ISSN:0975-9646

Analysis Clustering Techniques in Biological Data with R

Satish Kumar
Department of Computer Science
BGSBU, Rajouri (J&K) -185131, INDIA

Dr.Mohammed Asger
School of mathematical Sciences & Engineering
BGSBU, Rajouri (J&K) -185131, INDIA

***Abstract*—Clustering has been widely recognized as a powerful data mining technique. Clustering is an unsupervised learning technique, based on the concept of intra-clustering and inter-clustering. Clustering of biological data is well researched topic among computer sciences. Bio-informatics has become area that received most of the attention. In general bio-informatics aims to solve complicated problems. Examples: - categorized gene with their functionality, analysis of gene expression data obtained from micro-array experiments etc. However, the large number of genes and the complexity of biological networks greatly increase the challenges of comprehending and interpreting the resulting mass of data, which often consists of millions of measurements. A first step toward addressing this challenge is the use of clustering techniques with R.**

Clustering techniques are used for extracting/analyzing the biological structures. The different methods we study of clustering are k-mean, Self Organization Map (SOM), Hierarchical Clustering algorithms for Biological Data and their comparison using R programming tool.

***Keywords*: - k-mean algorithm, Self Organization Maps algorithm, Hierarchical Clustering algorithm.**

I. INTRODUCTION

Until recently, information technologies have established the distribution and storage of data more comfortable. Vast quantities of data accumulated at much higher speed. Still, prefect data are not that practicable because what people want is information hidden in the data sets. Information consider as the characteristics or patterns of the data. This information is practically more worthful than data. So, a new technology field has emerged to deal with the discovery of information from database called data mining. Uncovering hidden information is the goal of data mining. But the uncovered information must be [1]:

- **Correct:** Inappropriate selection of data will generate incorrect solutions. The mined information needs to be care-fully verified by domain experts.
- **New:** Common sense or known facts are not what is searched for.
- **Applicable:** The mined information should be able to be utilized in a certain problem domain.
- **Meaningful:** The mined information should mean something that can be easily realized.

To discovery information from data, data mining uses technologies from different information science and computer fields. The three major ones are *statistics*, *machine learning*, and *databases* [2]. Using high-throughput biotechnologies, biological data like RNA, DNA, and protein data are generated at high speed that store in datasets. In order to manage and analyze such large and complex data sets biologist needs data mining technologies. Web technologies and database are used to build a lot of online data banks for data storage. Large amount of data collected have been put on the World Wide Web and can be accessed and shared online.

II. HOW ALGORITHMS ARE IMPLEMENTED?

R TOOL: R is public domain software primarily used for statistical analysis and graphic techniques [17]. Before, R S language was used for statistical analysis but R has different implementation. S code can run on R tool without alteration R has more than 4 thousand packages in CRAN repository which include different functions used for analysis purposes. R tool provides a wide class of statistical that includes classical statistical tests, linear and nonlinear modeling, classification, time-series analysis, clustering, and various graphical functions.

PACKAGES FOR R: R uses collections of packages to perform different functions [18]. CRAN project Views provide numerous packages to different users according to their taste. R package contain different functions for data mining approaches. Few of them are listed below:-

- Statistical Learning.
- Classification & Cluster Analysis.
- Machine Learning.
- Sequential Patterns.
- Interface to Weka.
- Spatial Data analysis.
- Time Series data Analysis.

DATASET: To test the unsupervised clustering algorithms and compare among them is obtained from the site: (http://kdd.ics.uci.edu/. The data repositories used in this paper are The Iris Repository, Wine Repository [27].

TABLE I CLUSTER ANALYSIS PACKAGES (CRAN, BIOCONDUCTOR) IN R TOOL

Packages	Task	Arguments
1.cluster	Agnes	AGglomerative NESting
	Diana	DIvisive ANAlysis
	Pam	Partitioning Around Medoids
2.Stats	Heatmap	Heatmaps with row and column dendrograms
	cophenetic, hclust	Hierarchical clustering
3.hopach	Boothopach, hopach	Hierarchical Ordered Partitioning and Collapsing Hybrid

The aim of this work is the implementation and evaluation of several commonly used clustering algorithms for biological data. The program has to fulfil the following requirements:

- Import and export of multiple experiment datasets from flat files
- Graphical representation of the dataset in a user friendly and intuitive way using R tool
- Tools for data adjustment to gain a best possible representation for further analysis
- Facilitation of several distance measuring procedures
- Implementation of *k*-means Clustering, Hierarchical Clustering, Self Organizing Maps (SOM).
- 2 and 3-dimensional representation of the clustering results including the ability to export the results as images and data.

