POLITECNICO DI TORINO
Repository ISTITUZIONALE

MicroClAn: Microarray Clustering Analysis

Original
MicroClAn: Microarray Clustering Analysis / Bruno, Giulia; Fiori, Alessandro. - In: JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING. - ISSN 0743-7315. - 73:3(2013), pp. 360-370. [10.1016/j.jpdc.2012.09.008]

Availability:
This version is available at: 11583/2502981 since:

Publisher:
Viktor Prasanna, Elsevier

Published
DOI:10.1016/j.jpdc.2012.09.008

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 October 2024

MicroClAn: Microarray Clustering Analysis

Giulia Bruno[a], Alessandro Fiori[b,1,*]

[a]*Dipartimento di Ingegneria Gestionale e della Produzione, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino, Italy*
[b]*Fondazione Piemontese per la Ricerca sul Cancro-Onlus (FPRC), Institute for Cancer Research and Treatment (IRCC), Str. Prov. 142 Km. 3.95, 10060, Candiolo, Italy*

Abstract

Evaluating clustering results is a fundamental task in microarray data analysis, due to the lack of enough biological knowledge to know in advance the true partition of genes. Many quality indexes for gene clustering evaluation have been proposed. A critical issue in this domain is to compare and aggregate quality indexes to select the best clustering algorithm and the optimal parameter setting for a dataset. Furthermore, due to the huge amount of data generated by microarray experiments and the requirement of external resources such as ontologies to compute biological indexes, another critical issue is the performance decline in term of execution time. Thus, the distributed computation of algorithms and quality indexes becomes essential. Addressing these issues, this paper presents the MicroClAn framework, a distributed system to evaluate and compare clustering algorithms using the most exploited quality indexes. The best solution is selected through a two-step ranking aggregation of the ranks produced by quality indexes. A new index oriented to the biological validation of microarray clustering results is also introduced. Several scheduling strategies integrated in the framework allow to distribute tasks in the grid environment to optimize the completion time. Experimental results show the effectiveness of our aggregation strategy in identify the best rank among different clustering algorithms. Moreover, our framework achieves good performance in terms of completion time with few computational resources.

Keywords: Microarray, clustering analysis, quality indexes, rank aggregation, scheduling strategies

1. Introduction

Microarray technology allows the measurement of expression levels of thousands of genes simultaneously, thus it has become a fundamental tool in genomic research for studying biological processes of cancer pathologies [18, 31, 42, 47]. Clustering analysis of gene expressions is often the first step in microarray data analysis to discover co-expressed genes under different experimental conditions [23]. Furthermore, clustering can be used as a pre-processing step before a feature selection or a classification algorithm, to restrict the analysis to a specific category or to avoid redundancy by considering only a representative gene for each cluster.

Many conventional clustering algorithms have been applied or adapted to gene expression data, and new algorithms have been proposed specifically aiming at gene expression data (see [7, 12, 32, 25] for a survey). If a time information is available, time-series microarray data can be analyzed to group genes that show a similar trend across time instances [10, 27]. Otherwise, other approaches have been defined to group genes which have a common behavior [33, 2, 16, 1, 22, 37]. For example, [16] describes the application of the fuzzy c-means to microarray data, to overcome the problem that a gene can be associated to more than

*Corresponding author
Email addresses: giulia.bruno@polito.it (Giulia Bruno), alessandro.fiori@ircc.it (Alessandro Fiori)
[1]Telephone: +39 011 090 7084. Fax: +39 011 090 7099

one cluster. In [33], a fuzzy majority voting approach is proposed to combine the clustering solutions in the resultant Pareto-optimal set. [1] proposed an algorithm that adopts the idea of the k-means to cluster genes by replacing the distance measure with the interdependence redundancy measure. A variation of the hierarchical clustering algorithm is proposed in [22]. The basic idea is to consider a cluster as a high-dimensional dense area, where data objects are attracted with each other. Differently, in [3] the authors integrate in a hierarchical clustering method a new similarity measure to evaluate the similar expression profiles according to the samples' phenotypes.

Once a clustering algorithm has been proposed, there is the issue of evaluating results. Evaluating clustering results is a fundamental task to identify the best algorithm and parameters (e.g., similarity measure, number of clusters). The selection of the best clustering algorithm allows to improve the quality of further analyses oriented to identify patterns in the data. In literature many quality indexes for clustering evaluation have been proposed [51]. In the biological context of microarray data analysis, among the most exploited indexes there are classical data mining measures, such as Silhouette and Precision. However, other quality measures have been introduced to evaluate the biological meaning of clusters. For example, in [13] two biological indexes based on the categorization of the biological functions of genes have been proposed.

Some tools have been developed to compute quality indexes for microarray clustering [6]. For example, FunSpec [40] evaluates the biological enrichment of clusters of genes, while Onto-Express [21] automatically translates lists of differentially regulated genes into functional profiles characterizing the impact of the studied condition. However, these approaches provide only the quality index values, leaving to the final user the task of comparing and aggregating the results in order to select the best clustering result.

Some recent works addressed the issue of comparing and aggregating quality indexes to identify a rank among clustering results to select the best one [15, 35]. For example, in [35] a Monte Carlo method is exploited to combine different ranks produced by some quality indexes. In [36] an R package exploiting the Kemeny theory to aggregate different ranks is presented. These approaches aim at achieving the best rank aggregation disregarding the high computational cost of the clustering analysis.

In microarray analysis the management of the huge amount of data generated by microarray experiments becomes a critical issue. Since research centres often do not have efficient computational resources, some efforts have been done to optimize the microarray analysis [43]. Particularly, the partition of data and distribution to several nodes can solve the main memory problems and accelerates the methods. Clustering algorithms, especially the hierarchical ones, require a lot of time for analyzing a microarray dataset. Moreover, some quality indexes need external resources, such as ontologies. The interactions with these resources increase the overall computational cost. To discover the best clustering algorithm, multiple iterations are needed, thus, the distributed computation becomes essential.

Research efforts have been devoted to manage very large datasets and optimize the computation of data mining algorithms using distributed and parallel approaches [4, 11, 29, 52]. For instance, the fragmentation techniques, presented in [4] and based on structural constraints of XML documents, allows improving query processing on distributed environments. Similarly, in [11] a compression technique for multidimensional data cubes is exploited to extract and browse compressed two-dimensional OLAP views, coming from remote OLAP servers, on mobile devices. Differently, in [8] a parallel implementation of spectral clustering approach is presented. However, a distributed system to perform clustering analysis of microarray data is not yet proposed in literature. To the best of our knowledge, this is the first attempt to propose a distributed framework oriented to the comparison of different clustering algorithm in order to improve the quality of the final results.

Addressing the previously described issues, this paper presents the MicroClAn framework, a distributed system to evaluate and compare clustering algorithms using the most exploited quality indexes. A two-step aggregation procedure based on the Kemeny and Condorcet theories identifies the best result over all the ranks produced by the considered quality indexes. Furthermore, several scheduling strategies are exploited to distribute tasks in the grid environment to optimize the overall execution time.

The paper is organized as follows. Section 2 introduces the basic concepts for microarray cluster analysis and quality indexes. Section 3 presents the problem of ranking aggregation, while Section 4 describes the exploited strategies to distribute tasks in a grid environment. In Section 5 the contributions and the architecture of the MicroClAn framework are presented. Section 6 describes experimental results to validate

the two-step ranking aggregation, show the efficiency of the distributed computation, and evaluate the scheduling strategies. Finally, Section 7 draws conclusions and presents future developments of this work.

