Computers & Operations Research 39 (2012) 3046–3061

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Review

Clustering of high throughput gene expression data

Harun Pirim [a,*], Burak Ekşioğlu [a], Andy D. Perkins [b], Çetin Yüceer [c]

[a] Department of Industrial and Systems Engineering, Mississippi State University, P.O. Box 9542, Mississippi State, MS 39762, United States
[b] Department of Computer Science and Engineering, Mississippi State University, United States
[c] Department of Forestry, Mississippi State University, United States

ARTICLE INFO

Available online 29 March 2012

Keywords:
Clustering
Bioinformatics
Gene expression data
High throughput data
Microarrays

ABSTRACT

High throughput biological data need to be processed, analyzed, and interpreted to address problems in life sciences. Bioinformatics, computational biology, and systems biology deal with biological problems using computational methods. Clustering is one of the methods used to gain insight into biological processes, particularly at the genomics level. Clearly, clustering can be used in many areas of biological data analysis. However, this paper presents a review of the current clustering algorithms designed especially for analyzing *gene expression* data. It is also intended to introduce one of the main problems in bioinformatics – clustering gene expression data – to the operations research community.

Contents

1. Introduction . . . 3046
2. Biological background . . . 3047
3. Problem definition and representations of gene expression data . . . 3047
   3.1. Quantification of relations . . . 3048
   3.2. Validation of the partitions . . . 3049
   3.3. Representation of expression data and molecular interactions . . . 3050
4. Algorithms used for clustering gene expression data . . . 3051
   4.1. Flat clustering algorithms . . . 3051
   4.2. Hierarchical clustering algorithms . . . 3052
      4.2.1. Level selection methods . . . 3052
   4.3. Graph-based clustering algorithms . . . 3053
   4.4. Optimization-based algorithms . . . 3054
      4.4.1. Metaheuristic clustering algorithms . . . 3056
   4.5. Other algorithms . . . 3056
   4.6. Choice of an algorithm . . . 3057
5. Conclusion and future research for the operations research community . . . 3058
Acknowledgments . . . 3058
Appendix AGlossary . . . 3058
References . . . 3059

1. Introduction

Clustering in biology has a history that goes back to Aristotle's attempt to classify living organisms [6]. Today, clustering genomic data stands out as an approach to deal with high-dimensional data produced by high throughput technologies such as *gene* expression *microarrays* [84]. Biological data were limited to DNA sequence data before the *genome* age in the 1980s [68]. Nowadays, terabytes of high throughput biological information are generated with the advent of new technologies, such as microarrays, *eQTL* mapping, and *next generation sequencing*. Now, a need for exploiting computational methods exists to analyze and process such amounts of data in depth and in different ways to address complex biological questions regarding gene functions, gene co-expression, protein–protein interactions (PPI), personalized drug design, systems level functional analysis of plants and

* Corresponding author. Tel.: +1 662 325 4226.
E-mail address: harunpirim@gmail.com (H. Pirim).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.03.008

animals, and organism–environment interaction. This fact has given birth to disciplines like bioinformatics, computational biology, and *systems biology*.

In physics, before mathematical models were incorporated, i.e., before Newton, the discipline was stamp collecting (i.e., descriptive). Incorporation of mathematical models changed physics into a predictive science. In a similar manner, incorporation of computation into biology is changing the discipline from being a descriptive science to a predictive science. One of the prediction methods used in biology to analyze the high throughput data is clustering. As a data mining method, clustering of gene expression data was well studied during the last decade. Clustering is also a well-known and studied problem in the operations research (OR) field. However, clustering of gene expression data is not extensively studied by the OR community, although data mining techniques have been used in market segmentation and facility location problems.

Certain aspects of biological theories can be modeled using OR tools. One of these aspects is that a small subset of genes are typically involved in a particular cellular process of interest, and a cellular process happens only in a subset of the *samples* [65]. Another aspect is that genes of the same pathway may be induced or suppressed simultaneously or sequentially upon receiving stimuli [149]. A third aspect is that most biologists assume an approximately *scale-free topology*, or a *small world property*, for graphs constructed from gene expression data [145]. Hence, one may say that genes with high *connectivity* are much fewer in number than genes with low connectivity [131]. Thus, this review discusses many diverse approaches and algorithms that currently exist for clustering of gene expression data from an OR perspective by introducing background in molecular biology, and presenting clustering approaches and techniques. The paper is organized as follows: Section 2 gives concise information about molecular biology and relevant disciplines; Section 3 provides a problem definition for clustering gene expression data as well as representations of expression data; Section 4 reviews recent algorithms used for clustering gene expression data; Section 5 suggests future research directions for the operations research community; and Appendix A provides the glossary that includes definitions of the italicized words and phrases throughout the text.

2. Biological background

The essential cellular molecules for a biological system to function and interact with its surrounding include DNA, RNA, proteins, and *metabolite*s, all of which are under physiological and environmental control. Many different interaction layers exist among these molecules such as PPI networks, i.e., interactomes, gene regulatory networks (GRNs), biochemical networks, and gene co-expression networks. A holistic picture of these interactions is being studied through systems biology.

Based on the central dogma of molecular biology, DNA transcribes into RNA, and RNA translates into proteins, some of which then serve as catalysts in the production of metabolites. A gene is expressed upon receiving the transcriptional signal. Genes have *activator*s and *repressor*s. Genes reveal no or low expression values without activators. Repressors block gene expression, even in the presence of activators. *Transcription factor*s (TFs) are activator or repressor proteins produced by genes. TFs bind to *regulatory site*s and turn them on to transcribe RNA or off. Genes may show cascade interactions. For example, the product of one gene may increase or decrease the transcription rate of the other, and this process may continue downstream including temporal or causal order of molecular events.

Fig. 1. A microarray chip produced by Affymetrix courtesy. (*source*: http://www.affymetrix.com/about_affymetrix/media/image-library.affx).

It is often preferred to analyze thousands of genes' dynamics together rather than one at a time. The DNA microarray (Fig. 1) has been one of the commonly used technologies to measure thousands of gene expressions simultaneously [84], and microarray data have been stored in public databases such as the Gene Expression Omnibus (GEO) for further analysis. For example, the Affymetrix GeneChip Mouse Genome 430 2.0 Array provide 45,000 probe sets to analyze expression levels of more than 39,000 transcripts. Its *Feature* size is 11 μM. Eleven probe pairs per sequence are used.

The data extracted from microarrays or a similar technology is analyzed using a *reverse engineering* approach. A simplified framework of reverse engineering methodology for modeling GRNs from gene expression data is shown in Fig. 2, which is adapted from Lee and Tzou [76]. However, it is a challenging task to infer about GRNs because expression data are high-dimensional, complex, and non-linear. Further complicating the inference are the facts that, dynamic relations exist among thousands of genes, expression data involve *noise*, and the sample-to-gene ratio is normally very small [147] because the array chips corresponding to samples are expensive. Co-expressed genes show coherent *expression pattern*s, indicating that they may have similar functions [84] or co-exist in a pathway. However, different external conditions may trigger a gene to be expressed similarly with different group of genes [84]. Genes with similar expression patterns are more likely to regulate each other or to be regulated by a parent gene [92]. Here, the problem of quantifying the relations between genes arises.

A powerful clustering approach as well as a predictive model may detect patterns or relationships in expression data [84]. However, a predictive model should be guided by biological facts, meaning that results of predictive models should be validated by biological knowledge. On the other hand, biological experiments should be guided by computational methods to make the best use of data and reduce experimental and time costs (Fig. 3). Online databases exist to facilitate *validation* of the results obtained from predictive models. Incorporation of the database knowledge to modeling GRNs is essential for more accurate results or for comparing the model to reality.

3. Problem definition and representations of gene expression data

Clustering generates individual groups of data called a *partition*, rather than assigning *objects* into already known groups as in *classification* [8]. A partition is defined as follows:

$P=\{c_1,c_2,\ldots,c_s\}$, where s is the number of clusters; $\sum_{i=1}^{s}|c_i|=n$, where n is the number of objects; and $|c_i|$ is the cardinality of *cluster i*.

$X=\{x_1,x_2,\ldots,x_n\}$ is the set of n objects and $Y=\{y_1,y_2,\ldots,y_n\}$ is the set of n patterns, where $y_i \in R^d$ and d is the number of samples.

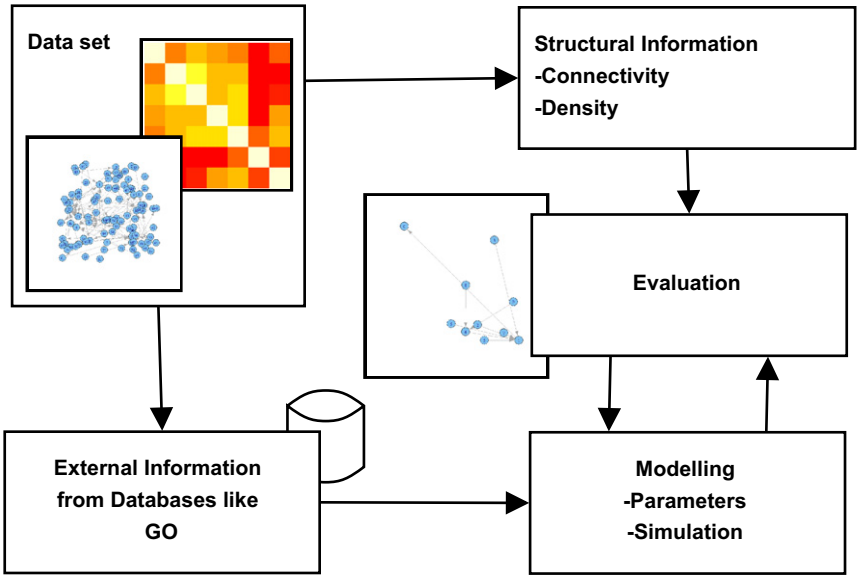

Fig. 2. Reverse engineering to infer about the extracted data.

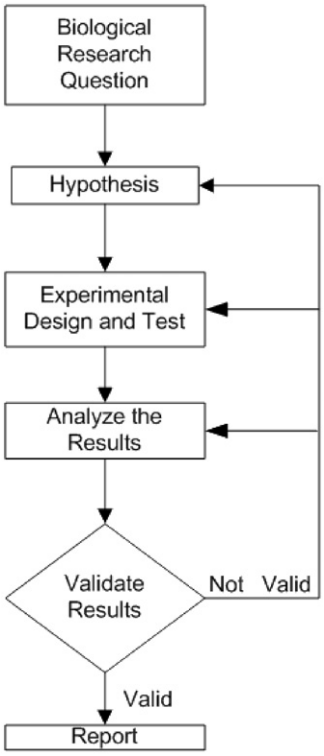

Fig. 3. Biological experiment and validation work flow.

The clustering problem is finding a partition that has clusters with objects having similar patterns.

There is no universally accepted definition of a cluster. However, objects in a cluster should be similar or coherent, and objects in different clusters should be dissimilar. In other words, similarity within a cluster should be maximized, and similarity between clusters should be minimized.

Clustering is often used in the gene expression data analysis which is an integrated process that comprises low-level and high-level analysis. Cluster analysis for gene expression data consists of three main steps: (1) pre-processing the data so that the clustering algorithm can use the data as an input; (2) using a clustering algorithm with an appropriate distance measure; and (3) using an index and/or a *biological database* to validate the quality of the clusters found. *Data pre-processing* is essential before clustering, since it affects clustering results. The effects of normalization and pre-clustering techniques have been demonstrated on clustering algorithms [120], so have the effects of filtering methods [127]. The distance measure can also affect the results of a clustering algorithm [57].

Although there are many problems associated with cluster analysis and there are many biological data types, this review mainly focuses on clustering algorithms as applied to microarray data unless otherwise mentioned. As an illustrative example we use a breast cancer microarray data set that is available at http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi. The data set is pre-processed [143], then 49 samples corresponding to 4 different collection of tumors consisting of 1213 genes each is used. The pre-processed expression image is shown in Fig. 4. Color densities and corresponding expression values are shown on the right vertical color bar of the figure. The samples are shown on the *y*-axis, while the genes are shown on the *x*-axis.

Since the real partition of the samples is known for the data in Fig. 4, clustering of samples is desired for the purpose of external validation. K-means (see Section 4.1) as applied in R base package is chosen for clustering. The *Eucledian distance* matrix between samples and the number of clusters, i.e., 4, are inputs to the K-means algorithm. The partition generated by K-means and the real partition are shown in Table 1. It should be noted that the order of the numbers identifying clusters of the real partition may not be the same in the generated partition. The last step of the cluster analysis is validation using the *C-rand* index. The *C-rand* value for this example is 0.343. This means that K-means could not find a partition very similar to the real one since the best *C-rand* value would be 1.