III APPROACHES USED FOR CLUSTERING

A. *HIERARCHICAL CLUSTERING*

Hierarchical clustering is an unsupervised procedure of transforming a distance matrix which is a result of pair wise similarity measurement between elements of a group, into a hierarchy of nested partitions. The hierarchy can be represented with a tree-like dendrogram in which each cluster is nested into the next cluster. Hierarchical algorithms can be further cate-gorized into two kinds [3]:

1. Agglomerative procedures: This procedure starts with *n* clusters and iteratively reduces the number of clusters by merging the two most similar objects or clusters, respectively, until only one cluster is remaining ($n \rightarrow 1$).

2. Divisive procedures: This procedure starts with 1 cluster and iteratively splits a cluster, so that the heterogeneity is reduced as far as possible ($1 \rightarrow n$).

If it is possible to find a reasonable distance definition between clusters, agglomerative procedures are less computationally expensive than divisive procedures, since in one step two out of maximum *n* elements have to be chosen for merging, whereas in divisive procedures, fundamentally all subsets have to be analyzed so that divisive procedures have an algorithmic complexity in the magnitude of O(2n) [4]. Agglomerative procedures have the drawback that an incorrect merging of clusters in an early stage often yields results, which are far away from the real cluster structure. Divisive procedures immediately start with interesting cluster arrangements and are therefore much more robust. Usually agglomerative procedures are used because of their efficiency.

The Hierarchical Clustering algorithm [25] below takes an n×n distance matrix d input and increasingly gives n different partitions of the data as the tree it outputs result. The largest partition has n single-element clusters, with every element forming its own cluster. The second-largest partition aggregates the two closest clusters from the largest partition, and thus has n − 1 clusters. In general, the i^{th} partition combines the two closest clusters from the $(i-1)^{th}$ partition and has (n − i + 1) clusters.

HIERARCHICAL CLUSTERING (d, p)

i. Start with n clusters each has 1 data items.
ii. Allotting an isolated vertex to each cluster to construct a graph H
iii. while there is more than 1 cluster
iv. Get two nearest clusters R_1 and S_1
v. Form new cluster R_2 on $|R_1|+|S_1|$ data items by combine R_1 and S_1 clusters
vi. Compute distance from R_2 to all other clusters
vii. Add a new vertex R_2 to S_2 and connect to vertices R_2 and S_2
viii. Remove rows and columns of d corresponding to R_2 and S_2
ix. Add a row and column to d for the new cluster R_2
x. Return H

Different formulas of calculating distance for hierarchical clustering techniques distances give different result from the same.

IMPLEMENTATION OF HIERARCHICAL CLUSTERING IN R: In this segment hclust( ) function on iris data is used to shows hierarchical clustering. First 50 records taken as sample from the iris data in order to ensure that clustering plot will not crowded. Subsequently apply hierarchical clustering to the data set.

```
> hci ←sample(1:dim(iris) [1], 40)
> SampleOiris ←iris[hci]
> SampleOiris$Species ←NULL
> hciR ←hclust(dist(SampleOiris),method="ave")
> plot(hciR, hang = -1,labels=iris$Species[hci])
    > # split tree in three clusters
    > rect.hclust(hciR, k=3)
> hcgroups ←cutree(hciR, k = 3)
```

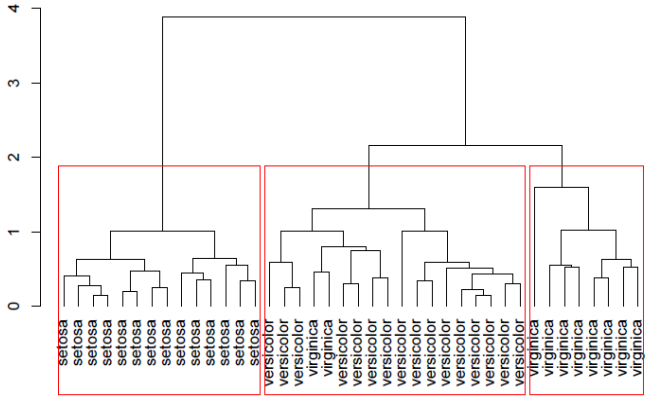

Figure 1 Cluster Dendrogram for Iris Dataset

B. *K-MEANS CLUSTERING*

In clustering commonly used method is *k-means* because it is based on a very simple principle and provides good results [5]. It is unsupervised and can be seen as a Bayesian (maximum likelihood) approach to clustering.

