimal result.

K-means clustering is perhaps the most representative partition-based clustering algorithms. It attempts to minimize squared error of the clustering ($e^2(x,c) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \left\| X_i^{(j)} - c_j \right\|$, where $X_i^{(j)}$ is the i^{th} pattern of cluster j^{th}, and c_j is the centroid of cluster j^{th}). It starts by randomly choosing k (the number of output clusters) centers in the feature space. Then assign each pattern in the pattern set to the nearest center. Re-compute the k centers based on the updated clusters. Repeat the assign-labels/compute-centers procedure until convergence, usually the number of iterations has reached the pre-defined limit or no change in label assignment. Since the initial centers are chosen randomly, it is usually necessary to run the algorithms multiple times and evaluate the results to select the optimal. Variances of k-means algorithm include k-medoids algorithm, dynamic clustering algorithm, and the ISODATA algorithm [1].

1.1.2.4 *Clusters validation*

As mentioned, one hallmark of clustering analysis is that the cluster labels of samples are not used during the clustering procedure. However, in many instances the true labels of samples can be generated by other means (usually by human experts) and are valuable during the validation step. Based on whether the label information is used, clustering validation can be further subcategorized into internal and external validation.

Internal validation doesn't rely on the labeling information, rather, solely bases on the data used to perform the clustering. Internal validation assigns high scores to clustering algorithms with high intra-cluster similarity and low inter-cluster similarity. Commonly used internal validations includes Davies-Bouldin index, Dunn index[5].

External validation, on the other hand, harnesses information that is not used for the clustering analysis, such as label information. Label information is usually referred to as ground truth, gold standard, or benchmark. External validation evaluates how the clustering output resembles the cluster labels given by ground truth. However, as well be discussed later, clustering is subjective and there might be multiple valid ground truths for the same input data. Therefore, one clustering algorithm that generates an inferior external validation result is not necessarily worse than another one that generates a better external validation result, according to the similarity to one particular set of ground truth. Another point noteworthy is that usually the purpose of clustering analysis is to discover novel knowledge but validation according to known ground truth may not be ideal for this aim [6]. Commonly used external evaluations of clustering algorithms include Rand index and its variation adjusted Rand index, F-measure, Jaccard index, mutual information and confusion matrix.

1.1.2.5Results interpretation and knowledge generation

Usually the goal of clustering analysis is to provide users a meaningful view of the original data. Therefore, it is usually necessary to abstract the original data, generating a simple, human or machine interpretable representation. One common way of representing a clustering result is to provide a compact abstraction for each cluster, usually a cluster prototype such as centroid.

1.1.3 The role of domain knowledge in clustering analysis

Assessing the results of clustering analysis is challenging and typically requires some domain knowledge. Implicitly, domain knowledge is used to define pattern representation of samples, select a proper distance function and choose an appropriate clustering method. Domain knowledge is also frequently used to generate cluster label information, which is valuable during clusters validation step, as mentioned in section **Clusters validation**.

1.1.4 Complexity of clustering algorithms

One major challenge of clustering analysis is that most algorithms are relatively computational expensive. Generally speaking, to abstain a optimal partition of n samples into k clusters is a NP-hard task [7] and all the aforementioned clustering algorithms are heuristic. Table 1.1 summarizes the time and space complexity of some commonly used clustering algorithms.

Table 1-1-1 Complexity of Clustering Algorithms

Clustering Algorithm	Time Complexity	Space Complexity	Notation
Single-linkage Hierarchical Clustering[8]	O(N^2)	O(N^2)	N: number of samples
Complete-linkage Hierarchical Clustering[9]	O(N^2)	O(N^2)	N: number of samples

k-means[10]	O(Nki)	O(N + k)	N: number of samples;
			k : number of clusters;

			i: number of iterations

k-medoids[11]	$O(N(N- k)^2i)$	O(N + k)	N: number of samples;
			k : number of clusters;
			i: number of iterations

1.1.5 Applications of Clustering Analysis

Never before have we generated such a huge volume of information, thanks to the advance in technologies such as the Internet, genomics, sensor network, etc. The growing of information demands the evolution in the means to analyze and interpret it. Without proper analysis, information is useless. Clustering analysis is one of the most fundamental data analysis techniques and has been broadly used as an essential step during data analysis. Examples of successful applications of clustering analysis include information retrieval [12] [13], clustering documents based on co-occurrence of terms to analysis the structure of large set of documents [14], and image segmentation, which is briefly discussed below.

Image segmentation is the process partitioning of an image into regions, each of which is generally considered homogenous with respect to certain properties, such as color or intensity. Image segmentation is fundamental to many computer vision applications and the idea that separating images into homogenous regions makes clustering analysis a valid candidate for solving image segmentation tasks. Indeed, over the years, numerous clustering algorithms have been proposed to tackle the image segmentation problems [15, 16].

1.2 Genomics and Sequencing

Genomics is the discipline that utilizes recombinant DNA, nucleic acid sequencing and bioinformatics techniques to study the structure and function of genomes[1]. As a result of advances in technologies, especially sequencing techniques, genomic studies have become one of the most exciting fields in biomedical researches. In addition to sequencing, microarray technology has been used to quantitatively determine the abundance of gene products. Another usage of microarray technology is to examine the presentence of certain single nucleotide polymorphisms (SNPs) in a sample. The major disadvantages of microarray technology includes: microarray relies on known sequence, therefore it can only detect known SNPs or expression levels of known transcripts, whereas it is usually interesting to find novel SNPs or transcripts; microarray is suffered from relatively high background noises; the dynamic range of microarray is relatively low; the sensitivity and specificity of microarray technology are limited. As a result of the aforementioned disadvantages, although microarray technology is still heavily used in biomedical studies, it is gradually replaced by another technology, i.e., next generation sequencing (NGS)[18]. NGS has the ability of detecting novel gene products, with low background, high dynamic range, sensitivity and specificity [19]. The reduced cost of NGS has greatly promoted its popularity in basic researches and diagnostic applications. Error rates of NGS still need to be reduced and the read lengths of most NGS platforms are relatively short. However, besides the technical limitations, the major hurdle slowing the progress of high throughput genomics (including both microarray and NGS) are bottlenecks in computational infrastructures to store and

[1] http://en.wikipedia.org/wiki/Genomics

analyze the genomics data. Therefore, development of efficient algorithms for genomics data is of great interest.

Typically, the raw results of microarray experiments are images and the intensities at different spots correspond to the expression levels of the genes at the spots. The images are digitalized and adjusted for further analysis. The raw results of NGS experiments are short reads of nucleic acid sequences, and the counts of short reads correlate with the levels of transcripts (which also depend on other parameters including the total number of reads sequenced, the length of the transcripts, etc.). Since the counts of short reads are not solely determined by the expression levels of their corresponding transcripts, it is necessary to normalize the counts to measures such as reads per kilobase per million (RPKM) or fragments per kilobase of exon per million fragments mapped (FPKM). The expression level of a transcript from both microarray and NGS experiments can be represent by a real number. In this thesis, I will represent the outputs for microarray and NGS experiments as *e x G* matrixes, with each row represents the expression of a transcript across all *e* samples and each column represents the expressions of all *G* annotated transcripts within a sample. Typically, the length of a column *G* is relatively stable and for human (Homo sapiens) *G* is usually around 20 000, the number of identified genes in human. However, the dimensionality of a row *e* in the matrix can vary widely. With the advance in NGS technology and reduction in cost, the dimensionality of rows is expected to increase significantly. Therefore, an infrastructure that efficiently stores and analyzes very large scale microarray and NGS data is critical for progress in biomedical researches.

1.3 References

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *Acm Computing Surveys,* vol. 31, pp. 264-323, Sep 1999.

[2] R. Xu and D. Wunsch, "Survey of clustering algorithms," *Ieee Transactions on Neural Networks,* vol. 16, pp. 645-678, May 2005.

[3] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," *Database Theory - Icdt'99,* vol. 1540, pp. 217-235, 1999.

[4] S. Watanabe, "Survey of Clustering Algorithms," *Ieee Transactions on Systems Man and Cybernetics,* vol. Smc1, pp. 398-&, 1971.

[5] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems,* vol. 17, pp. 107-145, 2001.

[6] D. Pfitzner, R. Leibbrandt, and D. Powers, "Characterization and evaluation of similarity measures for pairs of clusterings," *Knowledge and Information Systems,* vol. 19, pp. 361-394, Jun 2009.

[7] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Machine Learning,* vol. 75, pp. 245-248, May 2009.

[8] R. Sibson, " SLINK: an optimally efficient algorithm for the single-link cluster method," *The Computer Journal (British Computer Society),* vol. 16, pp. 30–34, 1973.

[9] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal (British Computer Society),* vol. 20, pp. 364–366, 1977.

[10] S. P. Lloyd, "Least-Squares Quantization in Pcm," *Ieee Transactions on Information Theory,* vol. 28, pp. 129-137, 1982.

[11] H. S. Park and C. H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications,* vol. 36, pp. 3336-3341, Mar 2009.

[12] L. o. Congress, *LC classification outline*: Library of Congress, Washington, DC, 1990.

[13] P. Willett, "Document Clustering Using an Inverted File Approach," *Journal of Information Science,* vol. 2, pp. 223-231, 1980.

[14] H. Small, E. Sweeney, and E. Greenlee, "Clustering the Science Citation Index Using Co-Citations .2. Mapping Science," *Scientometrics,* vol. 8, pp. 321-340, 1985.

[15] G. B. Coleman and H. C. Andrews, "Image Segmentation by Clustering," *Proceedings of the Ieee,* vol. 67, pp. 773-785, 1979.

[16] S. Sinha and S. Deb, "Image segmentation by intelligent clustering technique," *2013 Ieee Recent Advances in Intelligent Computational Systems (Raics),* pp. 272-276, 2013.

[17] F. Sanger and A. R. Coulson, "A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase," *J Mol Biol,* vol. 94, pp. 441-8, May 25 1975.

[18] P. J. Hurd and C. J. Nelson, "Advantages of next-generation sequencing versus the microarray in epigenetic research," *Brief Funct Genomic Proteomic,* vol. 8, pp. 174-83, May 2009.

[19] Y. Chu and D. R. Corey, "RNA sequencing: platform selection, experimental design, and data interpretation," *Nucleic Acid Ther,* vol. 22, pp. 271-4, Aug 2012.

2 Empirical Analysis of the Effect of Discretization on Clustering Genomic Data

Abstract

Genomic data from microarray and NGS experiments are usually represented as matrixes of real numbers and the dimensionality of these matrixes can be very large. Therefore, it is necessary to develop efficient algorithms to analysis these data. Clustering analysis is one of the most important approaches to interpret genomic data but they are usually computationally expensive. Discretization is the process transforming continuous data into discrete states. Discretization might be beneficial for the clustering task. By transforming continuous data into discrete states, we can employ more efficient distance methods such as Hamming distance. More importantly, discretization is also an essential pre-processing step for some feature extraction methods such as 1D-Jury, which is a linear algorithm that projects high dimensional data into much lower dimensions. Another reason to perform discretization is to smooth the noise within the input data and this might be helpful for the outcome of clustering analysis. In this chapter, we demonstrated that performing discretization on genomic data still preserved enough information for further clustering; and that in some instances the discretized data out-performed the original data. We also demonstrated that Hamming distance worked as good as Euclidean distance on discretized genomic data set, suggesting that ordinal information of the discrete state was not essential for proper clustering of genomic data.