3.1. Quantification of relations

Distance measures are used for defining relationships between the biological molecules of interest. Clustering algorithms use this relationship in different ways. Hoeffding's D measure outperforms the others in quantifying non-linear associations when Pearson correlation, Spearman correlation, and Hoeffding's measure were compared for gene expression association analysis [40]. Bandyopadhyay and Pal [12] propose new distance measures

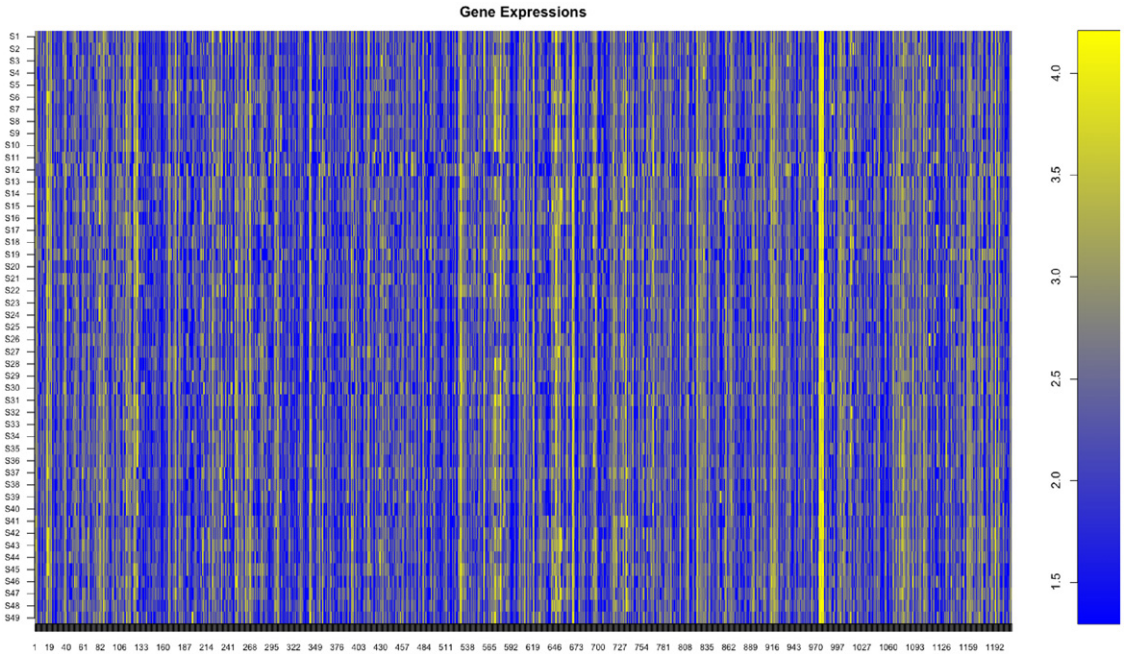

Fig. 4. Image plot of expression values.

based on Euclidean and *Manhattan distance* measures where *normalization* is dependent on the experiment type, i.e., samples. Balasubramaniyan et al. [9] also use a local, shape-based distance metric based on the Spearman rank correlation. The metric is used to identify local similar regions in gene expression profiles.

Pairwise relations between genes are often preferred for quantification, because it is computationally less costly than stochastic approaches where a relation is considered conditionally to other relations. Correlations or distance measures, e.g., Euclidean distances between gene pairs are calculated using the expression data, and then the resulting data matrix is used in a clustering algorithm to find the clusters of genes. However, use of direct distance measures between pairs of genes is somewhat traditional as opposed to transitive distance measures used between genes. Traditional use of a distance measure employs the "Guilt-by-Association" assumption that genes having similar expression values generally have similar functions and the genes with dissimilar expression values do not have similar functions [150]. The traditional approach is "Guilt-by-Association" because a biological function is often the result of many genes interacting with each other rather than a result of a simple pairwise relation [150]. However, transitive distance implies that there is at least one path, not necessarily of length 1 as in a pairwise relation, between two genes, and the length of this path is the distance between them. Researchers proposed that a transitive co-expression analysis applying a shortest path distance between two genes (Fig. 5) gives biologically meaningful results, rather than a direct pairwise distance measure [148,150]. Zhu et al. [150] use a hybrid distance matrix having both direct and shortest-path distances for clustering. Phan et al. [104] also use transitive directed acyclic graphs for representation of expression patterns. Once the data are clustered based on a distance measure, validation of the clustering algorithm's performance is essential.

3.2. Validation of the partitions

Before dealing with validation of the partitions generated by a clustering algorithm, there are sub-problems to consider: *filtering* mechanisms to be used for the data, algorithm to be used, the number of clusters to be chosen, distance metric to be used, cut-off height for the *dendrogram* of genes (in case a hierarchical clustering is used), approach to be used like agglomerative or divisive, validation methods, and measures for generated clusters. These are various aspects that will affect the validation results.

Outputs of clustering algorithms need validation to check whether or not the genes in the same clusters have biological relations. Clusters should make sense biologically, be reliable, not be formed by chance. The stability of a clustering algorithm, the validation of the generated cluster using *biological database*s, and the comparison with other algorithms are important aspects to measure reliability. Stability can be assessed by both sensitivity of the algorithm to the user-specified parameters and small modifications to the data sets [4].

There are mainly four different ways to validate the performance of a clustering algorithm: (1) visual validation: inspects if the algorithm detects a special structure of the data, e.g., number of clusters may be detected on 2D graphics. For example, the simulated data in Fig. 6 implies that the optimal number of clusters is two. (2) External validation: requires the knowledge of the real partition, e.g., *C-rand* or pre-defined structure of the data. (3) Internal validation: uses the features of the partition such as compactness, e.g., ensuring that variance within clusters are small and examining the separation of clusters, e.g., single linkage, average linkage, complete linkage. (4) Biological validation: uses biological annotations to see if the genes in clusters are enriched for biological terms significantly.

Each clustering validation technique has its own bias towards a given clustering criterion [36]. Ensemble and multi-objective clustering approaches [36] are used to address the problem of being biased towards a particular objective or a clustering criterion. A good clustering algorithm may or may not depend on prior knowledge or on user-defined parameters. Jiang et al. [65] propose that the algorithm should be able to extract useful information, detect the embedded and highly connected structure of gene expression data, and provide graphical representation of the cluster structure. Functions of some genes are published in relevant databases and genes with known similar functions may guide the clustering by being assigned to the same cluster. This partial knowledge can also be used as an input for a clustering

Table 1
Partition of samples S1–S49 into four clusters. (The numbers in the cells of the table indicate which cluster the sample belongs to. For example, S1, S2, S3 are all in the same cluster generated by the K-means algorithm and also based on the real partition.)

1–24	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24
K-means	4	4	4	2	1	1	2	2	2	2	3	3	1	1	1	1	4	4	3	4	4	2	1	1
Real	1	1	1	4	4	4	4	1	4	4	3	3	4	4	4	4	3	3	3	1	3	4	4	4
25–49	S25	S26	S27	S28	S29	S30	S31	S32	S33	S34	S35	S36	S37	S38	S39	S40	S41	S42	S43	S44	S45	S46	S47	S48	S49
K-means	4	1	1	1	2	3	2	4	4	2	2	2	1	4	4	4	1	1	1	1	1	2	1	1	2
Real	1	4	4	4	4	3	2	1	1	1	2	2	2	1	1	1	2	2	2	4	2	4	2	2	2

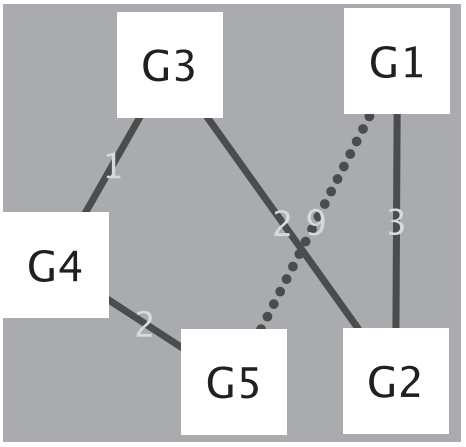

Fig. 5. Transitive distance – the distance between genes G1 and G5 – is 8 not 9 since the shortest path between genes is considered rather than the pairwise distance.

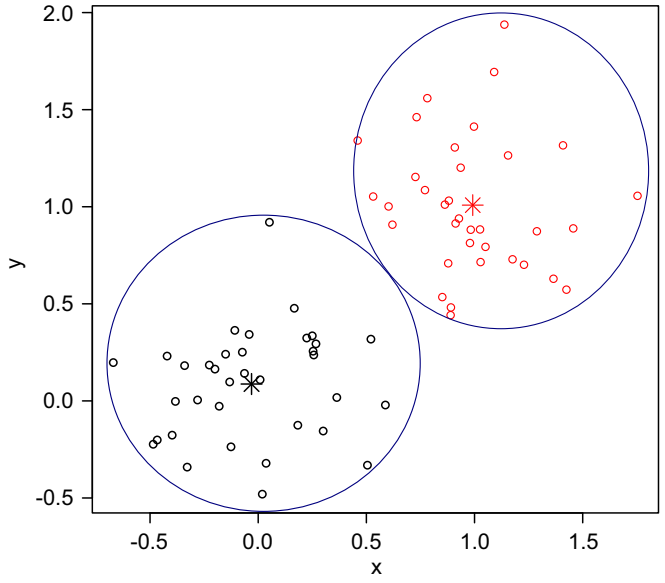

Fig. 6. 70 data points generated by two different normal distributions. Stars are the cluster centers to be used by the K-means algorithm. Circles represent the two clusters found by the K-means algorithm.

algorithm with the expectation that the resulting clusters will be more biologically meaningful [65]. For example, Cohen et al. [29] propose an algorithm that integrates semantic similarities from ontology structure to the procedure of getting clusters out of a dendrogram.

3.3. *Representation of expression data and molecular interactions*

Gene expression data is usually represented as an $n \times m$ matrix, where n is the number of genes and m is the number of time points or samples. Microarray features, or gene transcripts, are the rows of the expression matrix and are represented as vectors. Gene expression data sets are comprised of gene expression levels over time points, also called time course data (Table 2), or samples, such as control vs. treated. Clustering may be performed by grouping genes over samples or samples over genes. Since the number of genes is normally thousands and many of the genes have low or invariant expression values, filtering gene expression data to reduce the dimension of the $n \times m$ matrix is often necessary. Gene interactions

Table 2
A Sample Microarray Data [63]. The first column holds the names of the genes. For example, gene names starting with SID mean that they are not sequence verified. Other columns are the time series samples. The numbers in the cells of the table correspond to expression changes normalized to time zero. The changes are the ratios of the given time point expression levels to the serum-starved fibroblast expression levels. The data set is available at http://www.sciencemag.org/site/feature/data/984559.xhtml.

Gene name	0 h	15 min	30 min	1 h	2 h	4 h	6 h
EST W95908	1	0.72	0.1	0.57	1.08	0.66	0.39
SID487537 EST AA045003	1	1.58	1.05	1.15	1.22	0.54	0.73
SID486735	1	1.1	0.97	1	0.9	0.67	0.81
Genes	**Expression values**						
MAP kinase phosphatase-1	1	2.09	3.37	5.52	4.89	3.05	3.27
MAP kinase phosphatase-1	1	1.52	4.39	7.03	5.45	2.93	3.91
MAP kinase phosphatase-1	1	2.25	4.67	7.94	5.94	3.76	4.46

may be represented by graphs using an adjacency matrix. A graph G consists of vertices $V(G)$ that represent genes and edges $E(G)$ that represent relations between genes. Assuming a loopless, simple graph the adjacency matrix $A(G)$ has elements $a_{i,j}$ equal to 1 if i has relation with j, 0 otherwise. If the corresponding graph is not relational, i.e., binary, then a weight $w_{i,j}$ is associated with the edges showing the strength of the relation between i and j.

Clusters are generated by clustering algorithms that use a data representation as an input. The way the gene expression data is represented, whether it be a graph, matrix, or vector, may ease the computation for the problem on hand. For instance, a naive hierarchical clustering (HC) algorithm has time complexity of $O(n^3)$, however, the time complexity may be reduced to $O(n^2 \log n)$ using a *priority queue* data structure [85]. Representation of gene expression data as an $n \times m$ matrix or a graph may help a researcher focus on the genes of interest by making use of matrix theory and graph theory.