The *k-means* basic approach is to preserve two estimates:

- First is the center location for each cluster and
- Second is the partition of the data points according to which one goes into which cluster.

One estimate can be used to refine the other. If we have an estimate of the center locations, then (with reasonable prior assumptions) the maximum likelihood solution is that each data point should belong to the cluster with the nearest center. Hence, we can compute a new partition from a set of center locations, i.e. make one cluster from the set of vectors in each Voronoi cell.

For this reason, the k-means algorithm proceeds by a sequence of phases in which it alternates between moving data points to the cluster of the nearest center, and moving all cluster centers to the mean of their Voronoi sets.

K-Means clustering method [3] proposed for segmentation of x_n data items in data set into *k* clusters such that in each cluster data items link to cluster with the nearest mean value. The k-Means method gives k different cluster with largest differentiation. This method uses the squared error function. The objective of K-Means clustering method is to minimize squared error criterion or total intra-cluster variance:

$$j = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^{(j)} - c_j \|^2$$

Where j is objective function
k is number of clusters
n is number of cluster
x_i = case i
c_j is centroid of cluster j

The *k*-Means clustering algorithm is extensive clustering techniques to produce effective outcome in gene expression analysis. The k-algorithm randomly selects an arbitrary partition of points into k clusters and tries to improve this partition by moving some points between clusters. In the beginning one can choose arbitrary k points as cluster representatives.
The algorithm iteratively executes the following two steps until it converges:

- Assign each data item to the nearest cluster C_i center representative x_i ($1 \leq i \leq k$)
- After making assignment of all n data items, recomputed new cluster mode.
- Repeat till Convergence criterion satisfied

The K-means clustering algorithm is most popular, simple and fast clustering techniques. The time complexity of K-mean is ***O(n*k*d*i)*** [14] Where n represent number of data items, k represent numbers of clusters, d represent attributes of data items and represent number of iterations. The k-mean algorithm generally required few number of iteration to converge. The k-mean has various limitations as clustering algorithm for gene expression. Some of them are as follow:

- To predict number of clusters in a gene expression data set is generally unidentified in advance. In order to find the optimal number of clusters k-Mean algorithm run iteratively with various values of ***k*** and equates the clustering outcome. K-Mean algorithm is not fine tune process for huge volume of gene expression data set which holds thousands of genes.
- Gene expression data typically contain a huge amount of noise; however, the K-means algorithm forces each gene into a cluster, which may cause the algorithm to be sensitive to noise.

IMPLEMENTATION OF K-MEANS IN R

As mention early the number of clusters in *k-means* has to be set in advance [6]. For this analysis, *k*=4 was used for the number of clusters. As it is demonstrated here, hierarchical clustering can be used to predetermine the number of clusters for a specific dataset.In this segment kmean( ) function on iris data is used to shows partition clustering. First remove species column from the iris data set. After apply function kmeans( ) iris data set then clustering result store in kmeans outcome. In kmean( ) function the number of cluster is set to 3 in the following code .

```
> IrisDS ←iris
> IrisDS$Species ←NULL
> (kmeans.outcome ←kmeans(IrisDS, 4))
The K-means clustering with 4 clusters each of sizes 32, 50, 40, 28
Cluster mode :
   Sepal.Length Sepal.Width Petal.Length Petal.Width
1 6.912500    3.100000     5.846875    2.131250
2 5.006000    3.428000     1.462000    0.246000
3 6.252500    2.855000     4.815000    1.625000
4 5.532143    2.635714     3.960714    1.228571
Clustering vector:
[1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 4
 [55] 3  4 4 3 4 3 4 3 3 3 3 3 3 3 4 4 4 4 3 4 3 3 3 4 4 4 3 4 4 4 4 4 3 4 4 1 3 1 1 1 1 4 1
[109] 1 1 3 3 1 3 3 1 1 1 1 3 1 3 1 3 1 1 3 3 1 1 1 1 1 3 3 1 1 1 3 1 1 1 3 1 1 1 3 3 1 3
Inside cluster sum of squares by cluster:
[1] 18.703437 15.151000 13.624750 9.749286
 (between_SS / total_SS = 91.6 %)
> table(iris$Species, kmeans.outcome$cluster)
            1  2  3  4
  setosa     0 50  0  0
  versicolor 0  0 23 27
  virginica 32  0 17  1
```