Data

Data used in this chapter all have the cluster labels (hereafter referred to as True Label), but the label information is only used during the validation step.

- Simulation data set. The Simulation data set is a simulated 100 x 200 matrix. A 100 x 5 mean matrix was first generated and each row of this matrix was a random permutation of $\{1,2,3,4,5\}$ and all the rows were independent. For the ith column of the Simulation data set, its cluster label C$^{(i)}$ was chosen randomly from the set $\{1,2,3,4,5\}$ and its value was generated by adding a 0 mean, *sd* standard deviation Gaussian noise to the C$^{(i)}$ column of the mean matrix. In this study, sd-s of 3, 4 and 5 were used to generate data to represent data with different levels of cluster structures.
- The iris data set [1]. The Iris data set is one of the most popular data sets in the machine learning field. It has been widely used as benchmark data for both classification and clustering problems. It has measurements on 50 samples from each of three iris species (iris setosa, iris versicolor and iris virginical). Four features were measured: sepal length, sepal width, petal length and petal length. What makes this data set interesting is that while iris setosa can be easily separated from the other two species by most clustering algorithms, it is harder to separate iris versicolor from iris virginical.
- Collins data set[2] . This data set contained mRNA-sequencing results from four types of invasive breast carcinoma (BRCA) (basal-like, Her2-positive, luminal A, and luminal B), as well as normal and not identified (NA) tissues. The number of normal tissues was very small (n = 6) and the NA group (n = 485) was not a coherent group.

Therefore, we excluded them from our study and only used data from BRCA samples (n = 428). We selected two subsets of genes from this data set for our analysis. PAM50 is a 50-gene list with discriminatory power in separating subtypes of BRCA samples [3] and we took the expression of this 50 genes and referred to them as Collins_PAM50. There was also metadata about the p53 mutation status (wildtype vs. mutated) associated with the data set and we selected genes that were differentially expressed between these two categories (fold change >= 2 and p-value <= 0.001) and referred to this subset of data as Collins_p53.

- Miller data set [4] contained microarray results from 251 BRCA samples. This data set was associated with estrogen receptor(ER) (ER positive and negative), and p53 (p53 X0 and X1) status. We used these information to select genes that were differentially expressed between samples of different categories (fold change >= 2 and p-value <= 0.001). By this way, we got two subsets of the original data set and referred to them as Miller_ER and Miller_P53.
- Wang data set[5] contained microarray data from 148 prostate cancer samples with proper meta data about the percentage of stroma. The stroma percentage was discretized into 4 equal size categories (quart 1 to 4) and we used this information to select genes that were differentially expressed between categories (fold change >= 2 and p-value <= 0.001). This data set is referred to as Wang_Stroma.

Code

R scripts used in this chapter can be found: https://github.com/reformasky/xuanThesis.

Methods

Normalization functions

The expression levels of different genes in a genomic data set might be on different scales and therefore it is necessary to adjust them to a notionally common scale. Two normalization methods were used throughout this chapter. Feature scaling was defined as $\boldsymbol{X}^{normalized} = \frac{\boldsymbol{X}-\min(\boldsymbol{X})}{\max(\boldsymbol{X})-\min(\boldsymbol{X})}$ and Z Score was defined as $\boldsymbol{X}^{normalized} = \frac{\boldsymbol{X}-u}{\sigma}$ $(u{:}\, population\ mean; \sigma{:}\, standard\ deviation\ ).$

Discretization functions

A discretization function $D{:}\,\boldsymbol{X}, numOfStates \rightarrow \boldsymbol{X}^{\boldsymbol{discretized}}$ maps a vector of continuous values into a vector of discrete states {0, 1 …, numOfStates -1}. Three discretization functions were used in this chapter.

Equal frequency discretization was defined as $\boldsymbol{X}^{\boldsymbol{discretized}} = floor\left(\frac{rank(\boldsymbol{X}) * numOfStates}{size(\boldsymbol{X})}\right)$, where *rank(**X**)* returned the sample ranks of the values in ***X*** and *size(**X**)* returned the number of samples in ***X*** (For *rank(**X**)* == size(**X**), we specially defined $\boldsymbol{X}^{\boldsymbol{discretized}} = numOfStates - 1$).

Equal interval discretization was defined as $\boldsymbol{X}^{\boldsymbol{discretized}} = floor\big(\boldsymbol{X}^{normalized} * numOfStates$ where $\boldsymbol{X}normalized$ was normalized by feature scaling function (For $\boldsymbol{X}normalized{=}{=}1$, we specially defined $\boldsymbol{X}discretized = numOfStates - 1$).

Z Score discretization was defined as $\boldsymbol{X}^{\boldsymbol{discretized}} = floor(\ pnorm(\boldsymbol{X}^{normalized}) * numOfStates)$ where $\boldsymbol{X}^{normalized}$ was normalized by Z Score normalization and $pnorm(\boldsymbol{X}^{normalized})$ was the distribution function of standard Gaussian distribution.

Distance matrixes

Distance matrix D is a matrix representation of distances between pairs of patterns in a data set, with the element *D (i, j)* corresponds to the distance between the *i*th and *j* th patterns. Distance matrixes using Euclidean distance $d(\boldsymbol{X}_i, \boldsymbol{X}_j) = \left\|\boldsymbol{X}_i - \boldsymbol{X}_j\right\|_2$ and Hamming distance $d(\boldsymbol{X}_i, \boldsymbol{X}_j) = \left\|\boldsymbol{X}_i - \boldsymbol{X}_j\right\|_0$ were used throughout in this chapter.

Clustering functions

Unless otherwise specified, all clustering analysis was performed using complete-linkage hierarchical clustering method implemented in the stats package of R. We also evaluated other hierarchical clustering algorithms such as single-linkage, centroid, average, median and Ward's minimum variance method, all of which were also standard implementation in the stats package of R.

Evaluating the effect of discretization on clustering analysis

We evaluated the effect of discretization on clustering analysis using adjusted Rand index ($adjRand(G, G') = \frac{Rand(G,G') - E(Rand)}{1 - E(Rand)}$ and $Rand(G, G') = \frac{a+d}{\binom{n}{2}}$, where a is the number of pairs of elements that are in the same cluster in *G* and are in the same cluster *G'*; *d* is the number of pairs of elements that are in different clusters in both *G* and *G'*; n is the number of samples in the pattern set. *E (Rand)*: expected Rand index for two random clustering of the same data set into k clusters), which is implemented in the phyclust package of R. We used the cluster label information associated with the data sets as ground truth (True Label). In addition, in order to assess the similarity between clustering using discretized and non-discretized data, we also computed adjusted Rand index between clustering results using discretized data and non-discretized data.

Results

Complete linkage hierarchical clustering is appropriate to cluster genomic data

Compared with partitional clustering algorithms such as k-means, hierarchical clustering algorithms not only determine the cluster labels of samples, but also provide the structural information about clusters, which might be useful for further investigations. In addition, hierarchical clustering algorithms do not depend on the initial seeds for clusters, which is also beneficial for our exploratory analysis. Therefore, throughout this study, we will use hierarchical clustering as a demo algorithm.

Depending on how to define the distances between clusters, hierarchical clustering has several algorithms. Some of the most popular methods include single linkage, complete linkage, average linkage, median linkage, centroid linkage and Ward's minimum variance method (Ward). We clustered genomic data sets using the aforementioned algorithms and compared our clustering results against the True Label information came with the data sets. Since the expression levels of different genes were different, we also performed feature scaling normalization per gene before clustering. As demonstrated by figure 2-1, complete linkage and Ward methods performed similarly well on feature scaling normalized data sets and were superior to other linkage methods. Similar observation also held for Z Score function normalized data (Data not shown). Since the complexity of complete linkage algorithm is lower ($O(n^2)$) than Ward method ($O(n^3)$), we used complete-linkage hierarchical clustering throughout this chapter.

Discretization of the Simulation data set preserved cluster structure

In order to examine whether discretization would lead to excessive information loss, thereby disrupted the cluster structure within data, we first discretized the Simulation data set using equal frequency, equal interval and Z Score discretization functions and clustered the discretized data. We used both Euclidean distance and Hamming distance as distance metrics. For ground truth, we used both cluster labels associated with the data set (True Label) as well as clustering results using normalized but non-discretized data (non-discretized Label). As shown in figure 2-2, with relatively low noise ($\mathrm{sd} = 3$), we could achieve nearly perfect clustering using the non-discretized data. We could still similar results when clustering the discretized data using Euclidean distance, but not the Hamming distances. We observed the most consistent clustering results when using Z Score discretization (Figure 2-2 top). As noise increased (sd = 4 and 5), it became harder to cluster the data set. We barely found any similarity between the clustering results using discretized and non-discretized data, and neither of them was very close to the True Label. Even under the most difficult senior (sd = 5), however, the discretized data and non-discretized data still displayed similar level of resemblance towards the True Label, suggesting that discretization did not lead to excessive information lost.

Discretization of the Iris data set preserved cluster structure

We then tested whether the same observation still held on other data sets. We examined the effect of discretization on clustering the Iris data set. After discretization, we could still cluster the Iris data set efficiently (Figure 2-3). Interestingly, we observed that clustering on discretized data occasionally outperformed that clustering on the non-discretized data. For example, clustering the discretized data set by equal interval and Z Score discretization functions using Hamming metric was much better than the use of non-discretized data, which suggested that discretization might smooth the noise and therefore enhanced the clustering performance.

Discretization of genomic data sets preserved cluster structures

While it was attempting to observe that discretization of the Simulation and Iris data sets still preserved the cluster structures, it would be more interesting to test the effect of discretizations on genomic data. We first performed equal frequency, equal interval and Z Score discretizations on the Collins_PAM50 data set, which contained mRNA sequencing results of PAM50 genes from 4 types, totally 428 BRCA samples. We then clustered the discretized data using complete linkage hierarchy clustering method, with either Euclidean or Hamming distance as distance measures. As comparisons, we also performed hierarchical clustering on normalized but not discretized (by feature scaling or Z Score normalization) data. Since Hamming distance on the non-discretized data was meaningless, we only used

Euclidean distance to cluster the non-discretized data. Consistent with previous report [3], we also observed a distinct cluster of basal like BRCA samples after clustering the non-discretized data (Figure 2-4 **A** and **F**). In addition, Her2 positive samples were also mostly clustered separately when using the non-discretized data. Small clusters of luminal B samples were observed within larger clusters of luminal A samples (Figure 2-4 **A** and **F**). Discretization still preserved the overall cluster structures. Basal like BRCA samples were still clustered apart from other samples, and we could still observe separate Her2 positive samples (except for clustering using Hamming distance on equal interval discretized data (Figure 2-4 **C**)). We also observed slightly enhanced clustering outcome. For example, in Figure 2-4 **D**, discretization with equal frequency method and clustering using Euclidean distance efficiently separated luminal A and luminal B BRCA samples apart, which was not the case in the non-discretized data set. Overall speaking, clustering using equal interval and Z Score discretized data still retained appreciable similarity with the non-discretized data set; whereas clustering using equal frequency function discretized data displayed reduced similarity with the non-discretized data, especially when using Hamming distance as distance measure (Figure 2-4 **I,** top). We also evaluated the similarity between the clustering results using discretized data and True Label of the samples, and clustering using discretized data generated results with comparable resemblance with clustering using non-discretized data (Figure 2-4 **I**, bottom).