2. Quality indexes for microarray clustering

Let N be the number of genes and M the number of samples in a microarray experiment. After the clustering process, each gene x belongs to one of the k clusters $(C_1, C_2, ..., C_k)$ that are generated. Let n_i be the number of genes in cluster C_i. The notation $d(x, y)$ represents the distance measure between gene x and y, computed according to the distance measure exploited in the clustering process. Analogously, the notation $d(C_i, C_j)$ represents the distance measure between cluster C_i and C_j, computed according to the cluster distance measure exploited in the clustering process.

Instead of a distance measure, a similarity measure can be computed among elements. We refer to similarity between objects with the notation $s(x, y)$. Also this measure depends on the exploited clustering algorithm. If there is the need to switch from distance to similarity or vice-versa, if both are normalized in the range $[0, 1]$, the following relation holds: $d(x, y) = 1 - s(x, y)$ [46].

Based on following previous classifications both in the general [46] and microarray [12, 34, 20] context, we divided clustering indexes in the three categories of internal, external and biological indexes. The most exploited indexes belonging to each category are discussed in the following.

2.1. Internal indexes

Internal indexes evaluate the cohesion among cluster elements and the separation among clusters. These evaluations are done without knowing the true partition of objects. Among the most exploited indexes in this category, there are the Silhouette, the Homogeneity, the Separation, the Davies-Bouldin index, and the Figure of Merit. A brief description of each of them is reported in the following.

The Silhouette index [41] measures the appropriateness of a gene being in a specific cluster rather than in the others. The Silhouette value lies between -1 and 1. When it is close to 1 it means that the gene is appropriately clustered, while if it is close to -1 it means it would be more appropriate if the gene was clustered in its neighboring cluster. When the value is near zero, it means that the gene is on the border of two clusters.

The Silhouette of gene x is computed as

$$Silhouette_x = \frac{b(x) - a(x)}{max\{a(x), b(x)\}} \tag{1}$$

where $a(x)$ is the average distance between x and all other genes within the same cluster, and $b(x)$ is the average distance between x and all other genes within the nearest cluster.

Even if the Silhouette is defined for a single gene, it can be computed for a single cluster or the whole clustering, simply by summing the Silhouette values of genes in a cluster or in all clusters respectively. The average Silhouette of genes in a cluster is a measure of how tightly grouped are the data in the cluster and the average Silhouette of the entire dataset is a measure of how appropriately the data has been clustered.

The Homogeneity index [44] measures the compactness of clusters by analysing the similarity among elements in each cluster. The Homogeneity of cluster C_i is defined as follows:

$$Homogeneity_i = \frac{2}{n_i(n_i - 1)} \sum_{x,y \in C_i, x<y} s(x, y) \tag{2}$$

The Homogeneity increases if the solution improves. The Homogeneity can be also defined for the whole clustering, as the average value of Homogeneity of all clusters.

The Separation index [44] evaluates if the distance between members of the same cluster is lower than the distance between members of different clusters. It is defined as follows:

$$Separation = \frac{2}{N(N-1) - Q} \sum_{x \in C_i, y \in C_j, i \neq j, x<y} s(x, y) \tag{3}$$

where Q is the total number of gene pairs that are in the same cluster (i.e., $Q = \sum_{i=1}^{k} n_i(n_i - 1)$). The Separation decreases if the solution improves.

The Davies-Bouldin index (DBI) [14] combines the homogeneity and the separation in a single measure. It evaluates the average similarity between a cluster and its most similar one and it is defined as follows:

$$\text{DBI} = \frac{1}{k}\sum_{i=1}^{k} \max_{j,j\neq i}\left\{\frac{d_{intra}(C_i) + d_{intra}(C_j)}{d(C_i, C_j)}\right\} \tag{4}$$

where $d_{intra}(C_i)$ is the average distance between values and the centroid of cluster i, i.e., $d_{intra}(C_i) = \frac{1}{n_i}\sum_{x\in C_i} d(x, z_i)$ and z_i is the cluster representative (e.g., centroid). Low values of DBI correspond to good cluster quality.

The Figure of Merit (FOM) [51] estimates the predictive power of a clustering algorithm by doing the following steps: (i) the experiment j is removed from the data set, (ii) the genes are clustered basing on the remaining data and (iii) the within-cluster similarity of expression values in experiment j is measured. Particularly, when the experiment j is removed from the data, the FOM is defined as follows:

$$FOM(j) = \sqrt{\frac{1}{N}\sum_{i=1}^{k}\sum_{x\in C_i}(e_{xj} - \bar{e}_{j,i})^2} \tag{5}$$

where e_{xj} is the expression value of gene x in the experiment j in the raw data and $\bar{e}_{j,i}$ is the average expression value in condition j of genes in cluster C_i. This estimation is repeated M times to compute the aggregate figure of merit which evaluates the total predictive power of the clustering algorithm over all the conditions.

2.2. External indexes

If the object class labels are a priori known, the clustering result can be compared with the true partition. The indexes that are used to measure the agreement between cluster labels and the true partition are usually named external indexes. The most exploited external indexes are Precision, Recall, F-measure, Rand index, Jaccard index and Minkowski index.

In the following, we refer as $T_1, T_2, ..., T_h$ as the true class partition, while the obtained clustering is indicated as previously described $C_1, C_2, ..., C_k$. The number of objects of class j in cluster i is indicated as n_{ij}.

The precision of a cluster i with respect to a class j is the fraction of objects of class j in cluster i, while the recall of a cluster i with respect to a class j is the extent to which cluster i contains all objects of class j [46]. They are computed as follows:

$$Precision(i, j) = \frac{n_{ij}}{n_i} \tag{6}$$

$$Recall(i, j) = \frac{n_{ij}}{n_j} \tag{7}$$

where n_j is the number of object in the class j.

The precision and recall values can be combined to measure the extent to which a cluster contains only objects of a particular class and all objects of that class. Such measure is called F-measure [28] and is calculated as follows:

$$F(i, j) = \frac{2Precision(i, j)Recall(i, j)}{Precision(i, j) + Recall(i, j)} \tag{8}$$

The F-measure values are in the interval $[0, 1]$, and the larger its values, the better the clustering quality is. The F-measure of the whole clustering is the average value of the F-measures of each cluster.

Other measures have been defined with the aim of comparing results of two partitions. They can be used to compare the obtained clusters both to the true class partition and to a partition obtained by a different

clustering algorithm. The most used similarity oriented measure are the Rand index [38] and the Jaccard index [19].

The Rand index [38], also called Simple Matching [50], is defined as the fraction of agreements between two partitions, i.e., the number of object pairs that are either in the same group in both partitions or in different groups in both partitions, divided by the total number of objects. It lies between 0 and 1, and it is 1 if the two partitions agree perfectly. Also the Jaccard index [19] measures the similarity between two partitions, by using a little different formula, as reported below. If there is the need of presenting the proportion of disagreement instead of agreement between two partitions, the Minkowski index [45] can be exploited.

Considering two partitions C and T of N objects, for each pair of objects (i, j) there are four possible cases: 1) i and j are in the same cluster both in C and T, 2) i and j are in the same cluster in C but in different clusters in T, 3) i and j are in the same cluster in C but in different clusters in T, 4) i and j are in different clusters both in C and T.