Figure 2 Shows clusters and their mode are diagram. There are four dimensions in the data and only the first two dimensions are used to draw the diagram below.
Some black points close to the green center (asterisk) are actually closer to the black center in the four dimensional

space. We also need to be aware that the results of k-means clustering may vary from run to run, due to random selection of initial cluster centers.

```
> plot(IrisDS[c("Sepal.Length", "Sepal.Width")], col = kmeans.outcome$cluster)
> points(kmeans.outcome$centers[,c("Sepal.Length", "Sepal.Width")], col = 1:3, +  pch = 8, cex=2)
```

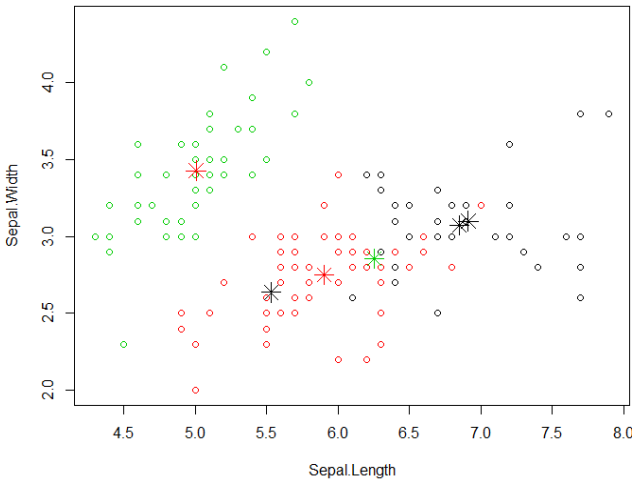

Figure 2 Shows four Clusters and their modes are diagrammed

C. *SELF ORGANIZATION MAP CLUSTERING*

Self-Organizing Maps (SOM) formulated by Prof. T Kohonen in1980s Self-organizing Maps (SOM) belongs to the family of neural network based on ***Competitive learning***. SOM learn through unsupervised learning process. It is applied for analysis and visualization of high dimensional dataset. In data mining SOM is used for clustering and presenting high dimensional datasets into lower dimensional normally 1-D, 2-D and 3-D. As Self-Organization map is an unsupervised learning algorithm, it does not need teacher and learns to group data items without any oversight.

In SOM, [26] neurons are presented by the nodes grid to which data items are introduced. All nodes of grid linked with inputs and no link exist between nodes of grid. The main function of SOM is translating input observation into 1-Dimensional or 2-Dimensional map. SOM is based on topological ordered technique.

Like K-means incremental clustering algorithm data items from dataset processed in each iteration and nearest mode is updated at a time. But SOM enforce topographical setting on mode, the neighbors mode are also updated. This process continues until some convergence limit encounter or where the mode values do not vary. The SOM approach generate outcome as set of mode that clearly define clusters. The cluster generated by SOM comprises of data points nearest to mode.

SOM approach is applied in many fields like gene array data or visualizing Web documents.

In Self Organization, the neurons are placed at the nodes of the lattice that is usually two dimensional. In SOM one dimensional or higher than two dimensional maps are also possible but not so common.

The principal goal of the SOM is to transform an incoming signal pattern of arbitrary dimension into a one or two dimensional discrete map and to perform this transformation adaptively in topological ordered fashion. The first application area of the SOM algorithm was speech recognition, or more accurately, speech to text transformation. Since then it has been used for variety of reasons. One of the most famous is WEBSOM.

In the course of training process these neurons become selectively tunes to various input patterns or classes of input patterns. The location of the neurons so tuned becomes ordered with respect to each other in such a way that a meaningful coordinate system for different input features is created over the lattice.

The SOM based on competitive learning, where the output neurons of the network compete among themselves to be activated, with the result that only one output neuron that wins the competition is on at any one time. An output neuron that wins the competition is call winner take all neurons or simply a winning neuron. The learning process in SOM is unsupervised, meaning that no teacher is required to define the correct output for an input.

SELF- ORGANIZATION MAP ALGORITHM: The technique responsible for the SOM continues first by initializing the synaptic weights in the network. First assigning small values select from random number generators, in doing so no prior order is imposed on the SOM map. After properly initialized, there are three major steps involved in the formation of the SOM, as described below:

Competition: The neurons in the map compute their respective values of a discriminant function for each input patterns. This discriminant function gives the basis for competition among the neurons network. The neuron with the largest value of discriminant function is declared winner of the competition.