Similarly, we also examined the effect of discretization on clustering analysis using another 4 subsets of genomic data sets from 3 independent studies. Again, we observed that in general, when compared with the True Labels associated with the data sets, clustering on discretized data generated comparable results with the clustering using non-discretized data (Figure 2-5 **A –D**). We summarized the overall information loss/gain by discretization in figure 2-5 **E,** where we evaluated the ratio of adjusted Rand index between adjRand(discretized, True Label) and adjRand(non-discretized, True Label) using results from the Collins_PAM50, Collins_p53, Miller_ER, Miller_p53DLDA and Wang_Stroma data sets. We concluded that clustering using discretized data by 3 discretization methods generated comparable results with their non-discretized counterparts, when using either Euclidean or Hamming distance as distance measures.

One interesting observation was that Hamming distance performed at least as well as Euclidean distance in clustering discretized genomic data. This demonstrated that at least for the genomic data sets tested here, the ordinal information was not critical for proper clustering. This observation laid foundations for our studies in Chapter 3, where we used 1D-Jury scores to project high dimensional data into lower dimensions, a method relied on nominal instead of ordinal categorical data. Another interesting observation was that the resemblance between clustering using discretized data and True Labels did not directly correlate with the resemblance between clustering using discretized data and non-discretized data. In other words, the similarity between clustering discretized data and non-discretized data was not a good index for evaluating the clustering results using discretized data. For example, in figure 2-5 **B**, clustering the discretized Miller_ER data set by Z Score function using Euclidean distance and Hamming distance generated groups equally similar to that using the non-discretized data (top), but the clustering using Hamming distance was obviously much more resemble the True Label than that using Euclidean distance (bottom).

Conclusions

In this chapter, we empirically analyzed the effect of discretizations on 7 data sets, including the Simulation, Iris, Collins_PAM50, Collins_p53, Miller_ER, Miller_p53DLDA and Wang_Stroma data sets. We demonstrated that discretization did not disrupt the clustering structure, especially when clustering using Euclidean distance. A unique feature of the genomic data set was that ordinal information of the discretized states was not necessary for further clustering, which was evident by the fact that Hamming distance and Euclidean distances performed equally well on the discretized genomic data.

While discretization methods examined in this chapter appeared to be acceptable transformations of continuous genomic data into discrete states, proper clustering of the data sets depended on reasonable feature selections to represent the samples. In this chapter, we used two methods to select features: for the Collins_PAM50 data set, we used a predefined short list of genes from literature —PAM50; for other genomic data sets, we selected genes that were differentially expressed in different categories of samples. Both approaches relied on the prior knowledge about the labels of samples, and therefore, in theory, our clustering practice here was not genuinely unsupervised. In chapter 3, we will proposed an indexing algorithm that was capable of projecting high-dimensional data into much lower dimensions without the involvement of label information during clustering.

References

[1] R. Fisher, "The use of multiple measurements in taxonomic problems.," *Annals of Eugenics,* vol. 7, pp. 179-188, 1936.

[2] F. S. Collins and A. D. Barker, "Mapping the cancer genome. Pinpointing the genes involved in cancer will help chart a new course across the complex landscape of human malignancies," *Sci Am,* vol. 296, pp. 50-7, Mar 2007.

[3] J. S. Parker, M. Mullins, M. C. Cheang, S. Leung, D. Voduc, T. Vickery*, et al.*, "Supervised risk predictor of breast cancer based on intrinsic subtypes," *J Clin Oncol,* vol. 27, pp. 1160-7, Mar 10 2009.

[4] L. D. Miller, J. Smeds, J. George, V. B. Vega, L. Vergara, A. Ploner*, et al.*, "An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival," *Proc Natl Acad Sci U S A,* vol. 102, pp. 13550-5, Sep 20 2005.

[5] Y. Wang, X. Q. Xia, Z. Jia, A. Sawyers, H. Yao, J. Wang-Rodriquez*, et al.*, "In silico estimates of tissue components in surgical samples based on expression profiling data," *Cancer Res,* vol. 70, pp. 6448-55, Aug 15 2010.

Figures

Figure 2-1 Evaluation of different hierarchical clustering algorithms

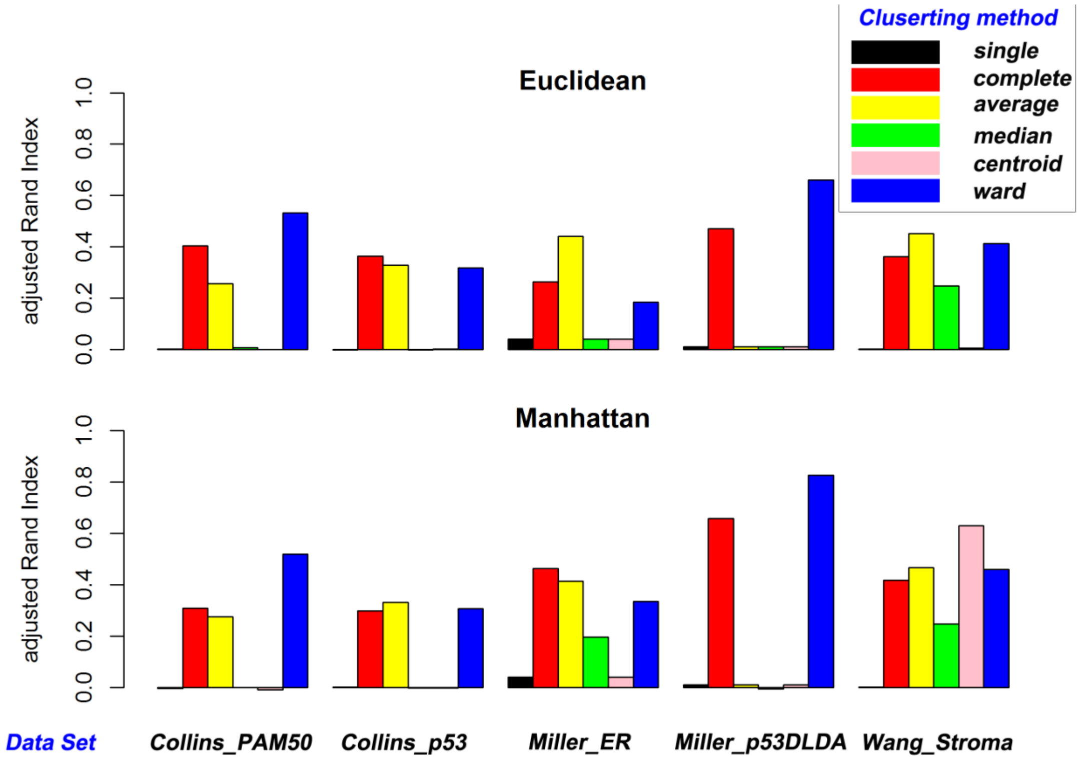


Figure 2-2 Discretization of the Simulation data set preserved its cluster structure