Assuming that a, b, c, and d are the number of object pairs in cases 1, 2, 3, and 4 respectively, the formulas of the previously described indexes are:

$$Rand = \frac{a+d}{a+b+c+d} \tag{9}$$

$$Jaccard = \frac{a}{a+b+c} \tag{10}$$

$$Minkowski = \sqrt{\frac{b+c}{a+c}} \tag{11}$$

2.3. Biological indexes

Differently from other contexts, the application of external measures is rarely possible on real microarray datasets because the true partition is not a priori known and there are few benchmarks to evaluate and compare algorithms. Since genes with similar expression profiles may imply similarity among their functions in the biological activities, gene clusters may represent specific biological functions. The idea of biological measures is to analyze biological annotations of genes in the same cluster and verify if they are coherent.

The most used biological measure is the functional enrichment. A cluster of genes is said to be enriched for a functional category if the proportion of genes within the cluster known to be in that category exceed the number that could reasonably be expected from random chance [40]. The degree of functional enrichment for a given cluster c and functional category f can be quantitatively assessed by the hyper-geometric data distribution P. For each category and cluster, the probability of observing such an overlap by chance is calculated as:

$$P_{c,f} = 1 - \sum_{i=0}^{n_{c,f}-1} \frac{\binom{n_f}{i}\binom{G-n_f}{n_c-1}}{\binom{G}{n_c}} \tag{12}$$

where G is the size of genome, n_f is the number of genes of the genome in the considered category, n_c is the cluster size and $n_{c,f}$ is the number of genes in the cluster which are in the same category. If this probability is sufficiently low (e.g., low than 0.01), then the cluster is said to be enriched for that category. Usually the biological enrichment is evaluated according to the Gene Ontology categories [30].

3. Quality index aggregation

In some applications it is possible to select the best clustering method by visualizing the clustering results. However, in microarray experiments a visual approach is not feasible due to the high number of dimension (i.e., samples). The quality indexes described in Section 2 can be exploited to identify the algorithm which obtain the best results.

Quality indexes evaluate clustering results by given a score to each of them. Thus, each index produce a list of results sorted by score. Even if scores are not directly comparable among indexes because they range within different values, the sorted lists can be compared to select the best results.

The problem of combining ranking alternatives is called the rank aggregation problem [17]. Let U be a set of elements, named universe. An ordered list τ with respect to U is a ranking of a subset $S \subseteq U$, i.e., $\tau = [x_1 \geq x_2 \geq ... \geq x_d]$, where $x_i \in S$ and $\geq$ is a ordering relation on S. Let $p^\tau(i)$ be the position of x_i in τ. Lowest values of $p^\tau(i)$ represent best elements.

In our context, U is the set of clustering results produced by different clustering algorithms. Each quality index produces a rank τ_i of the element in the universe.

Two exploited solutions to the rank aggregation problem are reported in the following sections.

3.1. Copeland aggregation

The Copeland aggregation belongs to the Condorcet methods, i.e., voting systems that selects the candidate (among elements belonging to a universe U) that is preferred by more order lists τ than every other candidate.

All the Condorcet methods exploit a data matrix to count for each pair of candidates (x_i, x_j) the number of time in which x_i wins against x_j (i.e., $p^\tau(i) \leq p^\tau(i)$) and the number of time in which x_i looses against x_j (i.e., $p^\tau(i) \geq p^\tau(i)$). The winner of each pairing is the candidate preferred by a majority of voters.

When all possible pairings of candidates have been considered, if one candidate always beats every other candidate then it is declared the Condorcet winner. If there is no Condorcet winner, a further method must be used to find the winner, and this mechanism varies from one Condorcet method to another.

Some Condorcet methods produce not just a single winner, but a ranking of all candidates from first to last place. The Copeland method is a Condorcet method in which candidates are ordered by the number of pairwise victories minus the number of pairwise defeats. When there are multiple members of the Smith set, which is the smallest non-empty set of candidates such that each member beats every other candidate outside the set in a pairwise election, this method often leads to ties.

3.2. Kemeny optimal aggregation

Differently from Condorcet methods, other methods aims at minimizing the total disagreement (in term of a distance function) among the several input rankings and their aggregation. The aggregation obtained by optimizing the Kendall distance is called Kemeny optimal aggregation. The problem can be formalized as an optimization task defined as:

$$\sigma^* = argmin \sum_{i=1}^{k} K(\sigma, \tau_i) \tag{13}$$

where $K(\sigma, \tau_i)$ is the Kendall distance, τ_i is an order list, and σ is the optimal aggregation list.

The Kendall τ distance [17] evaluates the number of pairwise disagreements between two lists. The distance is defined as follows:

$$K(\gamma, \delta) = \sum_{t,u \in \gamma \cup \delta} K^p_{tu} \tag{14}$$

where

$$K^p_{tu} = \begin{cases} 0 & if \ \ [r^\gamma(t) < r^\gamma(u) \wedge r^\delta(t) < r^\delta(u)] \\ & \quad \vee [r^\gamma(t) > r^\gamma(u) \wedge r^\delta(t) > r^\delta(u)] \\ 1 & if \ \ [r^\gamma(t) > r^\gamma(u) \wedge r^\delta(t) < r^\delta(u)] \\ & \quad \vee [r^\gamma(t) < r^\gamma(u) \wedge r^\delta(t) > r^\delta(u)] \end{cases} \tag{15}$$

where γ e δ are two order lists, t and u are two elements of γ e δ, respectively, and belonging to U. Moreover, $r^\gamma(t)$, $r^\gamma(u)$, $r^\delta(t)$ e $r^\delta(u)$ are the rank positions of t e u in the γ and δ lists. Lowest the Kendal distance is, highest is the similarity between the lists.

4. Scheduling algorithms

The computational cost of clustering execution and results evaluation is very expensive due to the complexity of these tasks. Several approaches to define the best strategy to distribute the tasks in a grid environment have been proposed [9, 26, 49, 5]. In the following we present some scheduling strategies which are integrated in our framework.

Given a metatask M defined as a set of independent tasks m_i and a set of N grid nodes n_j, let $t_{e_{ij}}$ be the execution time for task m_i on node n_j and $t_{c_{ij}}$ be the completion time for task m_i on node n_j given as the sum of the execution time of tasks previously assigned to n_j plus the $t_{e_{ij}}$ value. Furthermore, let t_{e_M} be the total execution time of all the tasks belong to M. The aim of the scheduling approaches is to determine the best strategy of scheduling tasks on nodes to minimize t_{e_M}.

We define the Expected Time Matrix (ETM) a matrix representation of the information for the grid environment. The ETM is a matrix of dimensions $T \times N$, where T is the number of tasks and N is the number of the grid nodes. Each cell of ETM stores the average execution time of each task on each grid node on several repetitions, i.e., average values of $t_{e_{ij}}$.

An ETM can present two properties: (i) linearity and (ii) homogeneity. The linearity depends on the node characteristics, while the homogeneity depends on task characteristics. An ETM is linear if given two nodes A and B, the execution time of each task on A is less/greater than the corresponding execution time on node B, i.e., A is always faster/slower than B. An ETM matrix is homogeneous if given a node A, the execution time of a task is very close to the execution times of all the other tasks, otherwise it is heterogeneous.

In the experimental section we will analyze different scheduling approaches based on the Expected Time Matrix evaluation. In the following a brief description of each approach is provided.

Opportunist load balancing (OLB). The Opportunistic Load Balancing approach assigns in a casual order each task to the first grid node available without considering the execution or completion times [5, 9, 26]. The aim is to exploit all the resources of the grid environment. This approach usually achieve bad results because there is no optimization over the execution times of tasks, but it is a good assignment strategy if the ETM is homogeneous.

