Shared farthest neighbor approach to clustering of high dimensionality, low cardinality data

Stefano Rovetta, [1] Francesco Masulli

Department of Computer and Information Sciences and CNISM, University of Genova,
Via Dodecaneso 35 I-16146 Genova, Italy

Abstract

Clustering algorithms are routinely used in biomedical disciplines, and are a basic tool in bioinformatics. Depending on the task at hand, there are two most popular options, the central partitional techniques and the Agglomerative Hierarchical Clustering techniques and their derivatives. These methods are well studied and well established. However, both categories have some drawbacks related to data dimensionality (for partitional algorithms) and to the bottom-up structure (for hierarchical agglomerative algorithms). To overcome these limitations, motivated by the problem of gene expression analysis with DNA microarrays, we present a hierarchical clustering algorithm based on a completely different principle, which is the analysis of shared *farthest neighbors*. We present a framework for clustering using ranks and indexes, and introduce the Shared Farthest Neighbors clustering criterion. We illustrate the properties of the method and present experimental results on different data sets, using the strategy of evaluating data clustering by extrinsic knowledge given by class labels.

1 Introduction

Data clustering is a routine step in biological data analysis, and a basic tool in bioinformatics [1–4]. Depending on the task at hand, there are two most popular options, provided by several commercial or professional systems, like Cluster/Treeview [5], Agilent GeneSpring [2] , Data Mining Tool by Affymetrix [3] , often with the additional choice of one or two less common techniques.

[1] Corresponding author. E-mail: rovetta@disi.unige.it. Phone: +39 010 353 6636. Fax: +39 010 353 6699.

[2] http://www.chem.agilent.com/scripts/pds.asp?lpage=37147

[3] http://www.affymetrix.com/products/software/index.affx

When the data cardinality n is high and the dimensionality d is not very large, it is possible to use iterative, partitional algorithms such as k-Means [6] or one of its many variations (a frequent choice is Self Organizing Maps [7], which continues to be used as a clustering method). When data dimensionality is very large, or the number observations is comparatively small, then hierarchical agglomerative algorithms are normally used [5].

The set of available tools is often limited to these categories only, probably because they are available in widespread software and, in the case of hierarchical agglomerative clustering, they can be easily interpreted and give rise to visually appealing representations like dendrograms [8] or color diagrams [5]. As a matter of fact, also other techniques like Spectral Clustering [9, 10] seem quite adequate for tasks in bioinformatics [11].

However, both categories have some drawbacks related to data dimensionality (for partitional methods) and to the bottom-up structure (for hierarchical agglomerative methods) [12].

The objective of this paper is to introduce the Shared Farthest Neighbors clustering technique, a hierarchical clustering algorithm based on a novel agglomeration principle.

The proposed method is motivated by a typical unsupervised problem in bioinformatics: clustering of tissue profiles or cell lines in microarray analysis of gene expression. Its applicability and its properties will be illustrated on a selection of diverse problems in the biomedical disciplines.

The approach shares some similarities with Jarvis-Patrick clustering [13], which however is based on the analysis of shared *nearest neighbors* and is not a hierarchical method.

We apply a validation method which is not typical of methodological research in the general field of data clustering, but rather of the specific application area of bioinformatics and medicine: namely, we compare the clustering result to supervised information available for the problems.

This paper is structured as follows. Section 2 describes dimensionality issues in data clustering and how rank-based techniques can be applied. Section 3 introduces the Shared Farthest Neighbor technique and Section 4 reports on the experimental verification. The last section summarizes the results and draws conclusions about the presented method.

2 Properties of clustering techniques

2.1 Clustering problems in genomic data analysis

We are given a set of n experimental observations $X = \{x_1, x_2, \cdots, x_n\}$, where each observation comprises d observed variables $x_i = \{x_{i1}, x_{i2}, \ldots, x_{id}\}$. Suppose further that we are given a proximity criterion to evaluate the data, either in the form of a proximity function $\delta(x, y)$ (distance or similarity) or as a $d \times d$ proximity matrix D.

We are addressing low-cardinality, high-dimensionality tasks and we need to establish parameters to decide whether we are in this scenario. Deciding whether the cardinality of a given data set is large or small is a problem-dependent task [14–16]. However, we propose the following, arbitrary criterion: we examine the ratio $r = \log_2 n/d$ (obviously $n \geq 2$). We can assume that cardinality is (relatively) high when $r >> 1$. For r around or below unity, we are in the low-dimensional, high cardinality case.