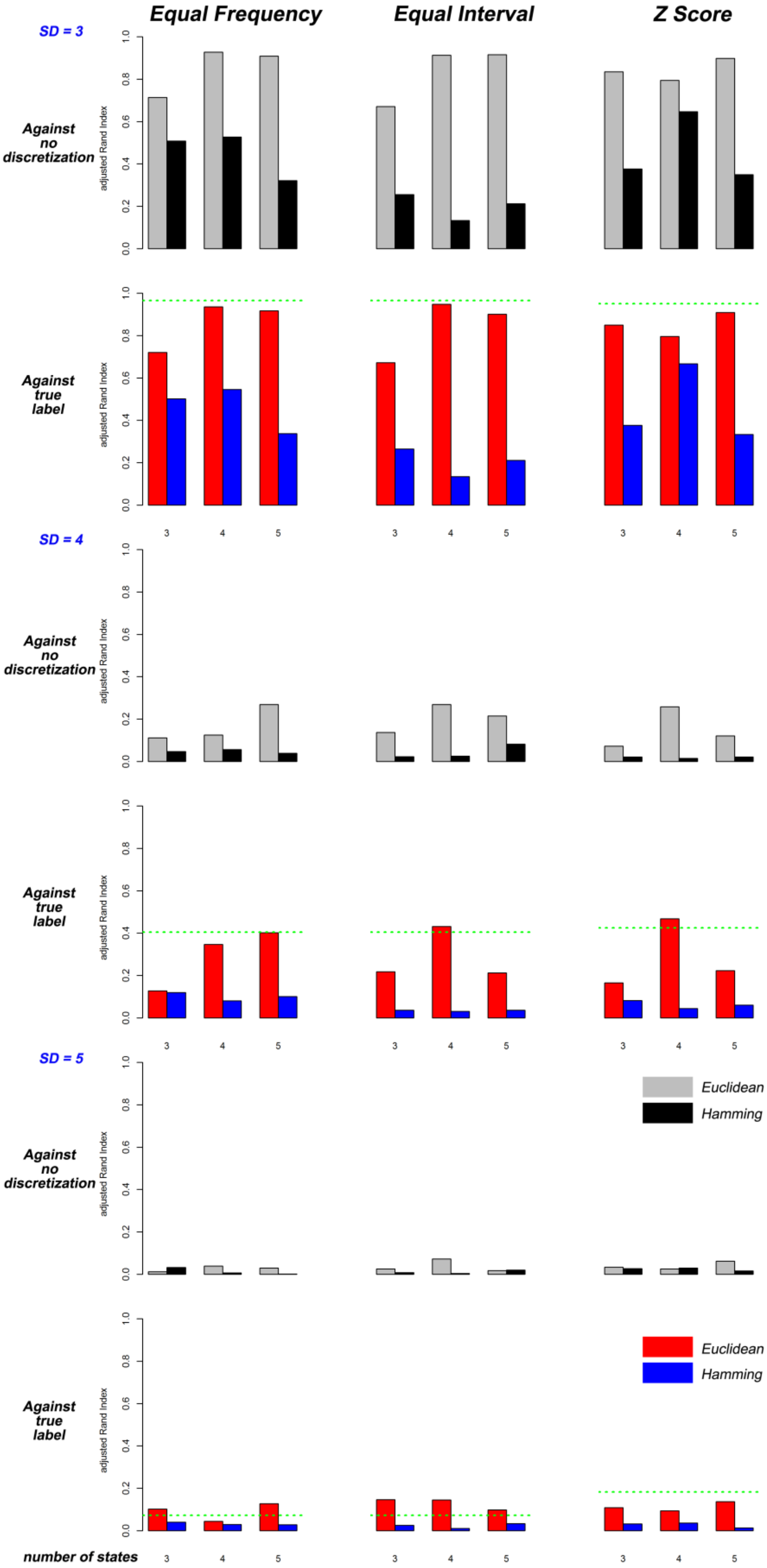


Figure 2-3 Discretization of the Iris data set preserved its cluster structure

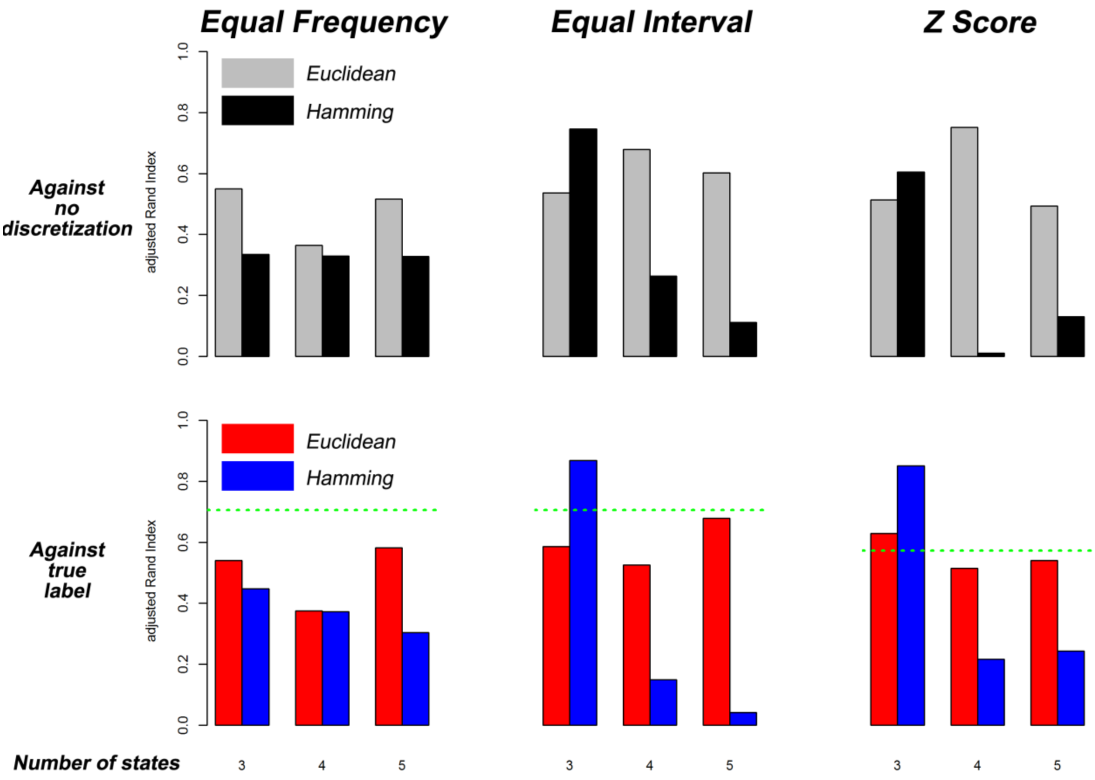


Figure 2-4 Discretization of the Collins_PAM50 data set preserved its cluster structures

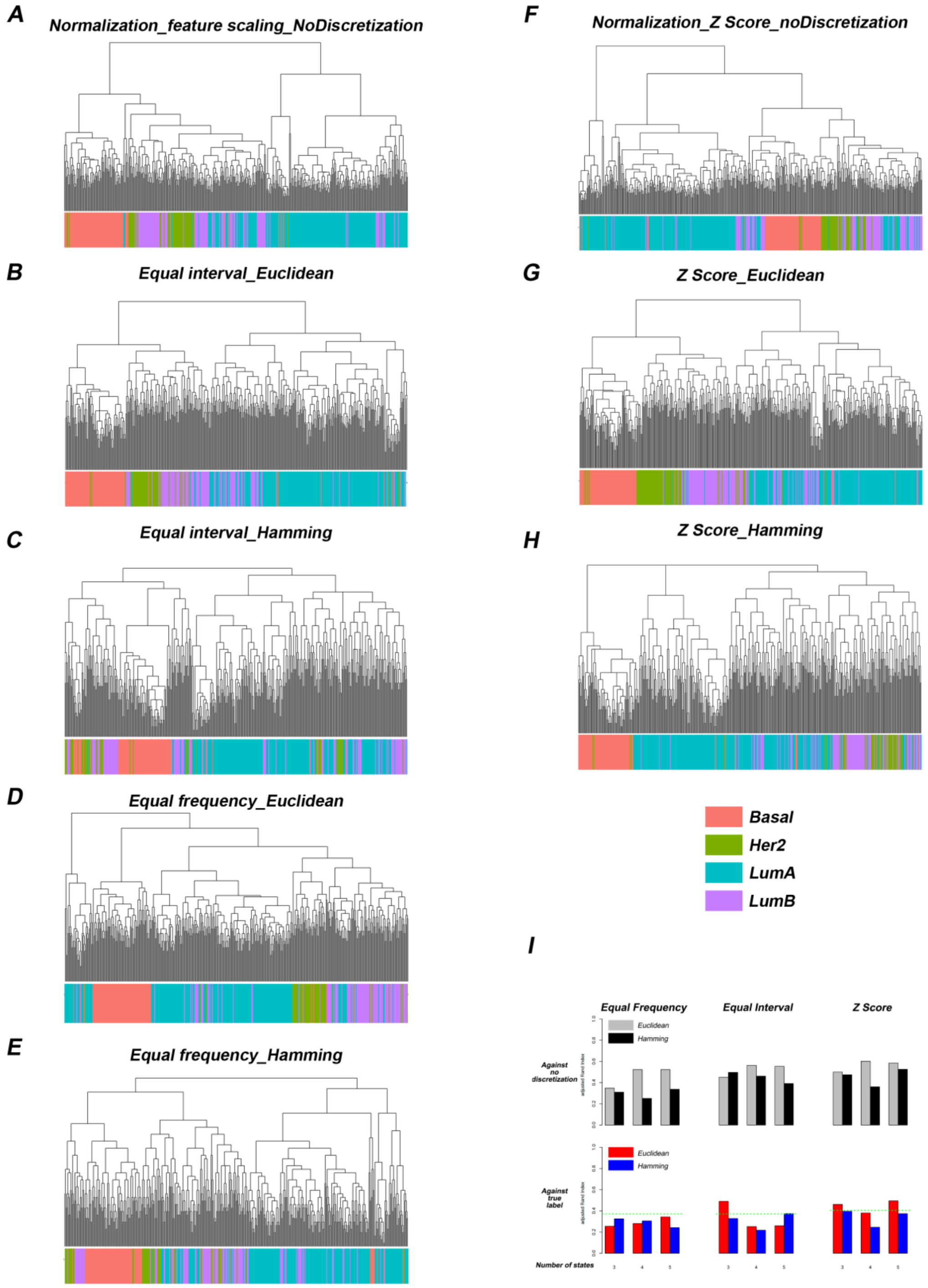


Figure 2-5 Discretization of other genomic data sets preserved their cluster structures

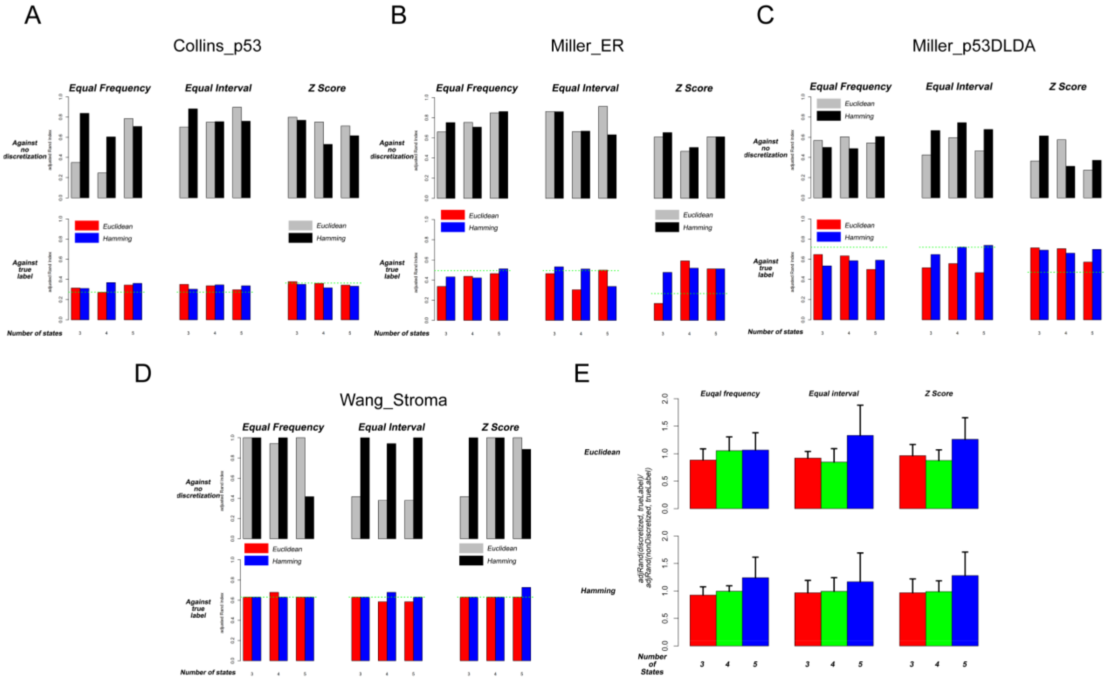


Figure legends

Figure Legend 2-1 Evaluation of different hierarchical clustering algorithms on genomic data sets

Genomic data sets Collins_PAM50, Collins_p53, Miller_ER, Miller_p53DLDA, and Wang_Stroma were first normalized using feature scaling method and then clustered using different hierarchical clustering algorithms as indicated. Adjusted Rand indexes between clustering results and cluster labels that were associated with data sets were shown. Clustering methods: single - single linkage; complete - complete linkage; average - average linkage; median - median linkage; centroid - centroid linkage; ward - Ward's minimum variance method. Euclidean (top) and Manhattan (bottom) distances were used as distance functions.

Figure Legend 2-2 Discretization of the Simulation data set preserved its cluster structure

Equal frequency, equal interval and Z Score functions were used to discretized the Simulation data sets with different standard deviations (sd = 3, 4,and 5) into 3, 4 and 5 discrete states. Hierarchical clustering was performed to cluster the discretized data into 5 groups (The number of centers in the Simulation data set) and the results of clustering were evaluated by adjusted Rand index against non-discretized Label (top, gray and black) and True Label (bottom, red and blue). Euclidean and Hamming metrics were used to define the distance between a pair of samples. Adjusted Rand index between True Labels and non-discretized (normalized by feature scaling for equal frequency/equal interval discretization and Z Score normalization for Z Score discretization) Labels were shown as reference lines in green in the bottom of each panel.

Figure Legend 2-3 Discretization of the Iris data set preserved its cluster structure

Equal frequency, equal interval and Z Score functions were used to discretized the Iris data set in to discrete states. Discretized data were clustered into 3 clusters by hierarchical clustering with Euclidean and Hamming metrics. The clustering results using discretized data were evaluated by adjusted Rand index against non-discretized Label (top, gray and black) and True Label (bottom, red and blue). Adjusted Rand indexes between True Label and non-discretized (normalized by feature scaling for equal frequency/equal interval discretization and Z Score normalization for Z Score discretization) Label were shown as reference lines in green in the bottom plot.

Figure Legend 2-4 Discretization of the Collins_PAM50 data set preserved its cluster structures

(**A** - **H**) Dendrograms (top) and distribution of True Label (bottom) after clustering non-discretized (**A** and **F**) and discretized (**B**, **C**, **D**, **E**, **G** and **H,** number of states = 3) Collins_PAM50 data set. For non-discretized data sets, only Euclidean distance were used for clustering analysis and for the discretized data sets, both Euclidean distance and Hamming distance were used. Color codes for BRCA sample types were shown.