Minimum Execution Time (MET). The Minimum Execution Time approach assigns in an arbitrary order each task to the grid node with the minimum execution time for that task, disregarding the availability of other nodes or the task completion time [48]. This approach aims at minimizing the execution time of each single task. However, the grid nodes with the best resources is usually overloaded.

Minimum Completion Time (MCT). The Minimum Completion Time approach is similar to the MET approach, but it assigns each task to the grid node with the minimum completion time for that task. This method exploits the benefits of the two previous approaches avoiding the critical aspects of both. However, since the tasks are considered in a casual order, the assignment can be not optimal, especially for heterogeneous ETM.

Min-min. Differently from the previous approaches which consider each task at a time, the Min-min approach evaluates the completion times of all tasks before assigning them to the grid nodes. For each task, the completion time for each grid node is retrieved. Then, the task with the minimum completion time is assigned to the corresponding grid node and it is removed from M. This procedure is repeated until M is not empty. This approach is more efficient but its complexity is $O(|M|)$, where $|M|$ is the number of tasks belonging to M.

Max-min. Similarly to the previous one, also the Max-min approach evaluates the completion times of all tasks before assigning them to the grid nodes. However, it starts assigning slowest tasks instead of fastest ones. In fact it selects the task m_i with the maximum completion time and assigns it to the grid node n_j with the minimum completion time for that task. Then, task m_i is removed form M and the procedure is repeated until M is not empty. This approach is particularly suitable for ETM heterogeneous.

5. MicroClAn framework

The MicroClAn framework provides an evaluation system for microarray clustering results based on a distributed execution. It provides two main contributions to the clustering analysis as well as the ranking aggregation: (1) the definition of a biological score and (2) the exploitation of a two-step ranking aggregation. The new contributions and the framework architecture are described in the following paragraphs.

5.1. BioScore

Since the enrichment does not assign a score to the whole clustering results, we defined a new the biological score $BioScore$. This score evaluates how many subset of genes show a p-value under a threshold (e.g., 0.01). It is defined as follows:

$$BioScore = \sum_{i}^{K} \sum_{cat}^{G} \delta_{i,cat} \tag{16}$$

where K is the number of genes in the considered dataset, G is the number of biological function categories stored in the external resource (e.g., GO). $\delta_{i,cat}$ is defined as follows:

$$\delta_{i,cat} = \begin{cases} 1 & \text{if } P_{i,cat} < t \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where $P_{i,cat}$ is the p-value related to the category cat associated to gene i, and t is a threshold. In this way all the biological categories associated to one gene are considered for the enrichment.

5.2. Two-step ranking aggregation

Since quality indexes belong to different categories and can have different priorities, it is not possible to directly compare the ranks produced by them, but a preliminary aggregation is needed among index of the same category. Thus, we performed a two-step aggregation.

The first step define the optimal rank for each index category by means of the Kemeny optimal aggregation. In this way, indexes evaluating the same aspect are grouped and analyzed together. Then, a Copeland aggregation is exploited to provide a final rank from the three category ranks. Furthermore, in the MicroClAn framework it is possible to assign a weight to each category to give different relevance to each of them or to exclude some of them if they are not relevant for the analysis.

The Kemeny aggregation performed on each index category guarantees that the best rank is identified. Moreover, using this approach, it is not possible to retrieve more than one optimal rank as with Condorcet approaches in some particular scenarios. The Copeland method, which is less computational expensive than Kemeny, is exploited to identify the winner clustering algorithm. The quality of the final rank may be less accurate only for the last positions, but it guarantees to identify the best approach.

Running example. Suppose an analyst wants to perform a comparison among four different clustering methods, named in the following A, B, C and D. The comparison may be based on the following measures: (i) Silhouette, (ii) Homogeneity, (iii) Rand Index, (iv) Jaccard and, (v) BioScore. As discussed in Section 2, the first two measures belong to the internal quality index category, while the third and fourth in the external one. The BioScore, which is based on the enrichment, belongs instead to the biological category. Suppose, for instance, to obtain the results resumed in Table 5.2 for each quality index. The two-step aggregation strategy integrated in the MicroClAn platform, as first step, considers each category separately. The Kemeny aggregation method is exploited to define the optimal rank according to the different measures belonging to the considered quality index category. For instance, in the internal category, the Kendall τ computed by the Formula 14 for the rank $[A, B, C, D]$ is equal to 0.43, while for $[A, B, D, C]$ is 0.28. Since, the Kemeny method should minimize the Kendall distance, the optimal rank for this category is $[A, B, D, C]$. Similarly, the optimal rank for the external category is $[B, D, C, A]$. Since in this example there is only quality index in the biological category, the MicroClAn framework will skip the aggregation strategy for this category. When all the ranks for each category are defined, the Copeland method is exploited to extract the final

Quality index category	Measure	Pos. 1	Pos. 2	Pos. 3	Pos. 4
Internal	Silhouette	$A = 0.97$	$C = 0.21$	$B = 0.09$	$D = 0.08$
	Homogeneity	$A = 0.99$	$B = 0.93$	$D = 0.92$	$C = 0.89$
External	Rand index	$D = 0.6$	$B = 0.56$	$C = 0.34$	$A = 0.03$
	Jaccard	$B = 0.49$	$D = 0.42$	$C = 0.33$	$A = 0.32$
Biological	BioScore	$A = 0.83$	$D = 0.77$	$C = 0.43$	$B = 0.18$

Table 1: Example of ranks according to different quality indexes. All the values are normalized in the range [0,1].

rank. The Copeland method orders the clustering algorithms by the number of pairwise victories, minus the number of pairwise defeats. Thus, for each method we obtain the following results: $A = 3$, $B = 2$, $C = 0$ and $D = 1$. The final rank results $[A, B, D, C]$.

5.3. Architecture

The MicroClAn framework is based on a distributed environment. The user interacts with the services provided by different servers by means of a client. The client provide two contribution to the system: (i) the scheduling of the tasks and (ii) the aggregation of the results. In MicroClAn the scheduling of the tasks can be performed by one of the methods discussed in Section 4, while the aggregation procedure is based on the two-step ranking aggregation presented in Section 5.2. The tasks provided by the servers are divided into four categories: (i) clustering algorithms, (ii) internal indexes, (iii) external indexes and (iv) biological indexes. Each server in the grid environment provides the subset of these services compatibly to its resources. For instance, the biological indexes are provided only by the servers which can access to Gene Ontology database. The location of the servers and the services available on each server are registered in a nameserver. The client exploits the nameserver information to submit the task to the servers which are registered and available in the distributed environment.

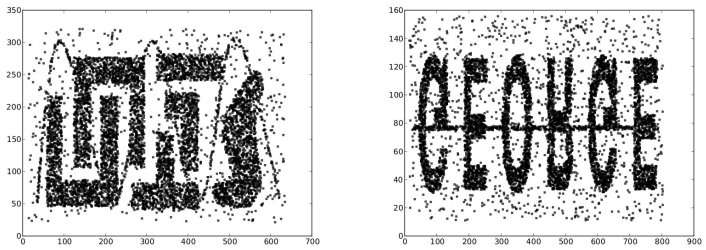
All the communications among client, servers and nameserver are based on the TCP/IP protocol. The servers send a registration request to the nameserver specifying the services which are available. The nameserver registers the service information and their location. The analyst, using the client application, defines the experimental design (i.e., metatask) which will be applied to one microarray dataset. The metatask is composed by tasks which are clustering algorithms and quality indexes of interest. The client retrieves the information about the network addresses of the servers and the services available on the grid environment from the nameserver. According to the information stored in the ETM, a scheduling approach is performed to assign the tasks to the servers which provide the right services and optimize the objective function of the scheduler. According to the scheduling strategy, the client send each task to the selected server using the address provided by the nameserver. When one server accomplish their tasks, it returns to the client the results. The client collects the results from the servers and send the necessary data to the other servers, if it is necessary. When all the tasks are accomplished by the servers, the client exploits the aggregation procedure to define the rank among the clustering approaches using the parameters set by the analyst.