Under these assumptions, it is easy to see that typical problems in tissue clustering with DNA microarray gene expression data fall in this category. With reference to the two problems described in the experimental part (see Section 4 for details), we have for the Leukemia problem, $d = 7192$ and $n = 72$ ($r = 0.86 \cdot 10^{-3}$), and for the Genoa lung cancer problem, $d = 1920$ and $n = 5$ ($r = 1.21 \cdot 10^{-3}$). As we can infer from numbers, gene clustering can be an easier problem from the standpoint of dimensionality, since d and n should be transposed.

2.2 A brief review of some clustering techniques

There is a vast literature about data clustering, and excellent reviews and introductions to the topic are provided in [6, 17, 18].

Partitional clustering methods [17–24] are usually based on centroids. They are especially suited to the case of small number of centroids and a sufficient number of data objects. The issue here is local minima. Available remedies include using fuzzy memberships [21, 25] and on-line optimization [26, 27]. In the general case of biomedical data, where observations are costly, and especially in microarray experiments, many variables are observed in relatively few experiments. This raises the issue of the curse of dimensionality [19, 28].

In some cases (with very low r) n is even less than d. The data span only a subspace of the data space. In these conditions, it is not even easy to define the concept of density. This makes k-Means type techniques typically adequate for clustering variables across experiments (e.g., gene clustering), rather than clustering experiments. There have been many efforts in solving the dimensionality problem for clustering [29]. Another drawback is related to the problem of model order selection (number of clusters).

With reference to hierarchical methods [30–32], divisive approaches [18] can generally exploit more global information in data with respect to agglomerative methods [17], and therefore yield better quality models. However bottom-up methods are generally more time-efficient.

The standard hierarchical approach does not require the selection of model order, simply because it makes no attempt at defining clusters. Cophenetic proximity matrix analysis [17], or agglomerative coefficients [18] are needed for an a-posteriori estimate.

Another, related problem is that the taxonomy obtained is not very stable. Usually this problem is tackled with resampling approaches [33,34], or simply by trying all possible combinations of parameters available in the specific software used [35].

Clusters based on distances also suffer from the nonintuitive fact [36] that, when space dimensionality is high or even moderate (as low as 10-15), the distance of a query point x_0 to its farthest neighbor x_{FN} and to its nearest neighbor x_{NN} tend to become statistically equal:

$$\lim_{d \to \infty} P\left\{\delta(x_0, x_{\mathrm{NN}}) = \delta(x_0, x_{\mathrm{FN}})\right\} = 1. \tag{1}$$

This causes the actual distance values, and the concept of “nearest neighbor” itself, to become less and less meaningful with growing dimensionality. This last observation has been described as the “boundary phenomenon” in [37]. See also [38].

Finally, agglomerative algorithms cannot produce a partial (rough) result, to be refined only if needed (“anytime” algorithms in the data mining jargon).

3 Shared farthest Neighbors: principle of operation, algorithm, and properties

3.1 Design goals for a rank-based clustering method

Based on the previous discussion, we summarize our main design goals (similar sets of design goals have been outlined for instance in [39]).

To avoid the model order selection problem, we should design a hierarchical method. Hierarchical techniques often provide easier interpretation.

At the same time, it should allow for more than two objects at any level in the hierarchy. Interpretation is even easier if the hierarchy depth is reduced by allowing for splits that are more than dichotomic.

The procedure will be divisive rather than agglomerative. In this way, the criterion used to divide each cluster into (possibly more than two) sub-clusters provides an indication of the “appropriate” number of clusters for that level in the hierarchy, although assessing that this number is the true number of natural clusters would typically require further analysis.

3.2 Use of the proximity matrix

When d is large but n is comparatively small it is known that clustering based on a proximity matrix [17] may be preferable. It is an efficient way to reduce the dimensionality of the working space from d to n, since a proximity matrix D can be interpreted as an embedding of a set of n data vectors in a space of dimension n. Kernel methods [40,41] are also a generalization of the concept of proximity. Studies [42] show how methods exploiting proximity information are able to perform better than generally expected. In some applications data may be directly available in the form of a proximity matrix [43].

The only assumptions we will make about the proximity function δ is that it is defined for all pairs of objects in X and it is reflexive ($\delta(x, x) = 0 \; \forall x \in X$).