(**I**) The Collins_PAM50 data set was discretized by equal frequency, equal interval and Z Score discretization functions. Discretized data were clustered into 4 clusters by hierarchical clustering with Euclidean and Hamming distances as distance measures. The clustering results on discretized data were evaluated by adjusted Rand index against non-discretized Label (top, gray and black) and True Label (bottom, red and blue). Adjusted Rand indexes between True Labels and non-discretized (normalized by feature scaling for equal frequency/equal interval discretization and Z Score normalization for Z Score discretization) Labels were shown as reference lines in green in the bottom plots.

Figure Legend 2-5 Discretization of other genomic data sets preserved their cluster structures

(**A**- **D**) Four genomic data sets from three studies, Collins_p53 (**A**), Miller_ER (**B**), Miller_p53DLDA (**C**) and Wang_Stroma (**D**) were discretized by equal frequency, equal interval and Z Score functions. Discretized data were clustered into 2 clusters by hierarchical clustering with Euclidean and Hamming metrics as distance measures. The clustering results on discretized data were evaluated by adjusted Rand index against non-discretized Label (top, gray and black) and True Label (bottom, red and blue). Adjusted Rand indexes between True Labels and non-discretized (normalized by feature scaling for equal frequency/equal interval discretization and Z Score normalization for Z Score discretization) Labels were shown as reference lines in green in the bottom plots of each panel.

(**E**) Information loss/gain after discretization. Ratios of adjusted Rand index adjRand(discretized, True Label) and adjRand(non-discretized, True Label) were calculated for the genomic data sets including Collins_PAM50, Collins_p53, Miller_ER, Miller_p53DLDA and Wang_Stroma. Clustering using Euclidean (top) and Hamming metrics (bottom) were shown. Data were represented as mean + sd.

3 Projecting High-dimensional Genomic Data into Lower Dimensions Using the 1D-Jury Score Function in Linear Time

Abstract

In the previous chapter, we demonstrated that discretization of genomic data still preserved cluster structure and that the ordinal information of the discretized data set was not essential for correct discretization. Based on these two observations, we proposed to adopt the 1D-Jury method, which has been shown to provide a fast clustering method and efficient approach for assessing protein models. In what follows, we specifically used the 1D-Jury approach to efficiently identify biologically relevant low dimensional projections of high dimensional gene expression profiles. We demonstrated that this approach was able to significantly reduce the dimensionality of the feature space while preserving or enhancing the recognition of patterns in gene expression data.

Data

In this chapter, two sets of genomic data were used. They both came with label information.

- ***Collins_SmallPValue[1].*** Collins_SmallPValue data set contains mRNA-sequencing results from four types of invasive breast carcinoma (BRCA) (basal-like, Her2-positive, luminal A, and luminal B), totally 428 samples. One-way ANOVA analysis on the four classes of BRAC samples was performed and top 500 genes with smallest p values were selected for further analysis.
- ***TCGA_4_Cancers.*** mRNA sequencing results from 4 different types of cancers including kidney chromophobe cancer(n = 66), kidney renal clear cell carcinoma(n = 392), ovarian serous cystadenocarcinoma(n = 158) and prostate adenocarcinoma(n = 246) were downloaded separately from cBioPortal[2][2, 3] and merged by common HUGO symbols (gene symbols). Genes with their expression marked as “NA” or “NaN” in any sample were discarded for further analysis. However, we also noticed that kidney renal clear cell carcinoma samples were not properly adjusted whereas other sample types were adjusted such that the mean of gene expression within each sample was 0 and the standard deviation was 1. Therefore, we also adjusted the kidney renal clear cell carcinoma samples to the same standard.

Code

[2] http://www.cbioportal.org/

R scripts used in this chapter can be found: https://github.com/reformasky/xuanThesis.

Methods

Gene program (GP)

GP is a collection of 22 gene lists, totally 6871 genes. Each list contains a gene network module with a biological theme (Table 3-1). Hoadley K et al. have demonstrated that this gene collection has discretionary power to separate 12 cancer types [4]. We first generated a union of all 6871 genes and cross-referenced these genes with the genes from the TCGA-4-Cancers data set. We only kept the genes in both the gene programs lists and the TCGA-4-Cancer data set, resulting a 5372 x 862 matrix.

Table 3-1 Lists of gene programs defined pathways

Gene pathway identifier	Link
PUJANA_CHEK2_PCC_NETWORK	http://www.broadinstitute.org/gsea/msigdb/cards/PUJANA_CHEK2_PCC_NETWORK
KEGG_HEMATOPOIETIC_CELL_LINEAGE	http://www.broadinstitute.org/gsea/msigdb/cards/KEGG_HEMATOPOIETIC_CELL_LINEAGE
DACOSTA_UV_RESPONSE_VIA_ERCC3_DN	http://www.broadinstitute.org/gsea/msigdb/cards/DACOSTA_UV_RESPONSE_VIA_ERCC3_DN
PerouLab --- HS_Red7 :: median || BMC Med Genomics. 2011 | PMID: 21214954	PerouLab --- HS_Red7 :: median || BMC Med Genomics. 2011 | PMID: 21214954
PerouLab --- MM_Myc_1pFDR_UP :: Median || Genome Biology 2007 | PMID:17493263	PerouLab --- MM_Myc_1pFDR_UP :: Median || Genome Biology 2007 | PMID:17493263
RICKMAN_TUMOR_DIFFERENTIATED_WELL_VS_POORLY_DN	http://www.broadinstitute.org/gsea/msigdb/cards/RICKMAN_TUMOR_DIFFERENTIATED_WELL_VS_POORLY_DN
SMID_BREAST_CANCER_BA	http://www.broadinstitute.org/gsea/msigdb/cards/SM

SAL_DN	ID_BREAST_CANCER_BASAL_DN
TTGTTT_V$FOXO4_01	http://www.broadinstitute.org/gsea/msigdb/cards/TTGTTT_V$FOXO4_01
PerouLab --- MClaudin_Cluster :: median || BMC Med Genomics. 2011 | PMID: 21214954	PerouLab --- MClaudin_Cluster :: median || BMC Med Genomics. 2011 | PMID: 21214954
CARBOXYLIC_ACID_METABOLIC_PROCESS	http://www.broadinstitute.org/gsea/msigdb/cards/CARBOXYLIC_ACID_METABOLIC_PROCESS
PerouLab --- MInterferon_Cluster :: median || BMC Med Genomics. 2011 | PMID: 21214954	PerouLab --- MInterferon_Cluster :: median || BMC Med Genomics. 2011 | PMID: 21214954
SEMENZA_HIF1_TARGETS	http://www.broadinstitute.org/gsea/msigdb/cards/SEMENZA_HIF1_TARGETS
MODULE_100	http://www.broadinstitute.org/gsea/msigdb/cards/MODULE_100
MORF_CNTN1	http://www.broadinstitute.org/gsea/msigdb/cards/MORF_CNTN1
NAGASHIMA_EGF_SIGNALING_UP	http://www.broadinstitute.org/gsea/msigdb/cards/NAGASHIMA_EGF_SIGNALING_UP
INTRACELLULAR_SIGNALING_CASCADE	http://www.broadinstitute.org/gsea/msigdb/cards/INTRACELLULAR_SIGNALING_CASCADE
SMID_BREAST_CANCER_BASAL_UP	http://www.broadinstitute.org/gsea/msigdb/cards/SMID_BREAST_CANCER_BASAL_UP
MEMBRANE_COAT	http://www.broadinstitute.org/gsea/msigdb/cards/MEMBRANE_COAT
PerouLab --- HS_Green17 :: median || BMC Med Genomics. 2011 | PMID: 21214954	PerouLab --- HS_Green17 :: median || BMC Med Genomics. 2011 | PMID: 21214954
GNF2_TAL1	http://www.broadinstitute.org/gsea/msigdb/cards/GNF2_TAL1
MORF_MT4	http://www.broadinstitute.org/gsea/msigdb/cards/MORF_MT4
PerouLab --- 16q24x :: median || BMC Med Genomics. 2011 | PMID: 21214954	PerouLab --- 16q24x :: median || BMC Med Genomics. 2011 | PMID: 21214954

1D-Jury function for genomic data sets

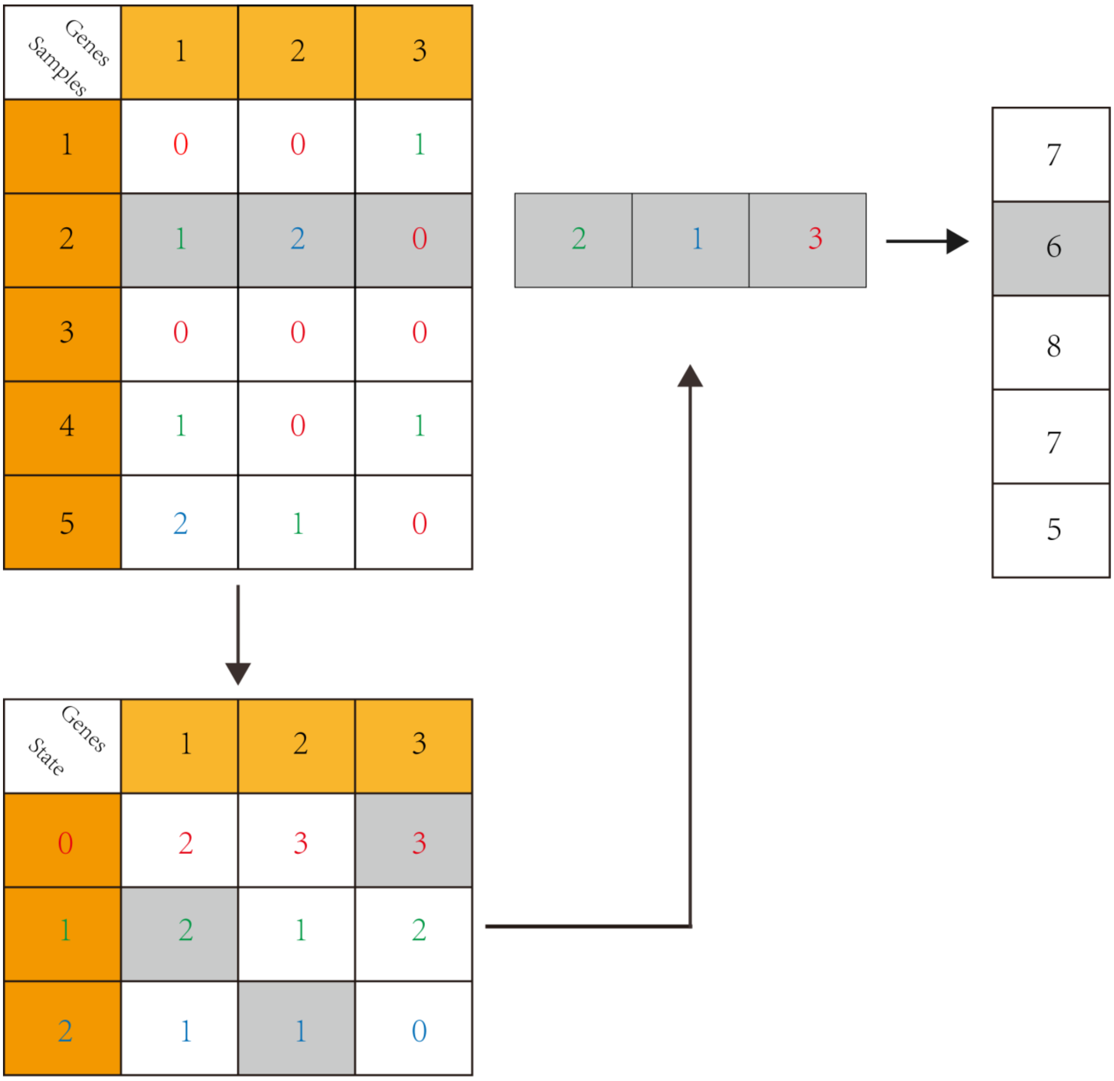
1D-Jury has been introduced as an efficient algorithm that projects a high-dimensional feature space to a much lower one[5]. It takes nominal categorical data and performs dimension reduction in linear time. Since we have demonstrated that discretization of genomic data still preserved cluster structure in chapter 2, it is natural to hypothesize that 1D-Jury can also be used for reducing dimensionality of genomic data.