6. Experimental results

We evaluated the MicroClAn framework by means of three sets of experiments, addressing the following issues: (i) the validation of the two-step rank aggregation, (ii) the efficiency of the distributed computation, and (iii) the evaluation of scheduling performance.

6.1. Validation of ranking aggregation

We evaluated the quality of the two-step rank aggregation described in Section 3 on both simulated and real datasets. Since no benchmark microarray dataset containing gene class labels is provided, to evaluate the ability of external indexes we choose two synthetic datasets for which the true partition is known [24]. In Figure 1 the two datasets, named in the following dataset A and B respectively, are shown. Seven different clustering algorithms have been exploited:



(a) dataset A (b) dataset B

Figure 1: Visualization of the two sysntethic datasets on which different clustering algorihtms have been applied

Position	Silhouette	Homogeneity	Separation	FOM	DBI
1	DBS	HR-A,HR-C,HR-S	HR-A	KM-A	KM-A
2	PAM		HR-C	DBS	DBS
3	HR-S		PAM	KM-M	KM-M
4	KM-A	DBS	DBS	PAM	PAM
5	KM-M	KM-A	KM-M	HR-C	HR-S
6	HR-A,HR-C	PAM	KM-A	HR-S	HR-C
7		KM-M	HR-S	HR-A	HR-A

Table 2: Internal quality index ranking for the dataset A

A) *DBS*: the density based clustering algorithm (DBSCAN)
B) *PAM*: the K-medoids method using the Euclidean distance
C) *KM-A*: the K-means based on the average centroid
D) *KM-M*: the K-means based on the median centroid
E) *HR-S*: the hierarchical clustering approach with single-linkage
F) *HR-C*: the hierarchical clustering approach with complete-linkage
G) *HR-A*: the hierarchical clustering approach with average-linkage

The algorithm parameters have been set according to the analyzed dataset. For instance, on the synthetic datasets the parameter K has been set equal to 6 since there are 6 natural clusters. Differently, the DBSCAN parameters have been set equal to $\epsilon = 12, minPts = 14$ for the dataset A, $\epsilon = 11, minPts = 6$ for the dataset B. We tested also different similarity measures for the K-means and hierarchical algorithms on both datasets. For the dataset A, we set the Manhattan distance for the K-means algorithms, while the Spearman similarity for the hierarchical clustering methods based on single and average linkage and the Manhattan distance for complete-linkage approach. Differently, on the dataset B we exploited the following similarity measure: (i) Manahattan distance for K-means based on the average centroid, (ii) Euclidean distance for DBScan, K-medoids and K-means based on the median centroid, (iii) the Pearson similarity for single-linkage hierarchical method, (iv) the Spearman measure for average-linkage hierarchical approach, and (v) the cosine similarity for complete-linkage algorithm. In the results, we will report only the identifier of the clustering method.

These algorithms represents the most exploited methods in clustering analysis. Indeed, the k-means approach, despite its simplicity, it is proved to be a good performing algorithm on microarray data [39]. The hierarchical clustering algorithm is instead frequently used in biological studies since it provide a natural way to graphically represent the results through a dendrogram [23]. Differently, the density-based algorithm (DBScan) allows the assignment of outlier samples to a separate noise cluster, thus being suited for microarray-based studies [37].

Considering the dataset A, different ranks, reported in Table 2, are retrieved for each internal quality index. Analyzing these ranks is very difficult to identify the best clustering approach. However, combining

Position	F-measure	Rand	Jaccard	Mikowski
1	DBS	DBS	DBS	DBS
2	KM-A	KM-A	KM-A	HR-A
3	KM-M	PAM	KM-M	HR-S
4	PAM	KM-M	PAM	HR-C
5	HR-C	HR-C	HR-C	PAM
6	HR-S	HR-S, HR-A	HR-S	KM-M
7	HR-A		HR-A	KM-A

Table 3: External quality index ranking for the dataset A

Position	Silhouette	Homogeneity	Separation	FOM	DBI
1	DBS	DBS	HR-S	DBS	DBS
2	KM-A	PAM	HR-C	KM-M	KM-A
3	KM-M	KM-A	HR-A	KM-A	KM-M
4	PAM	KM-M	DBS	PAM	PAM
5	HR-C	HR-S	KM-A	HR-C	HR-S
6	HR-A	HR-C	KM-M	HR-S	HR-C, HR-A
7	HR-S	HR-A	PAM	HR-A	

Table 4: Internal quality index ranking for the dataset B

the internal index ranks with the Kemeny optimal aggregation, the final rank is DBS, PAM, KM-M, KM-A, HR-C, HR-A, HR-S. If the external indexes are analyzed (Table 3), a different rank is retrieved by the Kemeny approach, i.e. DBS, KM-A, PAM, KM-M, HR-C, HR-S, HR-A. Even if the two ranks are quite similar, some methods achieve different performance with respect to the two sets of quality indexes. By exploiting the two-step aggregation, the MicroClAn framework allows to identify the optimal rank with respect to both indexes' categories which is: DBS, PAM, KM-A, KM-M, HR-C, HR-S, HR-A.

The DBSCAN approach results the best one, as also demonstrated by the visual inspection. In Figure 2 the clusters identified by (a) the DBSCAN, (b) PAM, and (c) HR-A methods are reported. The visual inspection of the results highlights the effectiveness of our framework since we can notice the main differences among the clusters identified by these approaches. DBSCAN result is the best one, followed by PAM which captures one of the six clusters but the other clusters are merged or split due to the presence of noise points. Similar result is obtained with KM-A that assigns the most of the points to the same clusters selected by PAM. Differently, the hierarchical algorithms identify only two big clusters while the other four clusters are composed only by noise points.

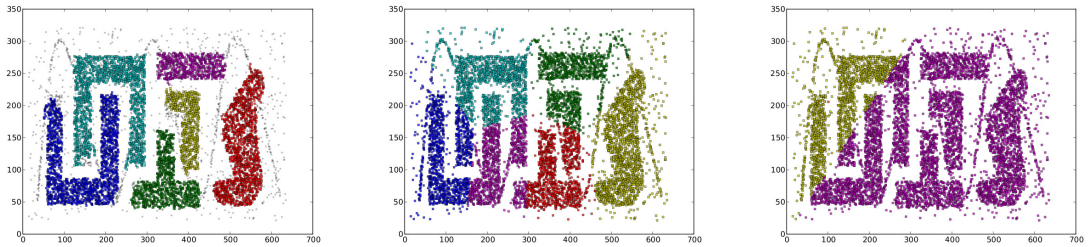
Similar results are obtained for the dataset B. In Table 4 and Table 5 are reported the ranks associated to each quality index belonging to the internal and external category respectively. The final rank obtained by the MicroClAn framework is the following: DBS, HR-C, HR-A, HR-S, PAM, KM-A, KM-M,. Also for this dataset, the visual inspection of the clustering results shows the effectiveness of our approach in select the best clustering approach as pointed out by the Figure 3.