Visualization and computational representation of complex interactions between molecular components of a biological cell as graphs enables wide range of applications [118]. Models of GRNs fall between abstractness (like Boolean networks, or relevance networks) and concreteness (including biochemical interactions with stochastic kinetics [76]). Abstract models are scalable to large graphs but are further from reality, whereas concrete models are not scalable to large graphs but more accurately reflect biological reality. Hence, there is a trade-off between scalability and concreteness. Network models can be discrete or continuous. For example, deterministic or probabilistic Boolean networks and Bayesian networks have discrete variables whereas the neural network models and models based on differential equations use continuous variables. Abstract networks such as co-expression networks use edges from hypothetical inference, whereas concrete ones such as PPI use edges inferred from physical interactions [150]. Chen et al. [22] construct a graph for experimentally detected PPI. Nodes represent proteins and edges are the interactions with edge weights calculated based on a pre-defined formula.

There may be different relations between molecular components. For instance, components may interact with each other, one of the components may regulate the expression of the other or inhibit or stimulate the activity of the other [34]. All these relationships can be represented using graphs. Graph structures are typically used to suggest some biological questions about discovering potential drug targets. Graph topology reflects functional relationships and neighborhoods of genes [34]. Graph models are a very popular way of formalizing available knowledge of cellular systems in a consistent framework [15]. For instance, *factor graphs* are minimal graphs for inferring expression data [15]. Expression data may be integrated with transcription factor (TF) binding data to further infer interaction networks, and time course expression data may be integrated with physical interaction networks to identify pathways [15].

4. Algorithms used for clustering gene expression data

The algorithms used in clustering gene expression data can typically be grouped into two classes: partitional and hierarchical. However, clustering algorithms may also be grouped based on the representation of data, relationship between clusters, distribution of the data, and other properties. For example, some of the classes of algorithms include flat or partition-based clustering, hierarchical clustering, biclustering, model-based clustering, fuzzy clustering, optimization-based clustering, network-based clustering, and ensemble clustering. Of course, these groups may have intersections, and there may be hybrid approaches [24]. Clusters may be exhaustive, meaning that each object is assigned to a cluster, or non-exhaustive, meaning that some objects may be assigned to no cluster. Exclusive clusters are non-exhaustive ones to which an object is either assigned or not [85]. Objects are assigned solely to one cluster in hard clustering; whereas soft clusters, sometimes called overlapping clusters, may have common objects with non-negative value memberships. For different definitions of hard, soft, and partitional clustering see Manning et al. [85]. Different types of clustering algorithms are defined based on diverse features, such as representation of data and relation between clusters. The following subsections review the most recent and widely used methods. It is important to note that some clustering algorithms deal with gene expression while other algorithms cluster gene network. In our review we focus on those algorithms that deal with gene expression data. More specifically, the algorithms presented in Sections 4.1 and 4.2 typically deal with gene expression data represented in matrix form, those in Section 4.3 deal with gene expression data presented as graphs, and the algorithms presented in Section 4.4 typically deal with gene expression data represented in vector form.

EBSCO host and PubMed databases were investigated for obtaining the articles used in the review. However, the articles utilized were not limited to these databases. "Clustering method" along with "microarray data" or "gene expression data" was used as keywords in EBSCO host. The search resulted in 250 publications of which 29 were identified as relevant to clustering gene expression data. "Clustering of gene expression data" was used as the keyword in PubMed. The search resulted in 6706 publications of which a little over 100 were identified as relevant. The results were filtered based on being recent (i.e., after 2005) and having potential contribution to our review (i.e., being related to microarray analysis). More than 100 articles were used for the review. Since one of our objectives is to increase the interest of OR researchers, we provide more details on some classes of algorithms such as the optimization based ones.

4.1. Flat clustering algorithms

In flat clustering, objects are partitioned based on a (dis) similarity metric. K-means is perhaps the most widely used method. K-means is a randomized algorithm which generates cluster centers randomly and assigns objects to the nearest cluster center. The algorithm modifies the location of the centers to minimize the sum of squared distances between objects and their closest cluster centers. Richards et al. [107] reported that K-means performed faster and resulted in more biologically enriched clusters compared to three other methods. In that study K-means was used to cluster human brain expression data sets which had approximately 20,000 genes and 120 samples. Bohland

et al. [16] used K-means to cluster all left hemisphere brain voxels, a $25{,}155 \times 271$ matrix is used as an input for the algorithm. Sharma et al. [116] used a two-stage hyperplane algorithm applied in a software package called HPCluster. The first stage reduced the data size and the second stage was the conventional K-means. The algorithm handled 44,460 genes without a failure. Tseng and Wong [129] developed a clustering method which does not force all the genes into clusters. The method employs a truncation of the clustering tree first, and then applies the K-means algorithm to avoid K-means being trapped at a local minimum. The method was applied on both simulated and embryonic stem cell data. The authors provided a C library and a package to implement the method and visualize the data. Tseng [128] developed a K-means derivative, applying a penalty to avoid scattered objects being assigned into clusters and weights to incorporate prior information. The developed method is used for both mass spectrometry and microarray data sets.

K-means requires specification of the number of clusters before clusters are generated. K-means is also sensitive to noise (such as scattered objects) that is prevalent in gene expression data [65]. Furthermore, a partition generated by K-means may not be globally optimum since it relies on randomly chosen initial centers. Hence, K-means is sensitive to initial partitions, and it is applicable to data with only spherical-shaped clusters [136], which is not always the case for gene expression data. The time complexity of the K-means algorithm is $O(i\,k\,n\,m)$, where i is the number of iterations, k is the number of clusters, n is the number of objects and m is the dimension of an object [85].

Partitioning Around Medoids (PAM) is another widely used flat clustering algorithm [69]. PAM computes medoids for each cluster. PAM is computationally more costly than K-means since it requires pairwise distance calculations in each cluster. Wang et al. [132] used the system evolution principle of thermodynamics based on PAM to predict the number of clusters accurately. Huang and Pan [60] incorporated a gene's function knowledge into a new distance metric. Distances between genes with known similar function are shrunk to 0 before the genes are clustered using the PAM algorithm. Then, remaining genes are assigned to existing clusters and/or new clusters.

Self-Organizing Map (SOM), which is developed based on neural network methods is another flat clustering approach widely used in gene clustering. Ghouila et al. [43] employed a multi-level SOM-based clustering algorithm in the analysis of macrophage gene expression data. SOM, like K-means and PAM, requires the number of clusters and the grid structure of neurons as inputs. SOM maps high-dimensional data into 2D or 3D space. The potential of merging distinct patterns into a cluster can make SOM ineffective [65].

Knowing or predicting the number of clusters correctly for a flat clustering algorithm affects the quality of the clusters. Jonnalagadda and Srinivasan [66] developed a method to find the number of clusters in gene expression data. They evaluated different partitions from a clustering algorithm and identified the partition that describes the data best. They used an index that measures information transfer for additional clusters.

Clusters generated using a flat clustering algorithm do not exhibit any relations with each other, while clusters generated by a hierarchical clustering algorithm form a hierarchy.

4.2. Hierarchical clustering algorithms

Hierarchical clustering (HC) algorithms generate dendrograms that show relationships of objects and clusters as hierarchies (Fig. 7). HC algorithms can be divided into two groups: agglomerative and divisive. In agglomerative clustering, all the objects begin in individual clusters. Then, the object pair with the highest similarity is found and merged to be included in the same cluster. The objects then merge, or agglomerate iteratively, until only one cluster exists which includes all the objects. The merging process can be stopped at any time with a stopping criterion. A complete run of an agglomerative clustering algorithm produces a complete graph where each node has relations with all other nodes and a dendrogram where relationships between all objects appear.

Divisive HC methods work contrary to agglomerative HC methods. Divisive clustering methods iteratively divide the complete graph into smaller components by finding the pair of objects that have the lowest similarity and removing the edges between them. Divisive clustering can be represented by a dendrogram that gives smaller components at each successive split. The dendrogram's branches are the clusters. These branches also give information about similarity between clusters.

4.2.1. Level selection methods

One challenge encountered in HC is selection of the level which is used to cut the dendrogram through a number of branches corresponding to the number of clusters. Wild and Blankley [135] tested nine cluster level selection methods based on their lack of parametrization and simplicity. Neither of these methods outperform the others consistently on all data sets used. Kelley et al. [70] presented an automated method for cut-off level selection to avoid the dangers of using a fixed valued cut-off.

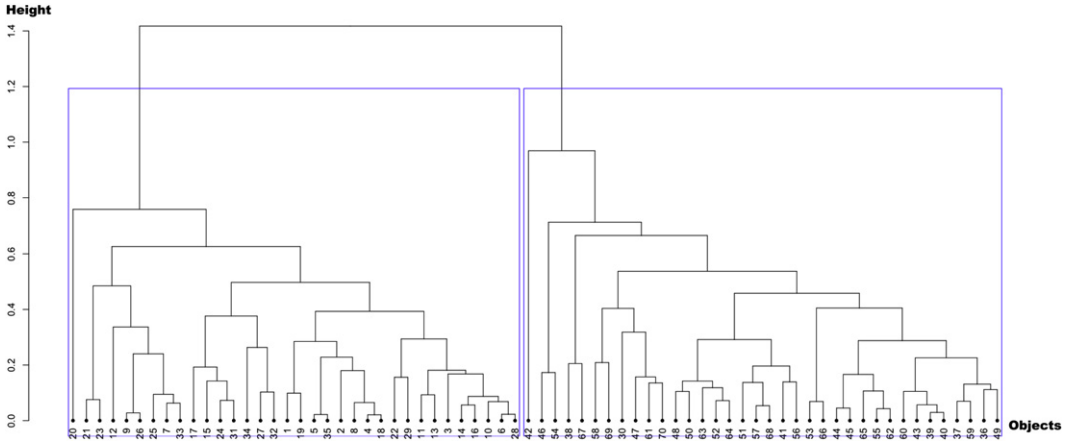

Fig. 7. Dendrogram of the simulated data generated for Fig. 6.

Langfelder et al. [73] proposed an algorithm that defines clusters from a hierarchical tree. However, they overcome the inflexibility of the fixed-height cut-off choice of the dendrogram. Their algorithm adapts to the shape of the dendrogram, is capable of detecting nested clusters, and can combine the advantages of hierarchical clustering and PAM. However, identification of optimal cutting parameters and estimation of number of clusters in the data set are still open research questions. They applied the algorithm on both human gene expression and simulated data. Although the algorithm has many user-defined parameters, it is reported that it works well with default settings compared to PAM and normal HC.

Liang and Wang [79] proposed a dynamic agglomerative clustering method and applied this on leukemia and avian pineal gland gene expression data. Based on the numerical results, the proposed method was convenient for data sets with or without noise, which is defined as scattered, singleton or mini-cluster genes. The method collected scattered genes in a cluster and grouped other clusters dynamically and agglomeratively.

HC algorithms are not robust to noise, and they have high computational complexity [65] which is $O(n^3)$ [85], where n is the number of objects. They are "greedy," meaning they combine the most similar two objects at the first step, and the following steps are affected by the initial step.

K-means and HC algorithms are root algorithms upon which many new algorithms are built. In choosing a clustering algorithm for an application, one should look at root clustering approaches and the desired features required for the application [126,124].

4.3. Graph-based clustering algorithms

Some HC algorithms make use of data represented as graphs. However, graph-based clustering algorithms are not all hierarchical. As mentioned earlier, biological data may be represented using graphs. For example, gene expression data may be regarded as a complete graph where the genes are the nodes of the network, and pairwise correlation values obtained from expression data are the edge weights of the node pairs. Hence, clustering in this case becomes a graph partitioning problem. Algebraic graph theory may be employed for the purpose of clustering. One algebraic graph theory tool is spectral clustering, a form of graph partitioning where the eigenvalues and eigenvectors in the Laplacian matrix, the difference between the adjacency and degree matrices, are usually used to reduce the dimension of the similarity matrix. The new matrix with reduced dimensions is used as an input for K-means or another algorithm [71]. Higham et al. [55] developed a class of spectral clustering algorithms. They tested the performance of the spectral algorithms on three different microarray data sets involving different types of diseases. Higham and Kalna [54] presented spectral analysis of *two-signed microarray expression data*. The time complexity of a general spectral clustering algorithm is $O(n^3)$ because of the eigenvalue computations.

Clustering based on each node's neighbors is also widely used for gene expression data. Huttenhower et al. [62] proposed a graph-based clustering algorithm called nearest neighbor networks (NNN). This algorithm first generates a directed graph with each gene connected to a specified number of nearest genes. Then, the graph is converted to an undirected one by keeping only the genes having a bidirectional relationship. Overlapping cliques of a specified size are merged to produce preliminary networks. Then, the preliminary networks containing cut-vertices are split, keeping the copies of the cut-vertices. They also introduced a software implementation of the algorithm proposed. Mete et al. [91] proposed an algorithm to find functional modules from large biological networks. The algorithm assigns nodes to the same cluster based on how they share common neighbors. Using three steps, the algorithm detects clusters, *hub*s, and outliers of the network. The first step checks every vertex for being core, i.e., a node having a predefined number of neighbors. If it is a core vertex, a new cluster is expanded. Otherwise, the vertex is labeled as a non-member. In the second step, the algorithm checks structure-reachable vertices, a specified similarity measure between vertices, from a core vertex. The third step classifies non-member vertices as hubs or outliers depending on whether or not the isolated vertices have edges connecting to two or more clusters. The worst case running time of the algorithm is $O(n^2)$, however, it reduces to $O(n)$ if the graph is random (i.e., edges of the graph are generated randomly).