A 1D-Jury score function $S_{1D}: \boldsymbol{M}_{m*n} \rightarrow \boldsymbol{X}'_m$ maps a discretized matrix representation of a **transposed** subset of genomic data set $\boldsymbol{M}_{m*n}$ which contains gene expression data of n genes from m samples into a vector $\boldsymbol{X}'_{\boldsymbol{m}}$. It is defined as following:

1. similarity between ith gene of samples **X, Y** $\in$ **M**: s($\mathbf{X}^{(i)}$, $\mathbf{Y}^{(i)}$) = 1 if $\mathbf{X}^{(i)}$ == $\mathbf{Y}^{(i)}$ else 0;
2. 1D-Jury score for the ith gene for a sample **X**$\in$ **M**: $\mathrm{S}_{1\mathrm{D}}(\mathbf{X}^{(\mathrm{i})}, \mathbf{M}) = \sum_{\mathrm{Y} \in \mathrm{M}} \mathrm{s}(\mathbf{X}^{(\mathrm{i})}, \mathbf{Y}^{(\mathrm{i})})$
3. 1D-Jury score for the entire sample **X**$\in$ **M**: $\mathrm{S}_{1\mathrm{D}}(\mathbf{X}, \mathbf{M}) = \sum_{\mathrm{i}=1}^{\mathrm{i}=\mathrm{n}} \mathrm{S}_{1\mathrm{D}}(\mathbf{X}^{(\mathrm{i})}, \mathbf{M})$

Since **M** is discretized into *d* distinct states, it is more advantageous to first create a cache matrix $\mathbf{C}_{\mathbf{d} * \mathbf{n}}$ to store the 1D-Jury score for each state at each of **M**'s genes, and then $\mathrm{S}_{1\mathrm{D}}(\mathrm{X}^{(\mathrm{i})}, \mathrm{M})$ can be easily derived as **C**[$\mathbf{X}^{(\mathbf{i})}$, i]. **C**[*s, i*] is defined as the number of elements in *ith* genes of **M** that are equal to state *s*. A graphic demonstration of 1D-Jury score algorithm is shown in figure 3-1.

Figure 3-1 Schematic demonstration of an efficient 1D-Jury score algorithm



Genes / Samples	1	2	3
1	0	0	1
2	1	2	0
3	0	0	0
4	1	0	1
5	2	1	0

2	1	3

7
6
8
7
5

Genes / State	1	2	3
0	2	3	3
1	2	1	2
2	1	1	0

Figure Legend 3-1 Schematic demonstration of an efficient 1D-Jury score algorithm.

A cache matrix **C**$_{d*n}$ to store the 1D-Jury score for each state at each of M's column was first calculated, with the element **C**[s, i] representing the number of elements in ith gene of **M** that are equal to state s. For example, in column 3 of **M,** there are three 0s; therefore, **C**[0, 3] = 3. To calculate 1D-Jury score for sample 1, we can add **C**[1,1], **C**[2,2] and **C**[0, 3] together, and gets 6.

Give the definition of $S_{1D}\colon \boldsymbol{M}_{m*n} \rightarrow \boldsymbol{X}'_{\boldsymbol{m}}$, it is very natural to further define $S_{1D}\colon \boldsymbol{M}_{m*n}, ids_l \rightarrow \boldsymbol{M}'_{\boldsymbol{m*l}}$, where ids_l is an indexing tuple consisting of l lists and the ith list of ids_l represents the indexes of columns in **M** that are used to calculate the ith

column of $\boldsymbol{M'_{m*l}}$. However, since the lengths of lists within ids_l might be different and therefore the scales of columns within $\boldsymbol{M'_{m*l}}$ would be different, it is necessary to normalize the i^th column of $\boldsymbol{M'_{m*l}}$ by dividing the length of $\mathrm{ids}_l^{(\mathrm{i})}$. A schematic illustration of $S_{1D}\colon \boldsymbol{M}_{m*n}, ids_l \rightarrow \boldsymbol{M'_{m*l}}$ is shown in figure 3-2.

Figure 3-2 Schematic illustration of 1D-Jury Score for an entire genomic data set

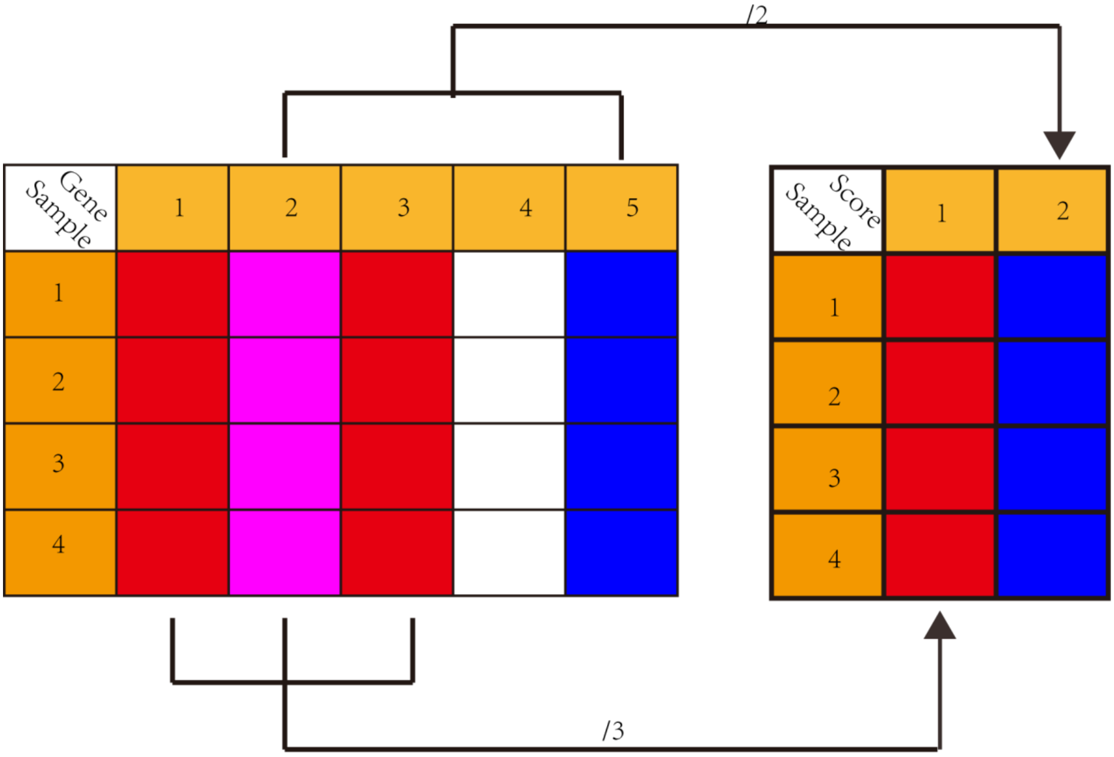


Figure Legend 3-2 Schematic illustration of 1D-Jury Score for an entire genomic data set

A transposed genomic data set was represented by a 4X5 matrix **M**, and an indexing tuple *ids₂* ({1, 2, 3}, {2, 5}) was used to select columns of genes to calculate 1D-Jury scores for subsets of **M.** The first column of the output matrix corresponded to subset {1, 2, 3} and the second column corresponded to subset {2, 5}**.** It is noteworthy that not all genes were involved in the calculation. Column 4 in **M** was not used at all. Additionally, some columns were shared by several lists in *ids*. For example, column 2 in **M** was used to calculate 1D-Jury score for both the red and blue group. In order to make sure features of the derived **M**$'$ were on the same scale, it was necessary to divide the i^th column of the derived 1D-Jury scores by the length of $\mathrm{ids}_2^{(\mathrm{i})}$, 3 and 2 respectively.

Results

Dimension reduction of Collins_SmallPValue data set by 1D-Jury score dampened cluster details but still preserved overall cluster structure

We first examined 1D-Jury function as a dimension reduction approach on the Collins_SmallPValue data set, which contained mRNA sequencing results of 500 genes from 428 BRCA samples. We randomly partitioned the 500 genes into 20 groups with 25 genes per group and used this partition as the indexing tuple ids_{20}. We discretized the Collins_SmallPValue data set into matrixes with 3, 4 and 5 discrete states and then transformed the discretized matrixes into 20 X 428 matrixes using our 1D-Jury function according to ids_{20}. For discretization, we used equal frequency, equal interval and Z Score discretization functions, all of which have been discussed in Chapter 2. We then clustered the projected matrixes using complete linkage hierarchical clustering with Euclidean distance. As comparisons, we also performed similar hierarchical clustering on the non-discretized but normalized data sets (by feature scaling and Z Score normalization functions). As shown in Figure 3-3 (**A** and **D**), before discretized-projection, the normalized Collins_SmallPValue data sets had obvious cluster structures, with the basal-like cancers samples were clearly separated from the other 3 types of cancers samples. We could also observe local aggregations of Her2-positive and luminal B samples within the luminal A samples. The clustering on discretized-projected data, still kept the basal like cluster separate from other samples (**B, C** and **E**). However, except for the one discretized by Z Score function (**E**), we could no longer observe local clusters of Her2 positive samples. In addition, we could only see small clusters of luminal B samples when clustering using the discretized-projected data sets (**B, C** and **E**). Even with loss of some fine grain details, we still observed significant resemblance between clustering using discretized-projected data and clustering using non-discretized data, especially for equal interval and Z Score discretized data sets (**F,** top). In addition, the similarity between clustering using discretized-projected data and true labels were quite comparable with the similarity between clustering using non-discretized data and true labels (**F,** bottom), suggesting that although discretization and 1D-Jury score projection lost some fine grain details of clustering, our operation still preserved the overall cluster structure.

Dimension reduction of TCGA_4_Cancers data set by 1D-Jury score enhanced cluster structure

It was assuring to observe that dimension reduction of Collins_SmallPValue data set with 1D-Jury score still preserved its cluster structure. However, two concerns regarding the aforementioned approach should be addressed. First, the selection of genes representing the data set still involved the usage of cluster labels; therefore, the method used in the previous section did not fully qualify the definition of clustering analysis. Second, the indexing tuple ids$_{20}$ was generated by random partition, which did not consider any common biological themes and therefore might be vulnerable to noise and redundancy. Ideally, we should find the indexing tuple ids containing non-redundant modules that were essential for all biological samples but also discretionary between different types of samples. Unfortunately, due to the massive diversity of biological samples, it is very hard to derive this index tuple. According to our knowledge, no indexing tuple has demonstrated such universal discretionary power. However, if we would limit our samples to cancer tissues, a gene collection containing 22 lists of genes termed Gene Programs (GPs) have been shown to be capable of separating 12 cancer types apart [4]. Based on this, we decided to test whether we were able to properly cluster the discretized cancer genomic data using 1D-Jury score projection according to GPs. Here we used TCGA_4_Cancers data set described in the Data section as an example.