The biological validation was performed on the Colon dataset[2] which contains the expression values of 2000 genes for 62 samples. Since the hierarchical clustering is one of the most clustering algorithm exploited in microarray data analysis, we compared the results of a hierarchical single-linkage approach with different values of K (i.e., from 5 to 20). We denote the different setting as HR0 for $k = 5$, HR1 for $k = 10$, HR2 for $k = 15$ and HR3 for $k = 20$. The internal and the biological indexes were exploited for the evaluation. The internal indexes identify HR0 as the best clustering method followed by HR2, HR1 and HR3. However,

[2] `http://datam.i2r.a-star.edu.sg/datasets/krbd/ColonTumor/ColonTumor.html`

Position	F-measure	Rand	Jaccard	Mikowski
1	DBS	DBS	DBS	DBS
2	HR-C	PAM	HR-C	HR-C
3	HR-A	KM-M	HR-A	HR-A
4	HR-S	KM-A	HR-S	HR-S
5	PAM	HR-C	PAM	PAM
6	KM-M	HR-S	KM-M	KM-M
7	KM-A	HR-A	KM-A	KM-A

Table 5: External quality index ranking for the dataset B



(a) DBS (b) PAM (c) HR-A

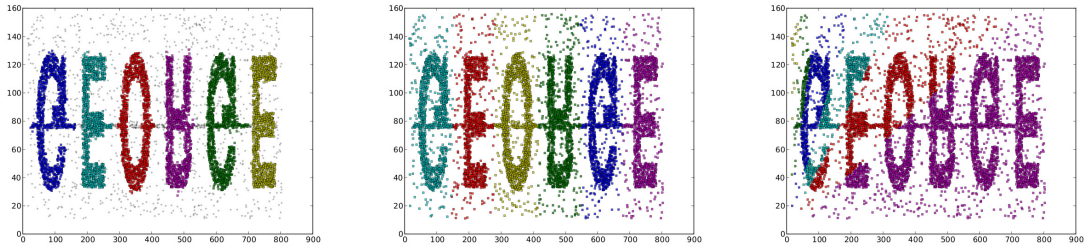
Figure 2: Visualization of different clustering results for the dataset A. (a) DBSCAN algorithm with $\epsilon = 12, minPts = 14$ using the Euclidean distance. (b) PAM method using the Euclidean distance and $K = 6$ as number of clusters. (c) Hierarchical average-linkage based on the Spearman similarity measure and the number of clusters $K = 6$.

the biological index retrieves a different rank equal to HR3, HR2, HR1 and HR0. The two ranks are in contradiction because the best algorithm for the internal indexes is the worst one in terms of biological meaning, and vice-versa. MicroClAn allows to overcome this contradiction by combing the two ranks. In fact, combining the two analyses MicroClAn identify the HR2 method as the best one, since it is a good trade-off between internal and biological evaluations.

6.2. Distributed execution

The main advantage of MicroClAn is the distributed execution of tasks, where a task is a clustering algorithm or a quality index computation. In Table 6 are reported the average execution time of each task. These statistics are related to several runs of MicroClAn on different datasets on one grid node. The grid node is a virtual machine with Linux server 64 bits with kernel 2.6.32-26-server, 4 GB of RAM using one Intel(R) Xeon(R) E5450 CPU @ 3.00GHz. The BioScore is the most expensive since, during the computation, many access to the database storing Gene Ontology are performed. Among the clustering algorithms the hierarchical clustering is the most expensive due to the update of the distance matrix exploited to identify each step of the aggregation. FOM is the internal index with a high complexity. In fact, the index should remove each object (e.g., sample), to evaluate the accuracy of the cluster result. The external indexes are the tasks with the lowest impact over all the clustering analysis process and they usually completed while other indexes are evaluated. Therefore, if the comparison of different clustering results is performed on one single machine, the completion time is affected by the most expensive tasks in terms of execution time and the number of clustering solutions which are involved in the experimental evaluation.

We evaluated the efficiency of MicroClAn by analyzing the execution time by using two different node types which present different performance on tasks. The node A is a virtual machine with Linux server 64 bits with kernel 2.6.32-26-server, while the node B runs Linux desktop 32 bits with kernel 2.6.32-30-generic-pae. Both nodes have 4 GB of RAM and each one uses one core of one Intel(R) Xeon(R) E5450 CPU @ 3.00GHz. The different performance are influenced mainly by the operating system architecture. For instance, the node



(a) DBS (b) KM-M (c) HR-C

Figure 3: Visualization of different clustering results for the dataset B. (a) DBSCAN algorithm with $\epsilon = 11, minPts = 6$ using the Euclidean distance. (b) K-means based on the median centroid using the Euclidean distance and $K = 6$. (c) Hierarchical complete-linkage based on the cosine similarity measure and the number of clusters $K = 6$.

	Task	**Execution time (s)**
Clustering	DBSCAN	45.20
	K-means	0.36
	PAM	17.68
	Hierarchical	69.88
Internal indexes	Silhouette	1154.96
	Homogeneity	15.64
	Separation	10.96
	FOM	3219.83
	DBI	3.18
External indexes	FMeasure	0.30
	Rand	0.31
	Jaccard	0.28
	Minkowski	0.33
Biological indexes	BioScore	23217.62

Table 6: Average execution time for MicroClAn tasks on one A node.

B ha better performance on computing the silhouette index, while the clustering algorithms are executed in less time on node A. Thus, in this case the ETM is not linear and not homogeneous. The experimental design was composed by evaluating two clustering algorithms, i.e. hierarchical and kmeans by means of four internal indexes (i.e., DBI, Silhouette, Separation and Homogeneity) on Colon dataset. We set the number of clusters equal to 50 and the similarity measure equal to the cosine. We run each experiments 10 times. In Table 7 are reported the average execution times respect to grid nodes involved in the computation.

The nodes A and B show different performance in completing tasks. The node B outperforms node A mainly in the index computation. In fact, the clustering algorithms are executed in around 5 seconds on each node, while the indexes show different execution times on the two node types. For instance, the silhouette index is computed on node A on average in 617 seconds while the node B terminates the task in 278 seconds. The scheduling procedure allows to reduce the completion time by a factor between 41% and 48% using two nodes. Adding more similar nodes does not reduce the completion time since there is one task which represents the bottleneck of the entire process. In the conducted experiments the silhouette index for hierarchical clustering is the most expensive task since the execution time is around 1000 and 300 seconds on node A and node B respectively. Thus, only when this task is completed the comparison of different ranks is performed. If we add nodes with better execution times for the silhouette task (i.e., node B), the completion time is reduced according to the performance of the new nodes on the critical tasks. For example, when two nodes B substitute two nodes A in a distributed environment of 4 nodes, the completion

# Nodes	Grid nodes	Clustering (sec)	Indexes (sec)	Total (sec)
1	A	5.088	2103.832	2108.920
1	B	5.946	595.834	601.780
2	$A-A$	4.566	1096.741	1101.307
2	$B-B$	5.336	348.293	353.629
4	$A-A-A-A$	4.554	1078.672	1083.226
4	$A-A-B-B$	4.504	323.721	328.225

Table 7: Average execution time using the distributed computing.