Using minimum spanning trees of a graph to cluster gene expression data is practical since edge removal divides one group of genes into two groups directly. Xu et al. [138] represented gene expression data as a minimum spanning tree (MST). Clusters are then found by three algorithms that use different objective functions to generate subtrees. One objective is partitioning the tree into a specific number of subtrees and minimizing the total edge distances of all subtrees. The second objective is to minimize the distance between the center of each cluster and its objects. The third objective is similar to the second, except that a representative point is used instead of a center. The study reported that not much information is lost using a tree representation of the data sets. They also proposed a number of clustering algorithms for MST which were implemented as a software (available from authors). Two of the algorithms guarantee global optimality for non-trivial objective functions.

Community structure finding algorithms use graph structures and attempt to optimize a measure called *modularity* [101]. Higher modularity values are desired. Community structure finding consists of dividing the graph into groups according to certain structural information, like *betweenness* of edges, rather than similarity information normally used in traditional clustering approaches. In Newman and Girvan [101] and Girvan and Newman [44], the edges responsible for connecting many pairs of vertices, not the edges having the lower weights, are removed to find communities. With this technique, one can count how many paths proceed along each edge with the expectation that this number will be largest for intercommunity edges. The simplest example of the betweenness measure is based on shortest paths. Communities are sub-graphs where the edges within have high density connections but the edges between have low density connections. Communities appear to have a hierarchical structure in most real world contexts [27]. For instance, people make up departments and departments make up a university, just like words make up sentences, sentences make up chapters, and chapters compose books. In that sense, community finding is similar to an HC approach. HC is equivalent to starting with the network of interest, attempting to find the least similar connected pairs of vertices, and removing the edges between them iteratively.

Newman [100] expressed modularity in terms of eigenvectors of the modularity matrix of the network and proposed an algorithm which has a running time of $O(n^2 \log n)$ to divide the network into clusters. Ruan and Zhang [110] introduced a heuristic that combines spectral graph partitioning and local search to optimize modularity and a recursive algorithm to deal with the resolution problem that refers to being unable to find clusters smaller than a scale in network community detection. Ruan and Zhang [110]'s algorithm has a higher weighted matching score for protein community complex than that of Newman [100]. The algorithm is also faster than [100] for networks having more than about 1500 vertices. Clauset et al. [28] presented a fast hierarchical agglomerative algorithm to detect community structures in very large networks. The algorithm has a time complexity of $O(m\,d \log n)$, where m is the number of edges, n is the number of vertices and d is the depth of the dendrogram. Schwarz et al. [114]

used this algorithm to resolve functional organization in the rat brain. Newman [97] introduced a method of mapping weighted graphs to unweighted multigraphs, or graphs with multiple edges, to be able to use community structure finding algorithms [101] for weighted graphs. Gómez et al. [46] presented a reformulation of modularity to be able to work on weighted, directed, looped graphs defined from correlated data. It is also mentioned that other methods such as clique percolation [103] may be employed for a similar task with a relevant adaptation. The clique percolation method was used to find overlapping communities in yeast protein interaction data. Zahoránszky et al. [144] presented a new cluster selection method designed especially for extracting clusters from different partitions generated by the clique percolation method. The method does not require a similarity measure and is suitable for data with a graph representation. It relies on cohesive clusters in which all pairs of objects are similar to each other. Stone and Ayroles [119] proposed an algorithm to maximize modularity that modulates weights of the edges of biological data, represented as a graph. The algorithm is applied on human and *Drosophila melanogaster* data, compared with an agglomerative HC and three spectral clustering algorithms using 10,000 simulated data sets. The proposed method (for which the MATLAB code is freely available) has the highest percentage of correctly clustered objects and correctly separated objects for a specified number of clusters.

Label propagation is a recently developed method for finding community structure. It defines a community as a set of nodes such that each node has at least as many neighbors in its own community as in any other one. In the initial stage of the method, all nodes form a distinct community where each node has its own label. Then, at each step, the nodes join with that community to which the largest fraction of their neighbors belong by adopting the corresponding label. If there are multiple choices, a random decision is made with uniform distribution [125].

Newman [98] and Fortunato [39] provide more detailed reviews on algorithmic methods to detect community structure in networks. There are other graph-based clustering approaches [58,17]. To ease the use of graphs in solving problems, libraries such as The Boost Graph Library (BGL) for C++ and igraph [30] have been developed. The igraph library can be embedded into higher level programs or programming languages like C/C++, Python, and R. NetworkX [49] is a Python-based package for complex network research. Cfinder [1], which is an implementation of the clique percolation method [103], is used for community structure finding. There are visualization and exploratory tools for gene clusters to be interpreted more easily. Cytoscape and the gcExplorer [113,112] package for R programming language are designed for such a purpose. Fig. 8 illustrates two different layouts for an expression data generated by Cytoscape.

4.4. Optimization-based algorithms

Optimization-based algorithms may be more attractive to the OR community since optimization is at the heart of OR. Glover and Kochenberger [45] proposed a new modeling and solution methodology for clustering that can be used for finding groups, or modules, in genomic data. Modules can be regarded as cliques of similar objects. They model the clique partitioning (CP) over

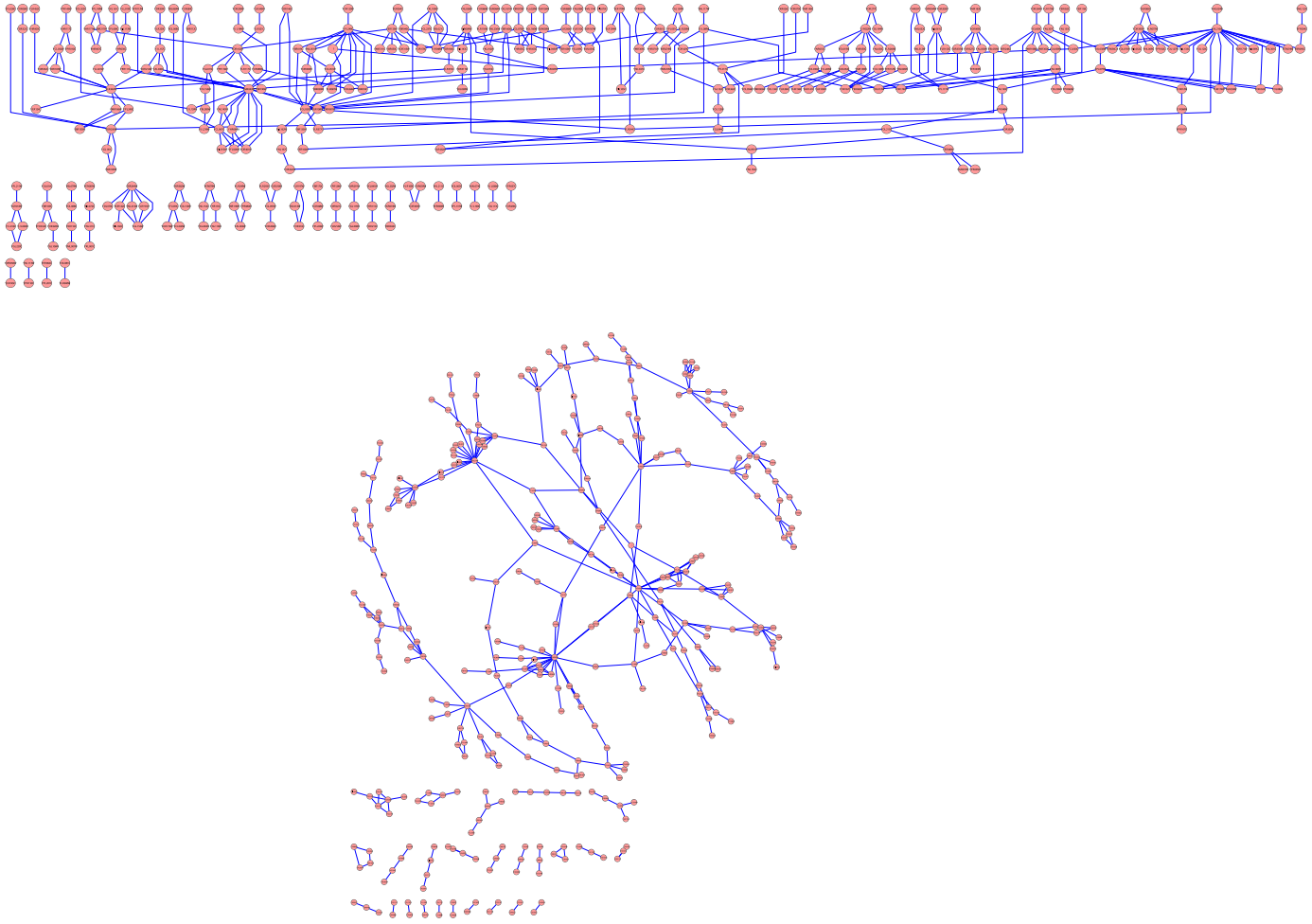

Fig. 8. Hierarchical and spring embedded layouts for protein–protein and protein–DNA interactions in yeast galactose metabolism.

nodes formulated as in (F2), rather than over edges as in (F1)

$$\text{(F1)}\quad \text{Maximize} \sum_{(i,j)\in E} w_{ij}x_{ij}$$

subject to

$$x_{ij}+x_{ir}-x_{jr}\le 1 \quad \forall i,j,r\in V, \quad i\ne j\ne r, \tag{1}$$

$$x_{ij}\in\{0,1\} \quad \forall i,j\in V. \tag{2}$$

$$\text{(F2)}\quad \text{Maximize} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_{ij}\sum_{k=1}^{K_{max}} x_{ik}x_{jk}$$

subject to

$$\sum_{k=1}^{K_{max}} x_{ik}=1 \quad \forall i\in V, \tag{3}$$

$$x_{ik}\in\{0,1\} \quad \forall i\in V, \quad k=1,\ldots,K_{max}. \tag{4}$$

In the first formulation (F1), x_{ij} is equal to 1 if the edge (i,j) is in the partition; 0 otherwise. The w_{ij} coefficient is the unrestricted weight of an edge between nodes i and j. E and V represent the set of edges and the set of vertices, respectively. In the second formulation (F2), x_{ik} is equal to 1 if node i is assigned to clique k. K_{max} is the maximum number of cliques or clusters allowed, n is the number of nodes, and w_{ij} is defined as in formulation (F1). Formulation (F2) has fewer variables and constraints, compared to (F1). Although (F2) is a quadratic model, it can be used for large instances of the CP problem. This model is similar to the one in Nascimento et al. [96] except that Glover and Kochenberger [45] used a maximization objective.

Nascimento et al. [96] used a greedy randomized adaptive search procedure (GRASP) for clustering different data sets of microarrays which was guided by an integer programming model similar to (F2).

Clustering based on the modularity measure introduced in Section 4.3 uses heuristic algorithms. Maximizing the modularity measure is also used as an objective function of the integer linear program (ILP) in Brandes et al. [18] as follows:

$$\text{Maximize}\frac{1}{2m}\sum_{(i,j\in V)}\left(E_{ij}-\frac{deg(i)deg(j)}{2m}\right)x_{ij}$$

subject to

$$x_{ii}=1 \quad \forall i\in V, \tag{5}$$

$$x_{ij}=x_{ji} \quad \forall i,j\in V, \tag{6}$$

$$x_{ij}+x_{jk}-2x_{ik}\le 1 \quad \forall i,j,k\in V, \tag{7}$$

$$x_{ik}+x_{ij}-2x_{jk}\le 1 \quad \forall i,j,k\in V, \tag{8}$$

$$x_{jk}+x_{ik}-2x_{ij}\le 1 \quad \forall i,j,k\in V, \tag{9}$$

$$x_{ij}\in\{0,1\} \quad \forall i,j. \tag{10}$$

The decision variables x_{ij} are defined as 1 if nodes i and j are assigned to the same cluster, 0 otherwise. E_{ij} is 1 if there is an edge between nodes i and j, 0 otherwise; $deg(i)$ and $deg(j)$ are the degrees of nodes i and j; m is the total number of edges. The equalities in (5) are the reflectivity constraints, (6) shows the symmetry constraints, (7)–(9) are the transitivity constraints, and (10) provides the binary constraints. The number of variables can be reduced to $\binom{n}{2}$, and the number of constraints can be reduced to $\binom{n}{3}$ by eliminating redundant variables and constraints where n is the number of nodes. Agarwal and Kempe [2] used the same ILP model with a different variable definition. To solve their model, they used a local search proposed by Newman [99] and a linear programming (LP) rounding algorithm to find upper bounds. Chen et al. [23] used LP to study the community structure of networks.