Similarly to our procedures on the Collins_SmallPValue data set, we also first discretized our entire data set per gene into 3, 4 and 5 distinct states using equal frequency, equal interval and Z Score discretization functions. We then projected the discretized data into lower dimensional matrixes using 1D-Jury score function according to GPs. We performed complete linkage hierarchical clustering on the discretized-projected data sets using Euclidean distance as distance measure. As always, we also included the same hierarchical clustering on normalized but not discretized (by feature scaling and Z Score normalization functions) data sets. As demonstrated in Figure 3-4, clustering using both feature scaling and Z Score normalized data sets generated a distinction cluster of kidney renal clear cells carcinoma cells (**A** and **D**). We could also observe small clusters of ovarian serous cystadenocarcinoma samples in the normalized data sets. However, it was obvious that the overall dendrograms from both normalized data set were highly skewed and the distances within kidney renal clear cell carcinoma samples were much shorter than other sample types. In contrast, clustering on discretized-projected data set generated a more balanced cluster structure. Not only did they properly separate the kidney renal clear cell carcinoma samples apart from others, they were also able to separate ovarian serous cystadenocarcinoma and prostate adenocarcinoma samples from each other. The group labels derived by cutting the dendrograms displayed significant similarity with the clustering result using feature scaling normalized data. One interesting observation is that clustering using equal frequency discretized data highly depended on the number of states, whereas the other two discretization functions were not sensitive to the number of states. Quantitatively speaking, clustering using the discretized-projected data generated results much better than using the normalized data, especially for the Z Score discretized data. In summary, we demonstrated that with proper selection of the indexing tuple and discretization functions, dimension reduction through 1D-Jury score function was able to enhance cluster structures which were otherwise buried within the high dimensional space.

Conclusions

In this chapter, we demonstrated that 1D-Jury was an efficient approach to significantly reduce dimensionality of the feature space and at the same time enhanced the cluster structures. We also showed by using gene collections such as GPs, we could properly cluster genomic data without relying on prior knowledge of sample identities.

Besides enhancing cluster structure of genomic data sets, projection using 1D-Jury score is also beneficial in reducing the computational complexity of clustering. Typical clustering algorithms involve the calculations of distances between pairs of patterns, and for data in high-dimensional feature space, calculating the pair-wise distances might be computationally expensive. After projecting the data into low dimensional space, distance calculations will become much more preferable.

There are a lot of successful dimension reduction algorithms proposed, including principal component analysis (PCA), linear discriminant analysis (LDA), and canonical correlation analysis (CCA). However, most of these methods are computationally expensive. For example, the time complexity of PCA is $O(\min(m^3, n^3))$, where m and n are the column and row dimensionality of the original matrix. In contrast, the complexity of our 1D Jury score is linearly dependent on the number of samples O(m), which makes our algorithm much more efficient than these classical dimension reduction techniques.

It is noteworthy that no matter how low the dimensionality of the projected space becomes, the overall complexity of clustering algorithm remains the same. For complete linkage hierarchical clustering that was used intensively in this study, the complexity is still $O(m^2)$, where m represents the number of samples to be clustered. What we have achieved in this chapter is to reduce the constant multiplier of $O(m^2)$.

References

[1] F. S. Collins and A. D. Barker, "Mapping the cancer genome. Pinpointing the genes involved in cancer will help chart a new course across the complex landscape of human malignancies," *Sci Am,* vol. 296, pp. 50-7, Mar 2007.

[2] J. Gao, B. A. Aksoy, U. Dogrusoz, G. Dresdner, B. Gross, S. O. Sumer*, et al.*, "Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal," *Sci Signal,* vol. 6, p. pl1, Apr 2 2013.

[3] E. Cerami, J. Gao, U. Dogrusoz, B. E. Gross, S. O. Sumer, B. A. Aksoy*, et al.*, "The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data," *Cancer Discov,* vol. 2, pp. 401-4, May 2012.

[4] K. A. Hoadley, C. Yau, D. M. Wolf, A. D. Cherniack, D. Tamborero, S. Ng*, et al.*, "Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin," *Cell,* vol. 158, pp. 929-44, Aug 14 2014.

[5] R. Adamczak, J. Pillardy, B. K. Vallat, and J. Meller, "Fast geometric consensus approach for protein model quality assessment," *J Comput Biol,* vol. 18, pp. 1807-18, Dec 2011.

Figures

Figure 3-3 Dimension reduction of Collins_SmallPValue data set by 1D-Jury score dampened cluster details but still preserved overall cluster structure

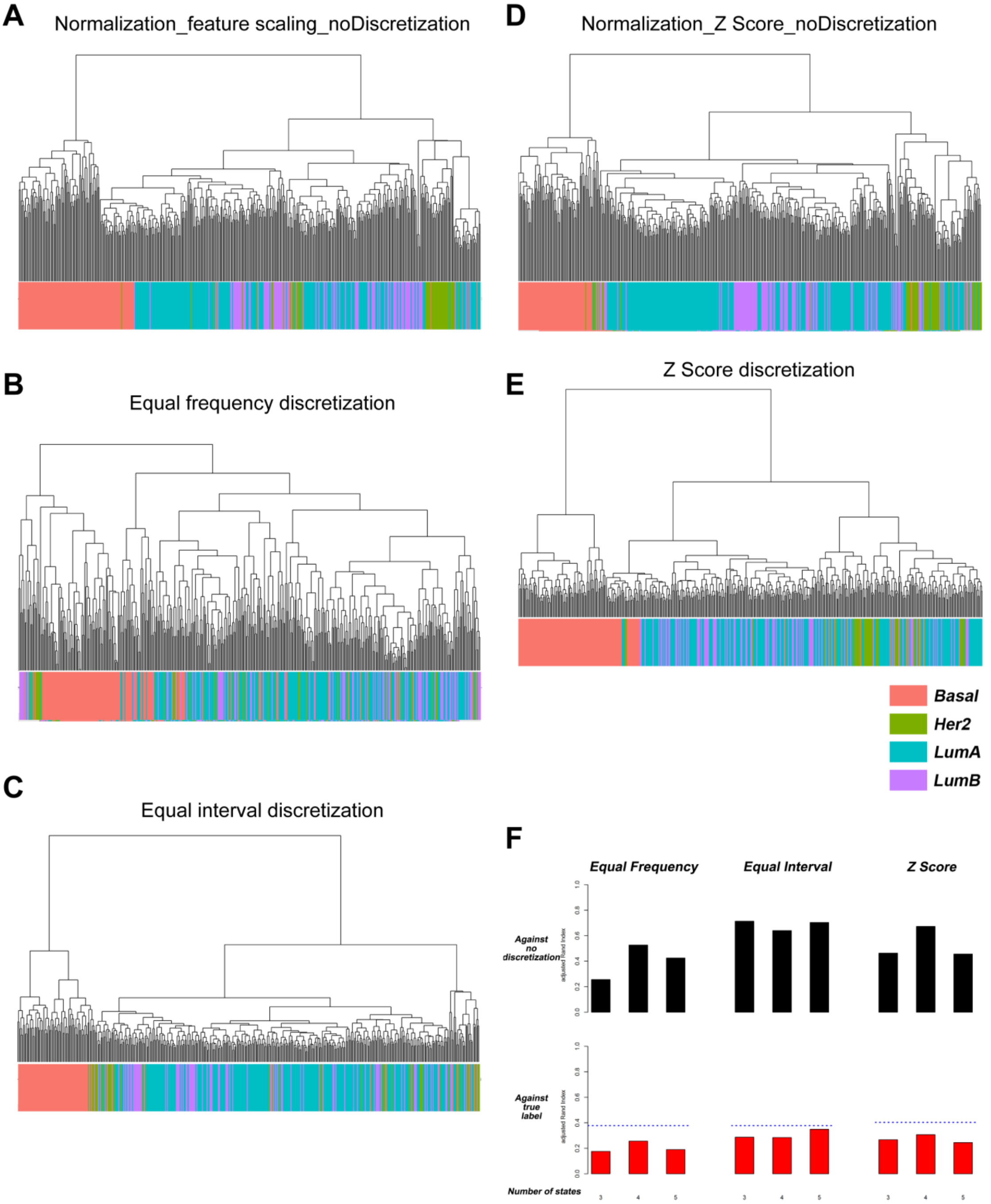


Figure 3-4 Dimension reduction of TCGA_4_Cancers data set by 1D-Jury score enhanced cluster structure

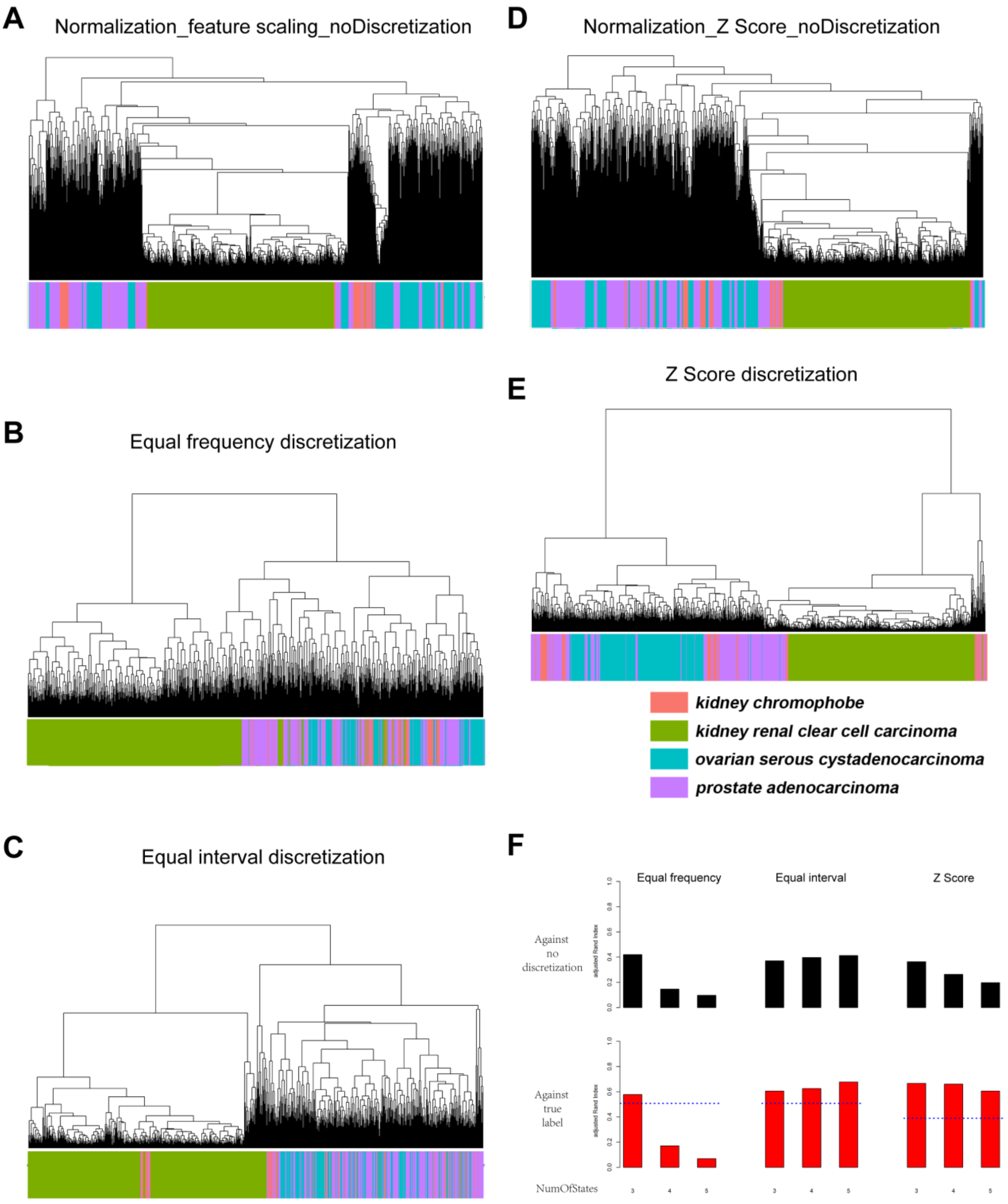


Figure Legends

Figure Legend 3-3 Dimension reduction of Collins_SmallPValue data set by 1D-Jury score dampened cluster details but still preserved overall cluster structure