ETM	OLB (sec)	MET (sec)	MCT (sec)	Min-min (sec)	Max-min (sec)
Linear and homogeneous	4.41 ± 0.30	8.09 ± 0.13	3.99 ± 0.41	3.83 ± 0.12	4.38 ± 0.49
Linear and heterogeneous	9.67 ± 0.84	24.17 ± 2.26	10.86 ± 3.02	10.61 ± 1.88	8.84 ± 0.97
Not linear and homogeneous	3.76 ± 0.18	5.89 ± 0.14	3.62 ± 0.19	3.49 ± 0.18	3.90 ± 0.29
Not linear and heterogeneous	4.40 ± 0.73	3.98 ± 0.32	3.64 ± 1.46	3.80 ± 0.88	3.48 ± 1.20

Table 8: Average execution time expressed in seconds for five scheduling strategies and initialized ETM.

time is reduced by a factor of 70%. In this case the scheduler will assign the silhouette task always to the nodes B which show the best performance for the silhouette index.

6.3. Scheduling strategies

This section aims at evaluating the different scheduling strategies that can be exploited in the MicroClAn framework. Each scheduling approach, discussed in Section 4, presents advantages and disadvantages. Thus, we performed a set of experiments to explore all the possible scenarios. Based on the execution times collected on different runs of our framework, we simulated four different scenarios which capture all the possible ETM property configurations: (i) ETM linear and homogeneous, (ii) ETM linear and heterogeneous, (iii) ETM not linear and homogeneous, and (iv) ETM not linear and heterogeneous.

When the framework runs for the first time, the ETM is not initialized. All the execution times related to each task for each node are set to -1. Since the execution times are not available, all the scheduling approaches, except OLB, assign all the tasks to one node each run during the first runs until the ETM is not filled. Differently, the OLB approach balances the task among all the grid nodes independently to execution time information. When the ETM matrix is filled with the execution times for each task, each scheduling strategy balance the workload among all the nodes according to its strategy. Therefore, the ETM initialization is useful to avoid the over-load at startup of only some grid nodes. To initialize the ETM we run the MicroClAn framework several times with the OLB approach until all the cells of the matrices assumed a value.

The grid environment employed to evaluate the MicroClAn framework on the aforementioned scenarios, was composed by four grid nodes, named in the following A, B, C and D. Each grid node is an Intel Xeon processor with 4 GB of RAM. Several metatasks composed by eight tasks were defined to evaluate the scheduling strategies. We collected the execution times of ten runs of each experiment. In order to exploit the approaches with arbitrary scheduling avoid result biases due to the task order, we sorted randomly the list of the tasks. In Table 8 we report the average execution time for each scenario when the ETM is initialized. In the following we discuss in more details the performance of each scheduling strategy according to the ETM characteristics.

ETM linear and homogeneous. Since this scenario models the system runs regularly, the scheduler has all the information of the average execution times for each task on each node. Thus, all the methods have a comparable average execution time except for the MET method which achieve the worst performance.

ETM linear and heterogeneous. When the tasks are heterogeneous in terms of execution times (i.e., high variance) but the nodes are linear, the performance in terms of average execution time are similar behavior for most of the approaches. Moreover, this scenario favorites the scheduling approaches which assign the most expensive task in terms of execution time to the fastest nodes. Indeed, the Max-min method achieves the best performance, while the MET, which is the worst approach, has a more relevant difference with respect to the other ones due to the heterogeneity of the tasks.
ETM not linear and homogeneous. In this scenario, the tasks have similar execution time but nodes are not linear, i.e., some tasks are faster on some nodes and slower on some other nodes. This kind of scenario favorites the approaches based on the minimum execution time. All the approaches present similar performance to the previous scenario. Moreover, the MET results the worst approach as in the aforementioned cases.
ETM not linear and heterogeneous. This scenario is the most similar to a real case in which the user wants perform a comparison among different clustering methods. Indeed, clustering algorithms and quality indexes have different complexities and the nodes usually have different performance according to each task. In this case all the approaches have similar performance. Since this scenario models the behavior of grid nodes with similar resources working on heterogeneous tasks, the assignment of a task to the fastest node means assign it to the free node. In this case, MET presents a similar behavior in task assignment with respect to the other approaches. The Max-min approach shows the best performance because the expected values stored in the matrix are the most stable.

Due to the heterogeneity of the clustering approaches and quality indexes in terms of complexity, MicroClAn framework adopts the Max-min approach which in many cases achieves the best performance. However, to initialize the ETM the OLB strategy was exploited in the first runs to guarantee good performance in term of completion time of the metatasks.

7. Conclusions

Microarray clustering gave rise to a lot of attention in recent years in the scientific community. Many tools and algorithms were proposed to address microarray clustering and clustering evaluation, while less attention has been devoted to the definition of a procedure to compare and aggregate clustering evaluations produced by quality indexes to select the best one. Moreover, when few computational resources are available to perform a large set of clustering analysis, the completion time of these tasks become a challenging task.

To address these issues, in this paper we proposed the MicroClAn framework, which evaluates clustering algorithms by means of internal, external and biological indexes and compares results by using a two-step aggregation procedure based on Kemeny and Condorcet theories. A distributed environment has been exploited to optimize the execution time and several scheduling strategies were evaluated.

Experimental evaluations on synthetic and real microarray datasets show the effectiveness and the efficiency of MicroClAn framework. The two-step aggregation procedure allows to evaluate separately the quality indexes belonging to different categories and assign more relevance to one aspect (e.g., biological quality) during the evaluation process. Moreover, the clustering analysis takes advantage in term of computational time exploiting a distributed environment and the most suitable scheduling procedure.

Future works will be addressed to include different scheduling procedures which consider also the features of the grid nodes and the network traffic. Moreover, we will study new techniques to improve the performance of some quality indexes by means of parallel algorithms. Finally, since our framework is modular, we plan to include new quality indexes and clustering algorithms which can be exploited for further analyses also in other contexts (e.g., geographical, financial, social networks). Interested readers are encouraged to try the system, which is available upon request.

Acknowledgments

The authors would like to thank Prof. Elena Baralis (Politecnico di Torino) for her support and fruitful discussions and Dr. Floriana Riccio for implementing parts of the MicroClAn framework.

References

[1] Au, W., Chan, K., Wong, A., Wang, Y., 2005. Attribute clustering for grouping, selection, and classification of gene expression data. IEEE/ACM transactions on computational biology and bioinformatics, 83–101.

[2] Bandyopadhyay, S., Mukhopadhyay, A., Maulik, U., 2007. An Improved Algorithm for Clustering Gene Expression Data. BMC Bioinformatics 23 (21), 2859–2865.

[3] Baralis, E., Bruno, G., Fiori, A., 2011. Measuring gene similarity by means of the classification distance. Knowledge and Information Systems, 1–21.

[4] Bonifati, A., Cuzzocrea, A., 2007. Efficient fragmentation of large xml documents. In: Database and Expert Systems Applications. pp. 539–550.

[5] Braun, T., Siegel, H., Beck, N., Boloni, L., Maheswaran, M., Reuther, A., Robertson, J., Theys, M., Yao, B., Hensgen, D., et al., 2001. A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems* 1. Journal of Parallel and Distributed computing 61 (6), 810–837.

[6] Brock, G., Pihur, V., Datta, S., Datta, S., 2008. clvalid: An r package for clustaring validation. Journal of Statistical Software 25 (4), 3201–3212.

[7] Bruno, G., Fiori, A., 2010. Microarray data mining: issues and prospect. IGI-Global, pp. 23–47.

[8] Chen, W., Song, Y., Bai, H., Lin, C., Chang, E., 2010. Parallel spectral clustering in distributed systems. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[9] Chen, W., Zhang, J., 2009. An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 39 (1), 29–43.