3.3 The rank matrix and the index matrix

Proximity data may not be reliable. In microarray experiments, sources of error include contamination due to washing, imperfect hybridization, variations of hybridization level across different chips, noise in the optical acquisition, effects of normalization method and parameters. D may also contain non significant information due to arbitrary design choices or ambiguous data.

To increase robustness, we can map D into R^δ, the rank matrix induced by the proximity δ. The rank $\rho_\delta(x_i, x_j, X)$ is the position of object x_j in the list of all objects in X sorted by their proximity to x_i, $\delta(x_i, x_j)$. The matrix R^δ, a transformation of the proximity matrix, is a proximity matrix itself; it is therefore another possible embedding of the data set X. (From now on, since X and δ are given, we simplify the notation by writing $\rho_\delta(x_i, x_j, X) = \rho_{ij}$ and $R^\delta = R$.)

This change in measurement level, from metric to ordinal, induces a loss of information that may or may not be significant. Other examples of this technique include Spearman’s rank-correlation index r_s [44], Kendall’s correlation index τ

and coefficient of concordance W [45], and Goodman and Kruskal's γ association statistics [46].

R provides a new representation of data object x_i as the rank vector r_i, the i-th row of R:

$$r_i = [\rho_{i1}, \rho_{i2}, \ldots, \rho_{in}] \,. \tag{2}$$

The rank matrix R can thus be considered as a transformed data set $R = \{r_1, r_2, \ldots, r_n\}$; clustering will be based on grouping objects by similarity of their rank vectors r_i, and the specific clustering criterion depends on the definition of this new proximity measure between rank vectors. We also define the index matrix I listing, for each object, the inexes of all objects in order of distance:

$$I_{i\rho_{ij}} = n - j + 1, \tag{3}$$

In the first position we have the index of the point with maximum rank (the farthest point x_{FN}), in the $(n-1)$-th position the index of the nearest point x_{NN}, and in the n-th position the index i of the point x_i itself.

3.4 Techniques based on rank or index

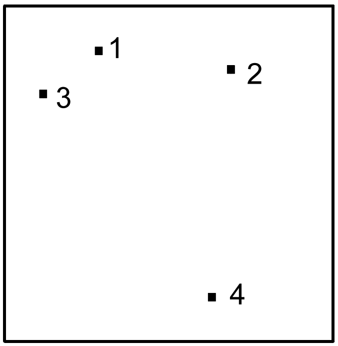
The matrices R and I convey essentially the same information; their use influences the algorithmic implementation of clustering methods, rather than the methods themselves. Usually rank-based methods are intended to be applied after a clustering has been obtained, as validity criteria, due to their computational weight. A clear example of this is Hubert's Γ statistics [47] [17] or Kendall's coefficient of concordance W as a cluster validity index [48], for which partitions of X should be exhaustively investigated.

Techniques based on the R mapping are presented in [49], for an iterative procedure, and in [50] for a rank-based hierarchical method. It should be noted that a simplified rank analysis is provided by methods based on nearest neighbors. An interesting related method is the Shared Near Neighbors (SNN) clustering by Jarvis and Patrick [13]. Here I is partitioned according to the following principle: the last k components of patterns I_i and I_j

$$\left\{I_{in}, I_{i(n-1)}, \ldots, I_{i(n-k+1)}\right\} \tag{4}$$

$$\left\{I_{jn}, I_{j(n-1)}, \ldots, I_{j(n-k+1)}\right\} \tag{5}$$

are compared, and x_i and x_j are in the same cluster if at least k_Q indexes are common to these two sub-patterns of length k. The parameters k and k_Q depend on the application and on the data, have to be selected by the user, and k_Q imposes a bias toward elongated ($k_Q \to 1$) or globular ($k_Q \to k$) cluster shapes.



Data points	1	2	3	4
I Neighbor	3	1	1	3
II Neighbor	2	3	2	2
III Neighbor	4	4	4	1

Fig. 1. An example data set to illustrate the “Points in perspective” principle. For each point the table lists the distance ranks of all other points.

The Jarvis-Patrick method is based on the consideration that points sharing the same near neighbors should belong to the same cluster. However this approach is not necessarily reliable for very sparse data.

The SNN produces the following odd result. The higher the rank of neighbors, the larger their “agglomerative” significance. Two points that are very close to each other and distant to other data points should be considered as a good cluster. But since the (first) nearest neighbor of either point is the other point, the first nearest neighbor is *always different*. This of course is not a major drawback (SNN simply counts $k > 1$ neighbors), but it offers some evidence that the principle itself may be only partially justified.