Lee et al. [75] proposed a graph-based optimization approach. They modeled clustering as a quadratic program. Their method automatically determines data distributions without a priori knowledge about the data that makes it superior to spectral clustering approaches.

Tan et al. [121] proposed a novel clustering approach based on a mixed integer non-linear program (MINLP). They converted their model to a mixed integer linear program (MILP) by introducing new variables and constraints. They applied a generalized Benders' Decomposition method to obtain lower and upper bounds for the solution of MILP to converge to a global optimal solution for large data sets. Their formulation is as follows:

$$\text{Minimize}\sum_{i=1}^{n}\sum_{j=1}^{c}\sum_{k=1}^{s} w_{ij}(a_{ik}-z_{jk})^2$$

subject to

$$\sum_{j=1}^{c} w_{ij}=1, \quad \forall i, \tag{11}$$

$$w_{ij}\in\{0,1\} \quad \forall i,j \quad \text{and} \quad z_{jk}\in R \quad \forall j,k. \tag{12}$$

Here, a_{ik} is the measure of distance for gene i having k features; w_{ij} are the binary variables having value of 1 if gene i is in cluster j, 0 otherwise. This model is expanded as

$$\text{Minimize}\sum_{j=1}^{c} w_{ij}\sum_{i=1}^{n}\sum_{k=1}^{s} a_{ik}^2-\sum_{i=1}^{n}\sum_{j=1}^{c}\sum_{k=1}^{s} a_{ik}w_{ij}z_{jk} + \sum_{j=1}^{c}\sum_{k=1}^{s} z_{jk}\sum_{i=1}^{n} w_{ij}(z_{jk}-a_{ik}).$$

Since the vector distance sum of all genes within a cluster to the cluster center, z_{jk}, must be 0, the optimality condition (13) holds

$$\sum_{i=1}^{n} w_{ij}(z_{jk}-a_{ik})=0 \quad \forall j,\ \forall k. \tag{13}$$

Parameter $suit_{ij}$ is introduced to the model to restrict some genes for specific clusters. It takes a value of 1 only for the cluster in which a gene is allowed to be involved, but 0 for the other clusters. This parameter reduces the computational burden of the problem and the formulation becomes

$$\text{Minimize}\sum_{i=1}^{n}\sum_{k=1}^{s} a_{ik}^2-\sum_{i=1}^{n}\sum_{j=1}^{c}\sum_{k=1}^{s}(suit_{ij})(a_{ik}w_{ij}z_{jk})$$

subject to

$$(suit_{ij})\left(z_{jk}\sum_{i=1}^{n} w_{ij}-\sum_{i=1}^{n} a_{ik}w_{ij}\right)=0 \quad \forall j,k, \tag{14}$$

$$\sum_{j=1}^{c}(suit_{ij})w_{ij}=1 \quad \forall i, \tag{15}$$

$$1\le\sum_{i=1}^{n}(suit_{ij})w_{ij}\le n-c+1 \quad \forall j, \tag{16}$$

$$w_{ij}\in\{0,1\} \quad \forall i,j, \tag{17}$$

$$z_{jk}^{L}\le z_{jk}\le z_{jk}^{U} \quad \forall j,k. \tag{18}$$

Constraints (14) are the necessary optimality conditions; (15) assure that each gene belongs to exactly one cluster; (16) assure that each cluster has at least one gene but no more than $n-c+1$ genes. The lower and upper bounds for the continuous variable z_{jk}

are z^L_{jk} and z^U_{jk} as shown in (18). To convert this non-linear model to a linear model, new variables and constraints are added to the model

$$y_{ijk} = w_{ij} z_{jk}, \tag{19}$$

$$z_{jk} - z^U_{jk}(1-w_{ij}) \le y_{ijk} \le z_{jk} - z^L_{jk}(1-w_{ij}), \tag{20}$$

$$z^L_{jk} w_{ij} \le y_{ijk} \le z^U_{jk} w_{ij} \quad \forall i, \forall j, \forall k. \tag{21}$$

Tan et al. [122] applied an algorithm guided by the above model to three different microarray data sets. Hayashida et al. [51] proposed two graph theoretic approaches: (1) maximizing the number of genes covered by at most a constant number of *reporter genes*, which are used to report the expression level of a gene, and (2) minimizing the number of reporter genes to cover all the nodes of the directed network. McAllister et al. [88] presented a computational study to solve the distance-dependent rearrangement clustering problem by developing a MILP. They presented three models based on the relative ordering of the elements, assignment of the elements to a final position, and distance assignment between a pair of elements. They reported that their models can be used for discoveries at the molecular level. Dittrich et al. [32] attempted to solve the problem of finding biologically meaningful sub-networks from PPI data. They transformed the problem to a price-collecting Steiner tree (PCST) problem, where the total sum of the edge weights of the subtree and the profits associated with the nodes not in the subtree are minimized. They were able to solve large instances of the problem in a reasonable time to optimality by the ILP approach for the transformed problem.

4.4.1. Metaheuristic clustering algorithms

Metaheuristics and heuristics are algorithms that generate feasible solutions to hard problems. They are used when it is impossible or too time costly to find an optimal solution to a problem. Metaheuristics are generally used in partition-based clustering and are rarely used in HC [13]. Genetic algorithms (GA), ant colony optimization (ACO), Tabu Search (TS), and simulated annealing (SA) are some widely used metaheuristics.

GAs are population-based heuristics and the steps are inspired from biological phenomena. Bandyopadhyay et al. [11] used a two-stage GA to cluster one artificial and three real microarray data sets. They employed a variable string length genetic scheme and multi-objectivity. In the first stage of the algorithm, they used an iterated version of Fuzzy C-Means (FCM), which is fuzzy version of K-means to detect the number of clusters. They compared the algorithm to an HC, an SOM and a Chinese restaurant-based clustering (CRC) algorithm [105] using two cluster validation indexes: *adjusted rand index* [61], and *silhouette index* [109]. Korkmaz et al. [72] also employed a multi-objective GA. One of the objectives is minimizing the total variation within clusters, which is identical to K-means' objective. The other one is minimizing the number of clusters in a partition. Faceli et al. [36] presented a Pareto-based multi-objective GA where objectives to be optimized are validation indices. Pareto set, the set including the best partitions based on different objective functions, is used to ensemble the partition pairs to have a consensus partition. The method is applied to six microarray data sets. The method is computationally expensive, including the dissimilarity matrix calculations the complexity is $O(n^2\, d)$, where n is the number of objects, and d is the dimension of an object. The crossover algorithm is $O(nk^2)$, where k is the number of clusters in the consensus partition. Wei and Cheng [134] developed an entropy-based clustering method in which a GA is applied. The method uses an adaptive threshold for similarity between objects and a fitness function to calculate the clustering accuracy. It was compared to K-means, FCM, and an entropy-based fuzzy clustering method upon which the proposed algorithm was developed. Four data sets, one of which is breast cancer data, were used for comparison. Hageman et al. [50] presented a GA-based biclustering algorithm with a homogeneous clustering criterion and introduced a cluster stability criterion. The method is used for metabolomics data sets.

He and Hui [52] investigated ACO-based algorithms for clustering gene expression data. The proposed algorithm, Ant-C, consists of four phases: initialization, tour construction, pheromone update where ants leave trails on the ground to guide other ants, and cluster output. Ant-C generates a fully connected graph where each node is a gene and each edge has a similarity weight, or pheromone intensity. Average pheromone intensity is used as a threshold to break the linkage of the fully connected graph to form clusters. MSTs are used in case of a partially connected graph to break the linkage of the network. Pheromone intensities are used as weights of the spanning tree. After finding the MST, it is partitioned into subtrees that form the clusters. Robbins et al. [108] also used an ACO algorithm for the *feature selection problem* in gene expression data.

TS moves away from the trap of local optimality by using diversification strategies. Gungor and Unler [48] apply a TS strategy to K-harmonic means clustering to avoid being trapped at local minima. The method is tested on Iris data. SA [47,19] also uses a diversification strategy to avoid being trapped at local optima. There are many other heuristic clustering approaches for gene expression data. Particle swarm optimization (PSO) [147,80,33,64], GRASP [31], honey-bee mating [38], memetic algorithms [90], furthest-point-first heuristic [42] are some of them.

4.5. Other algorithms

Clustering approaches are not limited to the methods listed in the sections above. The following explain some of the clustering approaches which can be classified in one or more of the above sections, or in a different section.

Fuzzy clustering allows an object to be assigned to more than one cluster. The strength of each object's belonging to a cluster is defined by a membership function that has a value between 0 and 1. The summation of membership values for each gene over all clusters is 1 [21]. Ravi et al. [106] proposed two fuzzy algorithms, variants of FCM, based on a *threshold accepting* heuristic. The algorithms are compared with FCM using *E. coli*, Iris, and Thyroid data sets. The comparison is based upon the number of clusters and the optimal values of objective functions. Ceccarelli and Maratea [21] used a learning metric to improve FCM. The new FCM is used on Iris, breast cancer, rat, sporulation, and yeast data sets. It is compared with FCM using a modified *entropy* index where membership values are considered as probabilities, normalized and raised to the power p. Saha and Bandyopadhyay [111] proposed a GA-based fuzzy method having a computational complexity of $O(k\ n \log n\ p\ g)$, where k is an estimate for the number of clusters, p is the population size, and g is the number of generations. The method is compared to an information-based clustering algorithm using a yeast expression data set and validated using both a biological validation tool and silhouette index. Mukhopadhyay and Maulik [95] proposed improved FCM- and GA-based fuzzy clustering algorithms using a support vector machine (SVM). The method is tested on diverse microarray data sets using *C-rand* and silhouette indices. Alshalalfah and Alhajj [5] also use FCM with SVM on three different microarray data sets. Other fuzzy clustering algorithms are provided by Hilt et al. [56], Maulik and Mukhopadhyay [87], Bandyopadhyay and Bhattacharyya [10].

Biclustering, or subspace clustering, finds a subset of similarly expressed genes over a subset of samples. It simultaneously clusters both rows (i.e., genes) and columns (i.e., conditions or samples) of a gene expression matrix [92]. One justification to use biclustering is that microarray data has large number of genes, which may not be relevant to the features in which a researcher is interested, and these features mask the contribution of the relevant ones [92]. Another justification is that co-expressed genes under certain conditions behave mostly independently [31]. Li et al. [78] extended a generic biclustering approach incorporating overlapping capability. The method is convenient for finding genomes with high genetic exchange and various conserved gene arrangements. The time complexity of the algorithm is $O(m^3(n^2+\log^2 m))$, where m is the number of data points and n is the number of dimensions. Subspace clustering error, row clustering error, coverage, and discrepancy in the number of clusters are used for validation purpose. Christinat et al. [26] showed that using discrete data coupled with a heuristic on continuous data leads to biclusters which are biologically meaningful. Li et al. [77] presented a qualitative biclustering algorithm (the source code and the server version of the algorithm are available), where an expression data matrix is composed of integer values only. The algorithm is applied on *E. coli* and yeast data sets and compared with other biclustering algorithms using biological enrichment criteria. Cano et al. [20] present an intelligent system for clustering. The system employs three novel algorithms of which two are biclustering algorithms.

Shen et al. [117] proposed a joint latent variable model for integrative clustering called iCluster. iCluster is scalable to different data types and enables the opportunity for next generation sequencing, a new emerging technology alternative to microarrays. Ma and Chan [83] proposed an iterative approach to mine overlapping patterns in gene expression data. Their approach consists of two steps. First, initial clusters are generated using any clustering algorithm. Second, cluster memberships are reassigned by a pattern discovery technique. At the end, a gene stays in the same cluster, changes clusters, or is copied to another cluster. Shaik and Yeasin [115] presented a unified framework to find differentially expressed genes from microarray data. The framework consists of three modules: gene ranking, significance analysis of the genes, and validation. An adaptive subspace iteration algorithm is used for clustering in the first module. Subspace structure is identified by an optimization procedure.

Yip et al. [141] presented some search algorithms to find dense regions in categorized or dichotomized gene expression data. Meng et al. [89] introduced an enrichment, a validation based on biological knowledge or database, constrained time dependent clustering algorithm. The algorithm is specially designed for time course data and integrated with biological knowledge guidance. Nueda et al. [102] also presented three novel methodologies for functional assessment of time course microarray data. Ernst et al. [35] designed an algorithm specifically for clustering short time series expression data.