[10] Cooke, E. J., Savage, R. S., Kirk, P. D., Darkins, R., Wild, D. L., 2011. Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements. BMC Bioinformatics 12, 399.

[11] Cuzzocrea, A., Furfaro, F., Saccà, D., 2009. Enabling olap in mobile environments via intelligent data cube compression techniques. Journal of Intelligent Information Systems 33 (2), 95–143.

[12] D. Jiang, C. T., Zhang, A., 2004. Cluster analysis for gene-expression data: a survey. IEEE Trans. Knowledge Data Eng. 16 (11), 1370–1386.

[13] Datta, S., Datta, S., 2006. Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. BMC bioinformatics 7 (1), 397.

[14] Davies, D., Bouldin, D., 1979. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (2), 224–227.

[15] DeConde, R., Hawley, S., Falcon, S., Clegg, N., Knudsen, B., Etzioni, R., 2006. Combining results of microarray experiments: a rank aggregation approach. Statistical Applications in Genetics and Molecular Biology 5 (1), 1204.

[16] Dembélé, D., Kastner, P., 2003. Fuzzy C-means method for Clustering Microarray Data. Bioinformatics 19 (8), 973–980.

[17] Dwork, C., Kumar, R., Naor, M., Sivakumar, D., 2001. Rank aggregation methods for the web. In: Proceedings of the 10th international conference on World Wide Web. ACM, pp. 613–622.

[18] Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., et al., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286 (5439), 531.

[19] Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2001. On clustering validation techniques. Journal of Intelligent Information Systems 17 (2), 107–145.

[20] Handl, J., Knowles, J., Kell, D., 2005. Computational cluster validation in post-genomic data analysis. Bioinformatics 21 (15), 3201.

[21] Jiang, D., Pei, J., Ramanathan, M., Tang, C., Zhang, A., 2004. Mining coherent gene clusters from gene-sample-time microarray data. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM New York, NY, USA, pp. 430–439.

[22] Jiang, D., Pei, J., Zhang, A., 2003. DHC: A density-based hierarchical clustering method for time series gene expression data. In: Proceedings of the 3rd IEEE Symposium on BioInformatics and BioEngineering. IEEE Computer Society Washington, DC, USA, p. 393.

[23] Jiang, D., Tang, C., Zhang, A., 2004. Cluster analysis for gene expression data: A survey. IEEE Transactions on Knowledge and Data Engineering 16 (11), 1370–1386.

[24] Karypis, G., Han, E., Kumar, V., 1999. Chameleon: Hierarchical clustering using dynamic modeling. Computer 32 (8), 68–75.

[25] Kerr, G., Ruskin, H., Crane, M., Doolan, P., 2008. Techniques for clustering gene expression data. Computers in Biology and Medicine 38, 283–293.

[26] Khalifa, A., Fergany, T., Ammar, R., Tolba, M., ???? Dynamic On-Line Allocation of Independent Task onto Heterogeneous Computing Systems to Maximize Load Balancing. In: Signal Processing and Information Technology, 2008. ISSPIT 2008. IEEE International Symposium on. IEEE, pp. 418–425.

[27] Krishna, R., Li, C. T., Buchanan-Wollaston, V., 2010. A temporal precedence based clustering method for gene expression microarray data. BMC Bioinformatics 11, 68.

[28] Larsen, B., Aone, C., 1999. Fast and effective text mining using linear-time document clustering. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM New York, NY, USA, pp. 16–22.

[29] Liu, C., Lee, C., Wang, L., 2007. Distributed clustering algorithms for data-gathering in wireless mobile sensor networks. Journal of Parallel and Distributed Computing 67 (11), 1187–1200.

[30] Liu, J., Wang, W., Yang, J., 2004. A framework for ontology-driven subspace clustering. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 623–628.
[31] Lockhart, D., Brown, E., Wong, G., Chee, M., Gingeras, T., Nov. 23 2004. Expression monitoring by hybridization to high density oligonucleotide arrays. US Patent App. 10/998,518.
[32] Madeira, S. C., Oliveira, A. L., 2004. Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans. Comput. Biol. Bioinformatics 1 (1), 24–45.
[33] Maulik, U., Mukhopadhyay, A., Bandyopadhyay, S., 2009. Combining Pareto-Optimal Clusters using Supervised Learning for Identifying Co-expressed Genes. BMC Bioinformatics 10.
[34] Pihur, V., Brock, G., Datta, S., 2009. Cluster Validation for Microarray Data: An Appraisal. Advances in Multivariate Statistical Methods, 79.
[35] Pihur, V., Datta, S., Datta, S., 2007. Weighted rank aggregation of cluster validation measures: a monte carlo cross-entropy approach. Bioinformatics 23 (13), 1607.
[36] Pihur, V., Datta, S., Datta, S., 2009. RankAggreg, an R package for weighted rank aggregation. BMC bioinformatics 10 (1), 62.
[37] Raczynski, L., Wozniak, K., Rubel, T., Zaremba, K., 2010. Application of Density Based Clustering to Microarray Data Analysis. Bioinformatics 56 (3), 281–286.
[38] Rand, W., 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association 66 (336), 846–850.
[39] Richards, A., Holmans, P., O'Donovan, M., Owen, M., Jones, L., 2008. A comparison of four clustering methods for brain expression microarray data. BMC bioinformatics 9 (1), 490.
[40] Robinson, M., Grigull, J., Mohammad, N., Hughes, T., 2002. Funspec: a web-based cluster interpreter for yeast. BMC Bioinformatics 3 (1), 35.
[41] Rousseeuw, P. J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Computational and Applied Mathematics 20, 53–65.
[42] Schena, M., Shalon, D., Davis, R., Brown, P., 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270 (5235), 467.
[43] Schmidberger, M., Mansmann, U., 2008. Parallelized preprocessing algorithms for high-density oligonucleotide array data. In: Proceedings of the 22th International Parallel and Distributed Processing Symposium.
[44] Sharan, R., Maron-Katz, A., Shamir, R., 2003. CLICK and EXPANDER: a system for clustering and visualizing gene expression data. Bioinformatics 19 (14), 1787–1799.
[45] Sokal, R., 1977. Clustering and classification: Background and current directions. Classification and Clustering 3.
[46] Tan, P., Steinbach, M., Kumar, V., 2005. Introduction to data mining. Pearson Addison Wesley Boston.
[47] Thompson, R., Deo, M., Turner, D., 2007. Analysis of microRNA expression by in situ hybridization with RNA oligonucleotide probes. Methods 43 (2), 153–161.
[48] Tobita, T., Kouda, M., Kasahara, H., 2002. Performance Evaluation of Minimum Execution Time Multiprocessor Scheduling Algorithms Using Standard Task Graph Set. Transactions 43 (4), 936–947.
[49] Topcuoglu, H., Hariri, S., Wu, M., 2002. Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE transactions on parallel and distributed systems, 260–274.
[50] Tseng, V., Kao, C.-P., 2005. Efficiently mining gene expression data via a novel parameterless clustering method. Computational Biology and Bioinformatics, IEEE/ACM Transactions on 2 (4), 355–365.
[51] Yeung, K., Haynor, D., Ruzzo, W., 2001. Validating clustering for gene expression data. Bioinformatics 17 (4), 309.
[52] Zhang, Y., Mueller, F., Cui, X., Potok, T., 2011. Data-intensive document clustering on graphics processing unit (GPU) clusters. Journal of Parallel and Distributed Computing 71 (2), 211–224.