Cooperation: Among the competition the winning neuron find the spatial location of the topological neighborhood of excited neurons.

Synaptic adaptation: The last process enables topological neighborhood the excited neurons to increase their individual values of the discrimenant function in relation to the input pattern through suitable adjustment applied to their synaptic weights. The adjustments made are such that the response of the winning neuron to the subsequent application of a similar input pattern is enhanced.

IMPLEMENTATION OF SELF ORGANIZATION MAP IN R

In order to implement the SOM in R the SOM( ) function is used .Using SOM function 177-sample from wine dataset are mapped to 5×4 hexagonally units. In R programming tool first load the package and then wine dataset.

Input data: The kohonen package includes dataset which is used as input data. This dataset carrying 177 rows and 13 columns, data items vintages comprised of family labels. The data obtained from the results of chemical analyses of wines produced in region in Italy, but it was

derived from 3 different varieties: - Grignolino, Barberas and Nebbiolo grapes kinds. These three types of wines contain information about quantities of several constituents and some spectroscopic variables of wine.

Execution: The dataset used to create the SOM map with the following code.

```
> Library (kohonen)
Loading required package: class
> data( wines )
> WineSc ← scale(wines)
> set.seed(7)
> Wine.SOM ← som( data=WineSc, grid = somgrid(5,4, "hexagonal"))
>  plot(Wine.SOM, main = "Wine data")
```

Output

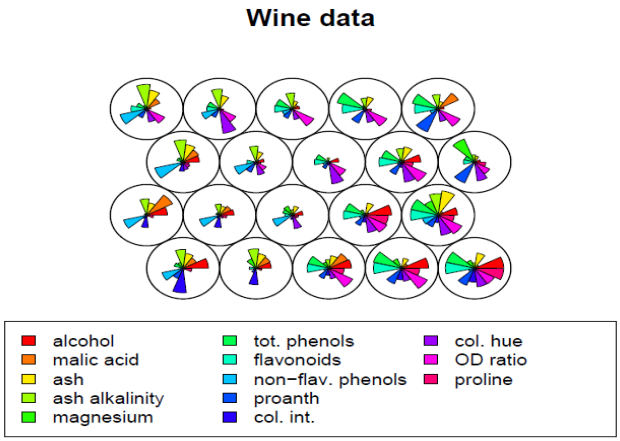

Figure 3 A plots of the codebook vectors of the 5-by-4 mapping of the wine data

TABLE III COMPARATIVE ANALYSIS OF K-MEANS, HIERARCHICAL AND SELF ORGANIZATION MAP ALGORITHM

K-MEANS BASED DATA CLUSTERING	AGGLOMERATIVE HIERARCHICAL ALGORITHM	SELF ORGANIZATION MAP
Partitioning Based Method	**Based on hierarchical tree**	**Based on neural network**
Input: k, dataset, randomly chosen k centroids	**Input randomly dataset**	**Random input vector from training dataset**
Objective: Minimizing sum of squared distance	**Objective: Minimizing sum of squared distance**	**Multidimensional data is mapped by competitive and unsupervised learning**
Final clustering may converge to local optima	**Final clustering may converge to local optima.**	**Final clustering may converge to local optima.**
Time complexity: O(n*k*d*i) Where n= no. of data points k= no. of clusters d= dimension of data i= no. of iterations	**Time complexity: $O(n^2 lgn)$**	**Time complexity: $O(m^2*l)$ Where m= dimensional input vector l=no of weight vector**

IV CONCLUSIONS

We have presented a comprehensive survey of the k-mean, Hierarchical, SOM clustering methods and applications developed in the field of clustering analysis. In this dissertation novel clustering algorithms which utilizes graph-theoretic and statistical techniques are explore. After analyzing the result of testing the clustering algorithms using R tool and running them under biological factors and situation the following conclusion are obtained:

- The performance becomes lower as the number of clusters k becomes greater.
- The SOM technique shows more accuracy in classing most of the data items into their suitable cluster than other technique.
- All the clustering techniques show ambiguity in some (noisy) data when clustered.
- Hierarchical clustering and SOM algorithms showed good result when using small data sets.
- Using R programming tool for Clustering, statistical computing and graphic result are represented.
- R supports for clustering tasks is as extensive as its support for classification and regression statistical computing and graphics and it has more techniques for clustering.