Model-based clustering algorithms [67,53,133,137,74] have an assumption that gene expression data follow a statistical distribution and try to recognize the distribution. Information-criterion based clustering algorithm [81], adaptive clustering [25], neural network [142], cluster ensemble [59], consensus clustering [93], and game theoretical applications [94,82] are some of the other diverse clustering approaches.

Table 3 presents a summary of the reviewed algorithm categories. The table includes only one example from each category. These examples were selected based on recency, availability of the algorithm, and the number of times they were cited.

4.6. Choice of an algorithm

One issue in choosing a clustering approach for gene expression data is its suitability for biological applications. Andreopoulos et al. [6] listed a general set of desired features that change based on application and data type used: scalability, robustness, order insensitivity, minimum user-specified input, mixed data types, arbitrary-shaped clusters, and point proportion admissibility. Scalability is concerned with time and memory requirements, which increase as the data set becomes larger. Robustness refer to ability to detect outliers. Order insensitivity means that clusters are not changed as the objects' orders change. Minimum user-specified input, as the name suggests, emphasizes a clustering algorithm's reliance on user-specified input as little as possible. Mixed data types and arbitrary-shaped clusters refer to allowing objects to have numerical descriptive attributes and an algorithm's ability to find arbitrarily shaped clusters. Point proportion admissibility means stability of the results when objects are duplicated and re-clustered.

Another issue in choosing a clustering approach is the ease with which its performance can be evaluated. Internal and external performance measures are developed for evaluation. Internal measures rely on the structure of the partition, whereas external measures use external information, such as the knowledge of the real clusters. Real clusters for samples are usually known in advance, since samples are the designed experiments or the time course data. Clusters of genes are not known in advance except for the well annotated genes. Thus, using external performance measures for algorithms that cluster genes is hard. After clustering genes, researchers validate the clusters from gene databases if specific knowledge about the genes is available. Modularity, as discussed in Section 4.3, is an internal measure that makes use of the graph's structure. Modularity is a strong measure in the sense that gene expression graphs exhibit some common structures. Silhouette [109] is another internal measure based on the compactness and separation of the clusters. *Adjusted rand index*, or *C-rand* [61], is an external measure of agreement between two different partitions, one of which is real. *C-rand* is

Table 3
Summary of Algorithm Classes (CRC is the Chinese Restaurant Cluster, ISA and memISA are biclustering algorithms, CAGED is an algorithm designed for time series data, g is the clique size, s is the significant profile size, and e is the number of edges).

Class	Algorithm	Compared with	Biological data sets used	Validation method	Complexity	Availability
Flat	Richards et al. [107]	CRC, ISA, MemISA	Brain expression ($\sim 20,000$ genes)	Biological	$O(i\ k\ n\ m)$	Software
Hierarchical	Langfelder et al. [73]	HC, PAM	Drosophila PPI	External, biological	$O(n^3)$	R package
Network	Huttenhower et al. [62]	Eight clustering algorithms	Yeast (~ 6000 genes)	Biological	$O(n^g)$	Java implementation
Optimization	Dittrich et al. [32]	A heuristic approach	Human PPI (~ 2500 proteins)	Biological	$O(e^2n+en^2 \log n)$	Software
Other	Ernst et al. [35]	K-means, CAGED	human (50 profiles)	Biological	s^4	Java implementation

applicable even if the partitions do not have the same partition size [139]. Using simulated data, clusters' stability on a partition [37], reproducibility of the clusters [41], statistical significance between clusters [146], and comparing clustering of a combination of conditions with remaining conditions [140] are other ways to test the performance of a gene clustering algorithm.

5. Conclusion and future research for the operations research community

Clustering is fundamentally an optimization problem [7]. The clustering problem has awakened more interest in the statistics and computer science disciplines than in the optimization community [123]. Hence, the OR community, with an optimization paradigm, may become involved in and contribute more to clustering problems in the bioinformatics, computational, and systems biology disciplines.

No clustering algorithm exists with the best performance for all clustering problems. This fact makes it necessary to use or design algorithms specialized for the task at hand. Algorithmic methods are challenged by the introduction of high throughput technologies [14]. Guiding any clustering method with biological theory regarding gene expression data is essential. Mathematical programming (MP) formalism offers flexibility to incorporate biological knowledge, and it is crucial to use algorithms guided by MP models for gene expression data analysis [7]. Hence, integer programming models taking into account the biological knowledge would be a promising research direction. Clustering of gene expression data as a data mining sub-problem includes challenges providing a relatively hot and fruitful arena for the OR community [45]. OR has been an underutilized resource in the research agenda popularized by graph theory [3]. Graph-based clustering problems may involve more OR researchers to contribute to the agenda.

Acknowledgments

We would like to thank the three anonymous reviewers who helped tremendously to improve the paper. This work was supported partially by the National Science Foundation under grant number NSF EPS-0903787 and by the Mississippi INBRE (P20RR016476) funded by the National Center for Research Resources, National Institutes of Health. Dr. Yüceer's work was also partially supported by the National Science Foundation (IOS-0845834).

Appendix A. Glossary

Activator: a metabolite that regulates genes by increasing the rate of transcription.

Adjusted rand index: see index.

Betweenness: number of shortest paths proceeding along an edge.

Biological database: database used for validating whether or not a clustering algorithm generates clusters that are biologically meaningful. Gene Ontology (GO) is one of the widely used biological databases.

Classification: a supervised learning technique assigning objects into groups already known.

Cluster: a group that includes objects with similar attributes. Clustering is an unsupervised learning technique. Output of clustering is a set of clusters including similar objects, i.e., genes. Clustering is also an exploratory technique for network decomposition [76]. Clustering gathers objects into the same group based on a cluster definition or criterion.

Clustering: see cluster.

Connectivity: minimum set of genes required to inhibit the synthesis of a product.

C-rand: see index.

Data pre-processing: a process applied to raw gene expression data obtained from microarray experiment. Pre-processing includes quality assessment, filtering, and normalization also referred to as low-level analysis.

Dendrogram: a tree showing the hierarchical relations between groups of objects. Level of a dendrogram is the cut-off value to cut the dendrogram to obtain clusters.

Distance measure: a measure of the relationship between a pair of objects. Eucledian (e_{ab}), Manhattan (m_{ab}), Minkovski (mn_{ab}) are some examples. Correlation (c_{ab}) is also a widely used distance measure. However, $\sqrt{1-c_{ab}}$ approximation is used to satisfy the triangle inequality attribute of a metric. $e_{ab}=\sqrt{\sum_{i=1}^{n}(d_{ai}-d_{bi})^2}$, $m_{ab}=\sum_{i=1}^{n}(d_{ai}-d_{bi})$, $mn_{ab}=\sqrt[p]{\sum_{i=1}^{n}(d_{ai}-d_{bi})^p}$, where d_{ai} and d_{bi} are the values of the dimension i for objects a and b.

Entropy index: see index.

Eucledian distance: see distance measure.

eQTL: expression quantitative trait loci, genomic locations where genotype affects gene expression.

Expression pattern: pattern that a gene exhibits through different conditions, samples.

Factor graph: spanning sub-graph of a graph.

Feature: attribute of a microarray either referring to a spot of it or a gene.

Feature selection problem: selection of the most important, relevant genes for further analysis to reduce the dimension of high-dimensional data.

Filtering: removing the genes that do not exhibit significant expression change through conditions.

Gene: a functional unit of DNA with coded information. Reporter genes encode fluorescent proteins by which the expression level of gene can be observed [51]. The study of genes is called genomics. Genome refers to all of the fundamental genetic units, hereditary information, in a biological cell.

Gene expression: transcription of DNA into RNA.

Genome: see gene.

Genomics: see gene.

Hub: gene with high connectivity.

Index: measure for validating the performance of a clustering algorithm. *Adjusted rand index* for partitions P_1 and P_2 (*C-rand*(P_1,P_2)), as an external validation index, is one of the most widely used index for comparing the partition generated by a clustering algorithm with the real partition. Silhouette index for partition P_1 ($S(P_1)$), as an internal validation index, is used when the real partition of a biological data is not known. Partition entropy index (*PE*) is a measure of asymmetry. *C-rand*(P_1,P_2), $S(P_1)$ and *PE* formulations are

$$C\text{-}rand(P_1,P_2)=\frac{\sum_{i,j}\binom{n_{i,j}}{2}-[\sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}]/\binom{n}{2}}{1/2[\sum_i\binom{n_{i.}}{2}+\sum_j\binom{n_{.j}}{2}]-[\sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}]/\binom{n}{2}}$$

where $n_{i,j}$ is the number of objects at the intersection of clusters i and j, i is the cluster index for P_1, j is the cluster index for P_2; $n_{i.}$ and $n_{.j}$ are the numbers of objects in clusters i and j, respectively.

$$S(P_1)=\frac{\sum_{i=1}^{n}\dfrac{g(i)-a(i)}{\max(o(i),s(i))}}{n}$$

where n is the number of genes, $o(i)$ is the minimum of average distances from gene i to the genes in the other clusters, $s(i)$ is the

average distance from gene i to the remaining genes in the same cluster.

$PE = (1/n)\sum_i^n \sum_j^k \mu_{ij} \log_a \mu_{ij}$, where k is the number of clusters and μ_{ij} is the membership of gene i in cluster j [21].

Manhattan distance: see distance measure.

Metabolite: product of metabolism.

Microarray: a chip consisting of thousands of microscopic spots, i.e., features containing genes. Two signed microarray data includes both positive and negative values corresponding to up and down regulation, respectively.

miRNA: small RNA that binds to mRNA to regulate expression.

mRNA: the RNA transcribed by a gene to be translated into a protein [86].

Modularity: a measure of improvement on random connectivity (see Section 4.4 for a mathematical formula).

Next generation sequencing: a high throughput technology that allows measuring DNA sequences directly rather than indirectly. Image processing of microarrays is an indirect technology.

Noise: irregularities in the expression data. The sources of noise are sample preparation and hybridization process [130]. Genes that are irrelevant to clustering, i.e., non-informative genes [65] are also regarded as noise.

Normalization: transformation of raw expression data to ensure the comparability of gene expression levels across samples with the purpose of minimizing the systematic variations arising from technological issues [120].

Object: gene or sample.

Partition: the output of a clustering algorithm, the set of the clusters generated.

Priority queue: a heap data structure. A binary tree has a heap property if and only if it is empty or the key of the root has a higher value than all of its subtrees. The root node has the highest value and once it is extracted, regeneration of a single tree from two subtrees takes $O(\log n)$ time where n is the number of nodes. Heap tree is filled from left to right, once the root is deleted the left most leaf is taken as the root. Fig. 9 illustrates a tree with heap property: (a) when the root is extracted and then the first move is to bring the left most leaf to vacant root position. Next, the root value (i.e., 6) is swapped with left subtree's root value (i.e., 8) and the resulting new heap tree is shown in (b). The number of swaps is at most the depth of the complete binary tree which is $\log n$.

Quality assessment: a procedure to be applied on microarray data to ensure that the data is ready for further analysis.

Regulatory site: 5–15 base-pairs of genes.

Reporter gene: see gene.

Repressor: a protein that represses the transcription of genes.

Reverse engineering: also referred as deconvolution, process of analyzing biological data to infer about the interaction of biological components.

Sample: a microarray chip.

Scale-free topology: a graph topology where the degree distribution of nodes follows a power law.

Silhouette index: see index.

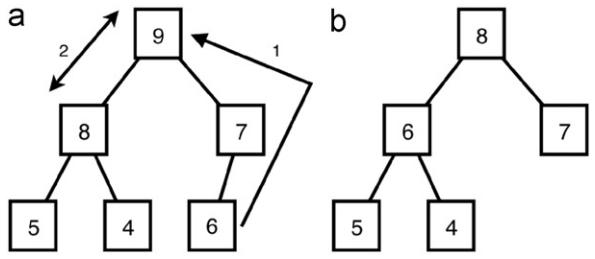

Fig. 9. Priority queue.

Small world property: a graph where each node has a small number of neighbors but can reach other nodes after a small number of steps.

Systems biology: a discipline that deals with the computational reconstruction of biological systems.

Transcription factor (*TF*): activator or repressor proteins produced by genes.

Threshold accepting: a local search strategy that allows up-hill moves for a minimization objective.

Two-signed microarray expression data: see microarray.

Validation: assessing the performance of a clustering algorithm either using performance indices or biologically.