(**A -E**) Dendrograms (top) and distributions of True Label (bottom) after clustering normalized (**A** and **D**) and discretized-projected (**B**, **C** and **E,** number of states = 3) Collins_SmallPValue data sets. Normalization and discretization methods were shown as titles of each panel. Complete linkage hierarchical clustering with Euclidean distance was used.

(**F**) Evaluation of clustering discretized-projected Collins_SmallPValue data set. The Collins_SmallPValue data set was discretized by equal frequency, equal interval and Z Score functions and projected into lower dimensions by 1D-Jury score function. Discretized-projected data were clustered into 4 clusters. The clustering results on discretized-projected data were evaluated using adjusted Rand index against non-discretized (but normalized) Label (top, black) and True Label (bottom, red). Adjusted Rand indexes between True Labels and clustering results using non-discretized (normalized by feature scaling for equal frequency/equal interval discretization and Z Score normalization for Z Score discretization) were shown as blue reference lines in the bottom plot.

Figure Legend 3-4 Dimension reduction of TCGA_4_Cancers data set by 1D-Jury score enhanced cluster structure

(**A -E**) Dendrograms (top) and distributions of True Label (bottom) after clustering normalized (**A** and **D**) and discretized-projected (**B**, **C** and **E,** number of states = 3) TCGA_4_Cancers data sets. Normalization and discretization methods were shown as titles of each panel. Complete linkage hierarchical clustering with Euclidean distance was used.

(**F**) Evaluation of clustering discretized-projected TCGA_4_Cancers data set. The TCGA_4_Cancers data set was discretized by equal frequency, equal interval and Z Score functions and projected into lower dimensions by 1D-Jury score function. Discretized-projected data were clustered into 4 clusters. The clustering results on discretized-projected data were evaluated using adjusted Rand index against non-discretized Label (top, black) and True Label (bottom, red). Adjusted Rand indexes between True Labels and clustering results using non-discretized (normalized by feature scaling for equal frequency/equal interval discretization and Z Score normalization for Z Score discretization) were shown as blue reference lines in the bottom plot.

4 Summaries and Perspectives

In this study, we empirically demonstrated that discretization of genomic data largely preserved the cluster structure while providing a basis for efficient clustering of genomic data. Based on these findings, we further investigated the usage of 1D-Jury score, the efficient implementation of which relied on nominal categorical data, for dimension reduction of genomic data as a pre-processing step for clustering analysis. We demonstrated that our discretization and 1D-Jury projection method efficiently reduced the dimensionality of feature space, while reducing the noise and improving the discovery of the cluster structure in the original data set. Therefore, the discretization-projection heuristic was able to significantly reduce the overall computational complexity while improving clustering outcome.

Our discretization-projection procedure reduced the complexity of calculating distances between pairs of patterns in the pattern set. For most distance measures, the complexity of distance calculation is linearly dependent on the dimensionality of the patterns. Our procedure did not alter the overall complexity of clustering analysis; and for complete linkage hierarchical clustering, the complexity is still $O(m^2)$, where m is the number of patterns in the pattern set. What is implicit here is that for this $O(m^2)$ algorithm, there is a very large constant multiplier when clustering the original data set, and our dimension reduction approach successfully reduced this multiplier by magnitudes.

However, with the advance of sequencing technology and reduction of sequencing cost, the number of samples within a genomic data set in the future might still become too large to cluster, even with the reduced dimensions. Therefore, it might be necessary to develop more efficient clustering algorithms for the fast growing of genomic data. Dr Meller and colleagues have proposed an ultrafast clustering algorithm on macromolecular structures based on geometric hashing technique called uQlust (unpublished data). This method takes a matrix of discrete states and clusters the samples in linear time. Since our discretization methods have been shown to maintain cluster structure, we hypothesize that we can also adopt uQlust to cluster the discretized (and possibly 1D-Jury score based dimension reduced) genomic data.

Throughout this thesis, we have focused on clustering mRNA-sequencing data, which quantified the expression of transcripts within samples. Genomic alteration analysis is another major type of genomic studies, which studies the single nucleotide polymorphism (SNP), mutation, insertion and deletion. Typically, this kind of studies generates categorical data, and therefore the discretization step proposed in this study would be unnecessary. However, since massive genomic alterations have been found, it might still need the dimension reduction operations such as the 1D-Jury score method. Future studies should investigate whether 1D-Jury score based feature extraction algorithms are applicable in genomic alternation data sets.

Another potential application of our discretization-projection operations is to generate gene signatures that are able to distinguish different categories of samples, which can be used for supervised classification and unsupervised clustering. Examples of the gene signatures include the PAM50 gene list used in chapter2, which is a very robust short list of genes capable of distinguish different types of BRCA samples. Another example is the gene programs used in chapter 3, which contains 22 lists of gene modules, each of which contains different number of genes with a common theme. The disadvantage of using individual genes is that expression of individual gene suffers from noise due to biological variations and sampling errors, and at the same time is vulnerable to redundancy, since genes in the same pathway tend to correlate with each other. On the other hand, using all genes from a gene list might be computationally expensive. In contrast, summarizing a list of genes belonging to a common theme using methods such as 1D-Jury score might be able to provide a more robust and concise gene signatures. One natural approach is to select 1D-Jury scores that are significantly different across different categories of samples. As a preliminary study, we used the TCGA_4_Cancers data set as an example. Using the same operations with chapter 3, we discretized and projected this data set using the 1D-Jury score function. However, instead of using 1D-Jury scores corresponding to all 22 gene lists, we performed one-way ANOVA analysis to find gene lists whose 1D-Jury scores were most significantly different across all 4 types of cancer samples and only used their corresponding 1D-Jury scores for clustering analysis. As shown in figure 4-1, 1D-Jury scores from the top 5 gene lists were able to cluster the samples equally well with 1D-Jury scores from all 22 gene list when using Z Score discretization and for equal frequency and equal interval discretization, we needed to use the top 10 gene lists, suggesting that we could further narrow down our gene lists to generate minimal gene signatures for classification and clustering analysis. It might be interesting to investigate the application of discretization and 1D-Jury score projection in generating gene signatures using other data sets.

Figure 4-1 Feature extraction on the discretized-projected TCGA_4_Cancers data set

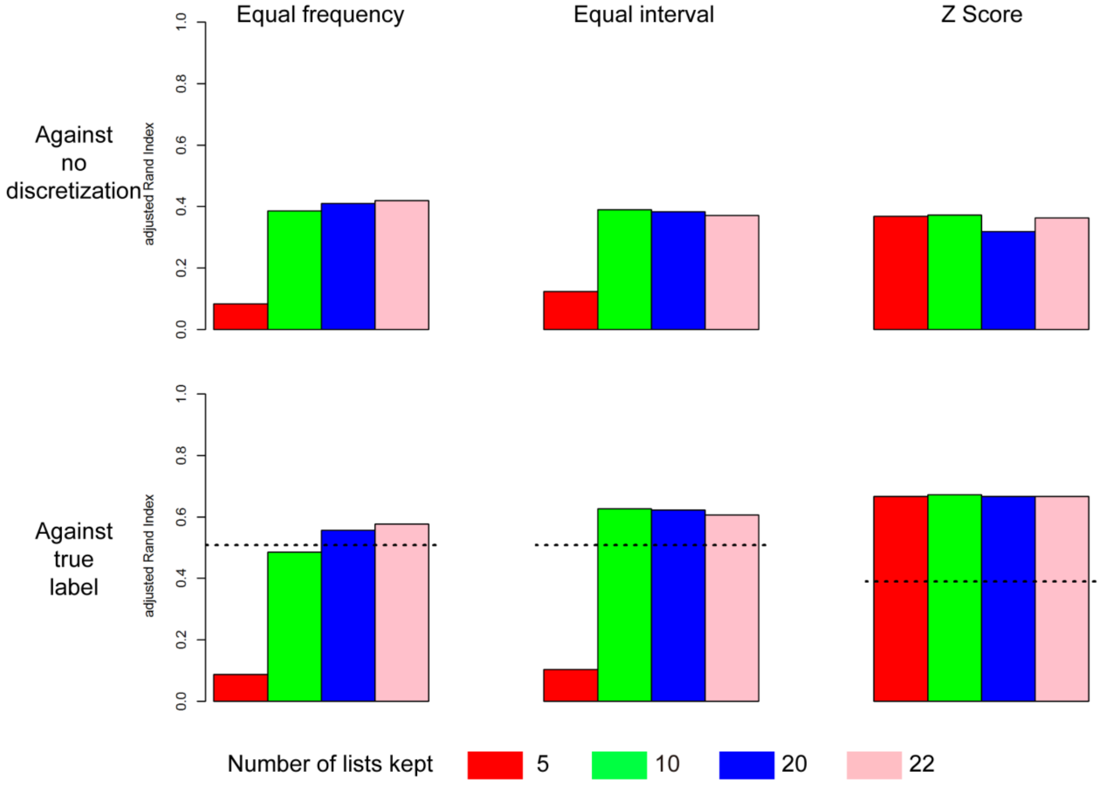


Figure Legend 4-1 Feature extraction on the discretized-projected TCGA_4_Cancers data set

Equal frequency, equal interval and Z Score discretization functions were used to discretized the TCGA_4_Cancers data set into 3 states and 1D-Jury scores for 22 gene lists in GPs were calculated using the 1D-Jury score function introduced in chapter 3. One-way ANOVA analysis were used to find gene lists whose corresponding 1D-Jury scores were most different across the 4 types of cancers. Top 5, top10, top 20 and all 22 gene lists were selected and hierarchical analysis were performed to cluster the samples represented by 1D-Jury scores corresponding to these gene lists. Top: adjusted rand indexes between clustering result and clustering using non-discretized data set; bottom: adjusted rand indexes between clustering result and True Label of the samples. In addition, adjusted rand indexes between clustering using non-discretized data set and True Label of the samples were shown in black in the bottom as reference.