It supports various clustering algorithms with different clustering packages execution which gives a platform for data mining research process.

V FUTURE WORK

In this paper "Analysis of clustering in biological data with R" meant to analyze clustering techniques in biological data using R programming tool. I was unable to find in my search any study that attempt to analyze clustering algorithms under investigation in R. As the future work of clustering biological data attempted according to different situation and environment, many researchers surveyed that using clustering techniques in biological data remain the most active research problem, both for biomedical sciences and for clustering research. In biological science there are many complex clustering problems which cannot be solved by using existing clustering techniques. Some of problems include various aspects, such as functional properties, DNA, 3D structures, and chemical properties. For clustering the most important and challenging task includes: - CRM/personalization, bioinformatics and security applications. For detailed these application more research are needed.

It will be necessary to continuously improve and adapt the developed software to the newly gained knowledge and standards to maintain the status of a valuable and flexible software package in functional genomics. Many possible improvements to near future include general features: - Automatic clusters coloring for cross-cluster analysis, Implication of user defined queries for each gene, execution of filter and significance tests, links to GenBank, etc extra graphical presentations.

REFERENCES

[1] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco, CA, 2006.

[2] Pang-Ning Tan, Michael Steinbach and Vipin Kumar. Introduction to Data Mining. Addison Wesley; US ed edition. May 12, 2005.

[4] Data Mining A Knowledge Discovery Approach Krzysztof J. Cios Witold Pedrycz Roman W. Swiniarski Lukasz A. Kurgan Springer

[5] Florin Gorunescu Data Mining Concepts,Models and Techniques Springer

[6] Data Mining with R: learning by case studies Luis Torgo

[7] Witten, I. H. (Ian H.) Data mining : practical machine learning tools and techniques / Ian H.Witten, Eibe Frank. – 2nd ed.

[8] Brazma, Alvis and Vilo, Jaak. Minireview: Gene expression data analysis. Federation of European Biochemical societies, June 2000.

[9] Herrero J., Valencia A. and Dopazo J. A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics.

[10] Data mining : multimedia, soft computing, and bioinformatics /Sushmita Mitra and Tinku Acharya. A John Wiley & Sons, Inc., Publication

[11] Advanced data mining technologies in bioinformatics / Hui-Hwang Hsu, editor.

[12] Bioinformatics: Databases and Systems edited by Stanley Letovsky Kluwer Academic publishers

[13] Data Mining Practical Machine Learning Tools and Techniques Third Edition
Ian H. Witten Eibe Frank Mark A. Hall

[14] Data clustering: algorithms and applications [edited by] Charu C. Aggarwal, Chandan K. Reddy. CRC Publication

[15] A Survey of Partitional and Hierarchical Clustering Algorithms Chandan K. Reddy

[16] Ramoni, M., Sebastiani, P., & Kohane, I. (2002). Cluster analysis of gene expression dynamics.

[17] Robert Gentleman Rafael A. Irizarry Vincent J. Carey Sandrine Dudoit Wolfgang Huber Editors Bioinformatics and Computational Biology Solutions Using R and Bioconductor, Springer

[18] R and Data Mining: Examples and Case Studies 1 Yanchang Zhao

[19] A Survey of Partitional and Hierarchical Clustering Algorithms Chandan K. Reddy , Bhanukiran Vinzamuris

[20] Clustering Algorithms and Applications Edited by Charu C. Aggarwal Chandan K. Reddy

[21] Data Mining Practical Machine Learning Tools and Techniques Third Edition Ian H. Witten Eibe Frank Mark A. Hall 2002.

[22] Advanced data mining technologies in bioinformatics / Hui-Hwang Hsu, editor 2009.

[23] J. Herrero, A. Valencia, and J. Dopazo, "A Hierarchical Unsupervised Growing Neural Network for Clustering Gene Expression Patterns," Bioinformatics, vol. 17, pp. 126-136, 2001.

[24] Zhexue Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. In Proc. SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997.

[25] Doug Fisher, Optimization and Simplification of Hierarchical Clusterings, KDD

[26] Vesanto J. Usisng SOM in Data Mining. Licentiate's thesis. Helsinki University of Technology, Department of Computer Science and Engineering. 2000.

[27] Frank, A. & Asuncion, A. UCI Machine Learning Repository (http:/ / archive. ics. uci. edu/ ml). Irvine, CA: University of California, School of Information and Computer Science 2010.