References

[1] Adamcsek B, Palla G, Farkas IJ, Derényi I, Vicsek T. Cfinder: locating cliques and overlapping modules in biological networks. Bioinformatics 2006;22(8):1021–3.
[2] Agarwal G, Kempe D. Modularity-maximizing graph communities via mathematical programming. European Physical Journal B 2008;66:409–18.
[3] Alderson DL. Catching the network science bug: insight and opportunity for the operations researcher. Operations Research 2008;56(5):1047–65.
[4] Allison DB, Page GP, Beasley TM, Edwards JW. DNA microarrays and related genomics techniques: design, analysis, and interpretation of experiments (biostatistics). Chapman and Hall/CRC; 2005.
[5] Alshalalfah M, Alhajj R. Cancer class prediction: two stage clustering approach to identify informative genes. Intelligent Data Analysis 2009;13(4):671–86.
[6] Andreopoulos B, An A, Wang X, Schroeder M. A roadmap of clustering algorithms: finding a match for a biomedical application. Briefings in Bioinformatics 2009;10(3):297–314.
[7] Androulakis IP. Mathematical programming approaches for the analysis of microarray data. In: Handbook of optimization in medicine, vol. 26. Springer; 2009. p. 357–78.
[8] Asyali MH, Colak D, Demirkaya O, Inan MS. Gene expression profile classification: a review. Current Bioinformatics 2006;1:55–73.
[9] Balasubramaniyan R, Hüllermeier E, Weskamp N, Kämper J. Clustering of gene expression data using a local shape-based similarity measure. Bioinformatics 2005;21(7):1069–77.
[10] Bandyopadhyay S, Bhattacharyya M. Analyzing miRNA co-expression networks to explore TF-miRNA regulation. BMC Bioinformatics 2009;10(163):1–16.
[11] Bandyopadhyay S, Mukhopadhyay A, Maulik U. An improved algorithm for clustering gene expression data. Bioinformatics 2007;23(21):2859–65.
[12] Bandyopadhyay S, Pal SK. Dynamic range-based distance measure for microarray expressions and a fast gene-ordering algorithm. IEEE Transactions on Systems, Man, and Cybernetics 2007;37(3):742–9.
[13] Barthelemy P, Brucher F, Osswald C. Combinatorial optimisation and hierarchical classifications. Annals of Operations Research 2007;153(1):179–214.
[14] Berretta Regina, Mendes Alexandre, Moscato Pablo. Integer programming models and algorithms for molecular classification of cancer from microarray data. In: Estivill-Castro, editor. Proceedings of the Twenty-eighth Australasian conference on computer science, vols. 38, 27; 2005. p. 361–70.
[15] Beyer A. Network-based models in molecular biology. In: Ganguly Niloy, Deutsch Andreas, Mukherjee Animesh, Bellomo Nicola, editors. Dynamics on and of complex networks; 2009. p. 35–56.
[16] Bohland JW, Bokil H, Pathak SD, Lee CK, Ng L, Lau C, et al. Clustering of spatial gene expression patterns in the mouse brain and comparison with classical neuroanatomy. Methods 2010;50(2):105–12.
[17] Boscolo R, Sabatti C, Liao JC, Roychowdhury VP. A generalized framework for network component analysis. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2005;2(4):289–301.
[18] Brandes U, Delling D, Gaertler M, Gorke R, Hoefer M, Nikoloski Z, et al. On modularity clustering. IEEE Transactions on Knowledge and Data Engineering 2008;20(2):172–88.
[19] Bushel PR. Clustering of gene expression data and end-point measurements by simulated annealing. Journal of Bioinformatics and Computational Biology 2009;7(1):193–215.
[20] Cano C, Garcia F, Lopez FJ, Blanco A. Intelligent system for the analysis of microarray data using principal components and estimation of distribution algorithms. Expert Systems with Applications 2009;36(3):4654–63.
[21] Ceccarelli M, Maratea A. Improving fuzzy clustering of biological data by metric learning with side information. International Journal of Approximate Reasoning 2008;47(1):45–57.
[22] Chen J, Hsu W, Lee ML, Ng SK. Discovering reliable protein interactions from high-throughput experimental data using network topology. Artificial Intelligence in Medicine 2005;35:37–47.
[23] Chen WYC, Dress AWM, Yu WQ. Community structure of networks. Mathematics in Computer Science 2008;1(3):441–57.
[24] Chipman H, Tibshirani R. Hybrid hierarchical clustering with applications to microarray data. Biostatistics 2006;7(2):286–301.

[25] Chouakria AD, Diallo A, Giroud F. Adaptive clustering for time series: application for identifying cell cycle expressed genes. Computational Statistics and Data Analysis 2009;53(4):1414–26.
[26] Christinat Y, Wachmann B, Zhang L. Gene expression data analysis using a novel approach to biclustering combining discrete and continuous data. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2008;5(4):583–93.
[27] Clauset A, Moore C, Newman MEJ. Hierarchical structure and the prediction of missing links in networks. Nature 2008;453(7191):98–101.
[28] Clauset A, Newman MEJ, Moore C. Finding community structure in very large networks. Physical Review E 2004;70.
[29] Cohen DD, Kasif S, Melkman AA. Seeing the forest for the trees: using the gene ontology to restructure hierarchical clustering. Bioinformatics 2009;25(14):1789–95.
[30] Csardi G, Nepusz T. The igraph software package for complex network research. InterJournal Complex Systems 2006;1695.
[31] Dharan S, Nair AS. Biclustering of gene expression data using reactive greedy randomized adaptive search procedure. BMC Bioinformatics 2009;10(S27):1–10.
[32] Dittrich MT, Klau GW, Rosenwald A, Dandekar T, Müller T. Identifying functional modules in protein–protein interaction networks: an integrated exact approach. Bioinformatics 2008;24(13):223–31.
[33] Du Z, Wang Y, Ji Z. Pk-means: a new algorithm for gene clustering. Computational Biology and Chemistry 2008;32(4):243–7.
[34] Eckman BA, Brown PG. Graph data management for molecular and cell biology. IBM Journal of Research and Development 2006;50(6):545–60.
[35] Ernst J, Nau GJ, Joseph ZB. Clustering short time series gene expression data. Bioinformatics 2005;21(1):159–68.
[36] Faceli K, Souto MCPD, Araujo DSAD, Carvalhoüç ACPLFD. Multi-objective clustering ensemble for gene expression data analysis. Neurocomputing 2009;72:2763–74.
[37] Famili AF, Liu G, Liu Z. Evaluation and optimization of clustering in gene expression data analysis. Bioinformatics 2004;20(10):1535–45.
[38] Fathian M, Amiri B, Maroosi A. Application of honey-bee mating optimization algorithm on clustering. Applied Mathematics and Computation 2007;190(2):1502–13.
[39] Fortunato S. Community detection in graphs. Physics Reports 2010;486: 75–174.
[40] Fujita A, Sato JR, Demasi MAA, Sogayar MC. Comparing Pearson, Spearman and Hoeffding's D measure for gene expression association analysis. Journal of Bioinformatics and Computational Biology 2009;7(4):663–84.
[41] Garge NR, Page GP, Sprague AP, Gorman BS, Allison DB. Reproducible clusters from microarray research: whither? BMC Bioinformatics 2005;6(S10): 1–11.
[42] Geraci F, Leoncini M, Montangero M, Pellegrini M, Renda ME. K-boost: a scalable algorithm for high-quality clustering of microarray gene expression data. Journal of Computational Biology 2009;16(6):859–73.
[43] Ghouila A, Yahia SB, Malouche D, Jmel H, Laouini D, Guerfali FZ, et al. Application of multi-SOM clustering approach to macrophage gene expression analysis. Infection, Genetics and Evolution 2009;9(3):328–36.
[44] Girvan M, Newman MEJ. Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America 2002;99(12):7821–6.
[45] Glover FW, Kochenberger G. New optimization models for data mining. International Journal of Information Technology and Decision Making 2006;5(4):605–9.
[46] Gómez S, Jensen P, Arenas A. Analysis of community structure in networks of correlated data. Physical Review E 2009;80(016114):1–5.
[47] Gungor Z, Unler A. K-harmonic means data clustering with simulated annealing heuristic. Applied Mathematics and Computation 2007;184(2): 199–209.
[48] Gungor Z, Unler A. K-harmonic means data clustering with tabu-search method. Applied Mathematical Modelling 2008;32(6):1115–25.
[49] Hagberg AA, Schult DA, Swart PJ. Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the seventh python in science conference (SciPy2008), Pasadena, CA USA; August 2008. p. 11–5.
[50] Hageman JA, Berg RAVD, Westerhuis JA, Werf MJVD, Smilde AK. Genetic algorithm based two-mode clustering of metabolomics data. Metabolomics 2008;4(2):141–9.
[51] Hayashida M, Sun F, Aburatani S, Horimoto K, Akutsu T. Integer programming-based approach to allocation of reporter genes for cell array analysis. In: The first international symposium on optimization and systems biology (OSB07); 2007. p. 288–301.
[52] He Y, Hui SC. Exploring ant-based algorithms for gene expression data analysis. Artificial Intelligence in Medicine 2009;47(2):105–19.
[53] Heath JW, Fu MC, Jank W. New global optimization algorithms for model-based clustering. Computational Statistics and Data Analysis 2009;53(12):3999–4017.
[54] Higham DJ, Kalna G. Spectral analysis of two-signed microarray gene expression data. Mathematical Medicine and Biology 2007;24(2):131–48.
[55] Higham DJ, Kalna G, Kibble M. Spectral clustering and its use in bioinformatics. Journal of Computational and Applied Mathematics 2007;204: 25–37.
[56] Hilt SW, Yelundur A, McChesney C, Landry M. Support vector machine implementations for classification and clustering. BMC Bioinformatics 2006;7(4):1–18.
[57] Horst E. Distance measures for MPEG-7-based retrieval. In: MIR '03: proceedings of the fifth ACM SIGMM international workshop on multimedia information retrieval. New York, NY, USA: ACM; 2003. p. 130–7.
[58] Hu X, Ng M, Wu FX, Sokhansanj BA. Mining modeling, and evaluation of subnetworks from large biomolecular networks and its comparison study. IEEE Transactions on Information Technology in Biomedicine 2009;13(2): 184–94.
[59] Hu X, Park EK, Zhang X. Microarray gene cluster identification and annotation through cluster ensemble and EM-based informative textual summarization. IEEE Transactions on Information Technology in Biomedicine 2009;13(5):832–40.
[60] Huang D, Pan W. Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. Bioinformatics 2006;22(10):1259–68.
[61] Hubert L, Arabie P. Comparing partitions. Journal of Classification 1985;2: 193–218.
[62] Huttenhower C, Flamholz AI, Landis JN, Sahi S, Myers CL, Olszewski KL, et al. Nearest neighbor networks: clustering expression data based on gene neighborhoods. BMC Bioinformatics 2007;8(250):1–13.
[63] Iyer VR, Eisen MB, Ross DT, Schuler G, Moore T, Lee JC, et al. The transcriptional program in the response of human fibroblasts to serum. Science 1999;283(5398):83–7.
[64] Jarboui B, Cheikh M, Siarry P, Rebai A. Combinatorial particle swarm optimization (CPSO) for partitional clustering problem. Applied Mathematics and Computation 2007;192(2):337–45.
[65] Jiang D, Tang C, Zhang A. Cluster analysis for gene expression data: a survey. IEEE Transactions on Knowledge and Data Engineering 2004;16(11): 1370–86.
[66] Jonnalagadda S, Srinivasan R. NIFTI: an evolutionary approach for finding number of clusters in microarray data. BMC Bioinformatics 2009;10(40):1–13.
[67] Joshi A, Smet RD, Marchal K, Peer YVD, Michoel T. Module networks revisited: computational assessment and prioritization of model predictions. Bioinformatics 2009;25(4):490–6.
[68] Kanehisa M, Bork P. Bioinformatics in the post-sequence era. Nature Genetics 2003;33:305–10.
[69] Kaufman L, Rousseeuw P. Finding groups in data: an introduction to cluster analysis. Wiley and Sons; 1990.
[70] Kelley LA, Gardner SP, Sutcliffe MJ. An automated approach for clustering an ensemble of NMR-derived protein structures into conformationally related subfamilies. Protein Engineering 1996;9(11):1063–5.
[71] Kim J, Choi S. Semidefinite spectral clustering. Pattern Recognition 2006;39:2025–35.
[72] Korkmaz EE, Du J, Alhajj R, Barker K. Combining advantages of new chromosome representation scheme and multi-objective genetic algorithms for better clustering. Intelligent Data Analysis 2006;10(2): 163–82.
[73] Langfelder P, Zhang B, Horvath S. Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for R. Bioinformatics Applications Note 2008;24(5):719–20.
[74] Lau JW, Green PJ. Bayesian model-based clustering procedures. Journal of Computational and Graphical Statistic 2007;16(3):526–58.
[75] Lee CH, Zaiane OR, Park HH, Huang J, Greiner R. Clustering high dimensional data: a graph-based relaxed optimization approach. Information Sciences 2008;178(23):4501–11.
[76] Lee WP, Tzou WS. Computational methods for discovering gene networks from expression data. Briefings in Bioinformatics 2009;10(4):408–23.
[77] Li G, Ma Q, Tang H, Paterson AH, Xu Y. Qubic: a qualitative biclustering algorithm for analyses of gene expression data. Nucleic Acids Research 2009;37(15):1–10.
[78] Li J, Halgamuge SK, Tang SL. Genome classification by gene distribution: an overlapping subspace clustering approach. BMC Evolutionary Biology 2008;8(116):1–15.
[79] Liang F, Wang N. Dynamic agglomerative clustering of gene expression profiles. Pattern Recognition Letters 2007;28(9):1062–76.
[80] Liu J, Li Z, Hu X, Chen Y. Biclustering of microarray data with MOSPO based on crowding distance. BMC Bioinformatics 2009;10(S9):1–10.
[81] Liu T, Lin N, Shi N, Zhang B. Information criterion-based clustering with order-restricted candidate profiles in short time-course microarray experiments. BMC Bioinformatics 2009;10(146):1–20.
[82] Lucchetti R, Moretti S, Patrone F, Radrizzani P. The Shapley and Banzhaf values in microarray games. Computers and Operations Research 2009;2342.
[83] Ma PCH, Chan KCC. An iterative data mining approach for mining overlapping co-expression patterns in noisy gene expression data. IEEE Transactions on Nanobioscience 2009;8(3):252–8.
[84] Ma PCH, Chan KCC. A novel approach for discovering overlapping clusters in gene expression data. IEEE Transactions on Biomedical Engineering 2009;56(7):1803–8.
[85] Manning CD, Raghavan P, Schutze H. An introduction to information retrieval. Cambridge University Press; 2009 (Online Edition).
[86] Marketa Z, Jeremy OB. Understanding bioinformatics. Garland Science; 2008.
[87] Maulik U, Mukhopadhyay A. Simulated annealing based automatic fuzzy clustering combined with ANN classification for analyzing microarray data. Computers and Operations Research 2010;37(8):1369–80.

[88] McAllister SR, DiMaggio PA, Floudas CA. Mathematical modeling and efficient optimization methods for the distance-dependent rearrangement clustering problem. Journal of Global Optimization 2009;45(1):111–29.
[89] Meng J, Gao SJ, Huang Y. Enrichment constrained time-dependent clustering analysis for finding meaningful temporal transcription modules. Bioinformatics 2009;25(12):1521–7.
[90] Merz P. Analysis of gene expression profiles: an application of memetic algorithms to the minimum sum-of-squares clustering problem. BioSystems 2003;72(1–2):99–109.
[91] Mete M, Tang F, Xu X, Yuruk N. A structural approach for finding functional modules from large biological networks. BMC Bioinformatics 2008;9(S19): 1–14.
[92] Mitra S, Das R, Banka H, Mukhopadhyay S. Gene interaction – an evolutionary biclustering approach. Information Fusion 2009;10:242–9.
[93] Monti S, Tamayo P, Mesirov J, Golub T. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 2003;52(1–2):91–118.
[94] Moretti S. Statistical analysis of the Shapley value for microarray games. Computers and Operations Research 2009;2341.
[95] Mukhopadhyay A, Maulik U. Towards improving fuzzy clustering using support vector machine: application to gene expression data. Pattern Recognition 2009;42(11):2744–63.
[96] Nascimento MCV, Toledo FMB, Carvalho ACPLFD. Investigation of a grasp-based clustering algorithm applied to biological data. Computers and Operations Research 2010;37(8):1381–8.
[97] Newman MEJ. Analysis of weighted networks. Physical Review E 2004;70(056131):1–9.
[98] Newman MEJ. Detecting community structure in networks. European Physical Journal B 2004;38(2):321–30.
[99] Newman MEJ. Finding community structure in networks using the eigen vectors of matrices. Physical Review E 2006;74(036104).
[100] Newman MEJ. Modularity and community structure in networks. Proceedings of the National Academy of Sciences of the United States of America 2006;103(23):8577–82.
[101] Newman MEJ, Girvan M. Finding and evaluating community structure in networks. Physical Review E 2006;69(026113):1–15.
[102] Nueda MJ, Sebastián P, Tarazona S, García FG, Dopazo J, Ferrer A, et al. Functional assessment of time course microarray data. BMC Bioinformatics 2009;10(S9):1–18.
[103] Palla G, Derényi I, Farkas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. Nature 2005;435(9): 814–8.
[104] Phan V, George EO, Tran QT, Goodwin S. Analyzing microarray data with transitive directed acyclic graphs. Journal of Bioinformatics and Computational Biology 2009;7(1):135–56.
[105] Qin ZS. Clustering microarray gene expression data using weighted chinese restaurant process. Bioinformatics 2006;22(16):1988–97.
[106] Ravi V, Bin M, Kumar PR. Threshold accepting based fuzzy clustering algorithms. International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems 2006;14(5):617–32.
[107] Richards AL, Holmans P, O'Donovan MC, Owen MJ, Jones L. A comparison of four clustering methods for brain expression microarray data. BMC Bioinformatics 2008;9(490):1–17.
[108] Robbins KR, Zhang W, Bertrand JK. The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification. Mathematical Medicine and Biology 2007;24(4):413–26.
[109] Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 1987;20:53–65.
[110] Ruan J, Zhang W. Identifying network communities with a high resolution. Physical Review E 2008;77(016104):1–12.
[111] Saha S, Bandyopadhyay S. A new point symmetry based fuzzy genetic clustering technique for automatic evolution of clusters. Information Sciences 2009;179(19):3230–46.
[112] Scharl T, Leisch F. gcExplorer: interactive exploration of gene clusters. Bioinformatics 2009;25(8):1089–90.
[113] Scharl T, Voglhuber I, Leisch F. Exploratory and inferential analysis of gene cluster neighborhood graphs. BMC Bioinformatics 2009;10(288):1–14.
[114] Schwarz AJ, Gozzi A, Bifone A. Community structure in networks of functional connectivity: resolving functional organization in the rat brain with pharmacological MRI. NeuroImage 2009;47(1):302–11.
[115] Shaik ZS, Yeasin M. A unified framework for finding differentially expressed genes from microarray experiments. BMC Bioinformatics 2007;8(347): 1–21.
[116] Sharma A, Podolsky R, Zhao J, McIndoe RA. A modified hyperplane clustering algorithm allows for efficient and accurate clustering of extremely large datasets. Bioinformatics 2009;25(9):1152–7.
[117] Shen R, Olshen AB, Ladanyi M. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. Bioinformatics 2009;25(22):2906–12.
[118] Steggies LJ, Banks R, Shaw O, Wipat A. Qualitatively modeling and analyzing genetic regulatory networks: a Petri net approach. Bioinformatics 2007;23(3): 336–43.
[119] Stone EA, Ayroles JF. Modulated modularity clustering as an exploratory tool for functional genomic inference. PLoS Genetics 2009;5(5):1–13.
[120] Tan MP, Broach JR, Floudas CA. Evaluation of normalization and pre-clustering issues in a novel clustering approach: global optimum search with enhanced positioning. Journal of Bioinformatics and Computational Biology 2007;5(4):895–913.
[121] Tan MP, Broach JR, Floudas CA. A novel clustering approach and prediction of optimal number of clusters: global optimum search with enhanced positioning. Journal of Global Optimization 2007;39(3):323–46.
[122] Tan MP, Smith EN, Broach JR, Floudas CA. Microarray data mining: a novel optimization-based approach to uncover biologically coherent structures. BMC Bioinformatics 2008;9(268):1–21.
[123] Teboulle M. A unified continuous optimization framework for center-based clustering methods. Journal of Machine Learning Research 2007;8:65–102.
[124] Thalamuthu A, Mukhopadhyay I, Zheng X, Tseng GC. Evaluation and comparison of gene clustering methods in microarray analysis. Bioinformatics 2006;22(19):2405–12.
[125] Tibely G, Kertesz J. On the equivalence of the label propagation method of community detection and a Potts model approach. Physica A: Statistical Mechanics and its Applications 2008;387(19–20):4982–4.
[126] Torrente A, Kapushesky M, Brazma A. A new algorithm for comparing and visualizing relationships between hierarchical and at gene expression data clusterings. Bioinformatics 2005;21(21):3993–9.
[127] Tritchler D, Parkhomenko E, Beyene J. Filtering genes for cluster and network analysis. BMC Bioinformatics 2009;10(193):1–9.
[128] Tseng GC. Penalized and weighted k-means for clustering with scattered objects and prior information in high-throughput biological data. Bioinformatics 2007;23(17):2247–55.
[129] Tseng GC, Wong WH. Tight clustering: a resampling-based approach for identifying stable and tight patterns in data. Biometrics 2005;61(1):10–6.
[130] Tu Y, Stolovitzky G, Klein U. Quantitative noise analysis for gene expression microarray experiments. Proceedings of the National Academy of Sciences of the United States of America 2002;99(22):14031–6.
[131] Tyler AL, Asselbergs FW, Williams SM, Moore JH. Shadows of complexity: what biological networks reveal about epistasis and pleiotropy. BioEssays 2009;31(2):220–7.
[132] Wang K, Zheng J, Zhang J, Dong J. Estimating the number of clusters via system evolution for cluster analysis of gene expression data. IEEE Transactions on Information Technology in Biomedicine 2009;13(5):848–53.
[133] Wang S, Zhu J. Variable selection for model-based high-dimensional clustering and its application to microarray data. Biometrics 2008;64(2):440–8.
[134] Wei LY, Cheng CH. An entropy clustering analysis based on genetic algorithm. Journal of Intelligent and Fuzzy Systems 2008;19(4–5):235–41.
[135] Wild DJ, Blankley CJ. Comparison of 2d fingerprint types and hierarchy level selection methods for structural grouping using wards clustering. Journal of Chemical Information and Computer Sciences 2000;40:155–62.
[136] Wu FX. Genetic weighted k-means algorithm for clustering large-scale gene expression data. BMC Bioinformatics 2008;9(S12):1–10.
[137] Xie B, Pan W, Shen X. Variable selection in penalized model-based clustering via regularization on grouped parameters. Biometrics 2008;64(3): 921–30.
[138] Xu Y, Olman V, Xu D. Clustering gene expression data using graph-theoretic approach: an application of minimum spanning trees. Bioinformatics 2002;18(4):536–45.
[139] Yeung KY, Fraley C, Murua A, Raftery AE, Ruzzo WL. Model-based clustering and data transformations for gene expression data. Tech Report, UW-CSE; 2001.
[140] Yeung KY, Haynor DR, Ruzzo WL. Validating clustering for gene expression data. Bioinformatics 2001;17(4):309–18.
[141] Yip AM, Ng MK, Wu EH, Chan TF. Strategies for identifying statistically significant dense regions in microarray data. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2007;4(3):415–29.
[142] Yu Z, Wong HS. Class discovery from gene expression data based on perturbation and cluster ensemble. IEEE Transactions on Nanobioscience 2009;8(2):147–60.
[143] Yujin H, Philippe BJ, Pablo T, Golub TR, Mesirov JP. Subclass mapping: identifying common subtypes in independent disease data sets. PLoS One 2007;2(11).
[144] Zahoránszky LA, Katona GY, Hári P, Csizmadia AM, Zweig KA, Köhalmi GZ. Breaking the hierarchy – a new cluster selection mechanism for hierarchical clustering methods. Algorithms for Molecular Biology 2009;4(12):1–22.
[145] Zhang B, Horvath S. A general framework for weighted gene co-expression network analysis. Statistical Applications in Genetics and Molecular Biology 2005;4(17).
[146] Zhang W, Fang HB, Song J. Principal component tests: applied to temporal gene expression data. BMC Bioinformatics 2009;10(S26):1–9.
[147] Zhang Y, Xuan J, Reyes BGDL, Clarke R, Ressom HW. Reverse engineering module networks by PSO-RNN hybrid modeling. BMC Genomics 2009;10(S15):1–10.
[148] Zhou X, Kao MCJ, Wong WH. Transitive functional annotation by shortest-path analysis of gene expression data. Proceedings of the National Academy of Sciences of the United States of America 2002;99(20):12783–8.
[149] Zhu D, Dequeéant M-L, Li H. (2008) Comparative analysis of clustering methods for microarray data. In: Emmert-Streib F, Dehmer M, editors. Analysis of microarray data: a network-based approach, Weinheim, Germany; Wiley-VCH Verlag GmbH & Co. KGaA; http://dx.doi.org/10.1002/9783527622818.ch2.
[150] Zhu D, Hero AO, Cheng H, Khanna R, Swaroop A. Network constrained clustering for gene microarray data. Bioinformatics 2005;21(21):4014–20.