As a last remark, we recall that we are interested in a hierarchical method, and SNN provides only partitional clustering, although in the original presentation the authors suggest repeated applications of the method to obtain tree-structured clusters.

3.5 The “Points in Perspective” principle

We propose to adopt the following principle of operation: *Two points should be considered similar if they share the same farthest point among all remaining data.* We term this the “Points in Perspective” principle, since the points are examined not with reference to their neighborhood (locally), but with reference to far-away points in the data set, therefore in perspective. The example shown in Figure 1 clarifies the approach.

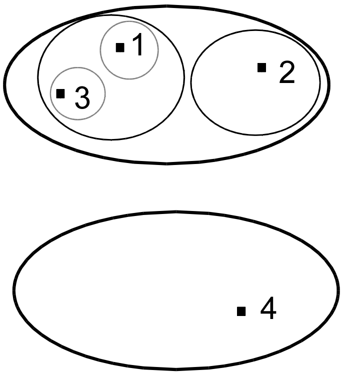


Fig. 2. The example data set clustered according to the proposed method.

The proposed "Points in Perspective" principle of operation yields a hierarchical clustering procedure, which proceeds as follows. First, D is computed or obtained as input. Then, R is computed from D and I from R.

All points sharing the same farthest point are in the same cluster of level 1. So, a cluster in the first level is defined as the set of points with the same value in the first column of I. In general, at level l, the k-th cluster is defined as

$$X_{lk} = \{x_i \in X | I_{il} = v_k\} \,. \qquad (6)$$

The procedure is recursively repeated until no further differentiation is found (all points within a level $l-1$ cluster share the same l-th farthest neighbor), or until a predefined maximum level is reached.

We term this technique *Shared Farthest Neighbor* clustering (SFN). The example shown in Figure 2 illustrated the result of applying the SFN procedure to the data of Figure 1.

3.6 Algorithm structure and complexity considerations

Here a proposed implementation of the SFN algorithm is sketched. The algorithm starts by computing the proximity matrix D. This is the phase where, if required, we can take care of missing data by adequately defining δ. From the time complexity standpoint, computation of the proximity matrix D is the most demanding part of the algorithm, requiring $k(d) \cdot O(n^2)$ time (and $O(n^2)$ space) for a proximity computation requiring $k(d)$ time for a pair of data objects. For instance, $k(d) = O(d)$ for a large number of proximities, including Euclidean distance and all Minkowski metrics, Hamming distance, cosine distance, and many others. The space required is $f \cdot n^2/2$ for symmetric proximities (metrics), $f \cdot n^2$ for the general case, where f is the storage space for a floating point value.

Once we have D, which may also be given as the input to the algorithm, we proceed as follows. For each point in the data set (a row of D) the distances to other data points are ordered and the corresponding rank is written in place of the actual distance, obtaining the rank matrix R.

Now each row of this matrix should be “inverted”, that is, cell contents should be swapped with the corresponding cell indexes to obtain the index matrix I listing, for each data point, all point labels in order of distance. This can be done simply looking up ranks in R and writing point labels in the corresponding position of X according to the definition given in Equation (3).

Clustering is now performed simply by sorting the rows of matrix I. This requires $O(n)$ additional space, either for row swapping or for an auxiliary index vector. Conceptually, this sorting is done according to each column, starting from the last (nearest neighbors) up to the first. This procedure may be performed in $O(n^2 \log n)$ time and requires a stable sorting algorithm. Stopping clustering at depth level n' implies starting from column n' rather than n, and a decrease in time complexity to $O(nn' \log n)$ (marginal, since presumably n' is proportional to n).

However, we can decrease the algorithm complexity as a function of data cardinality, and at the same time allow for a partial clustering, i.e. stopping before reaching a given level of the hierarchy if there is no further diversity in values. This can be obtained if we start sorting from the farthest neighbor, then partially sort the rows within each individual cluster, and so on. In the worst case the time needed is still $O(n^2 \log n)$, but on average it is probably $O(n \log^2 n)$ or better (depending on the data), while space requirements are at most the same as before. Any sorting algorithm may now be used, without requirement of stability.

If the above considerations about time and space complexity are applied to the actual numbers presented in the experimental part (Section 4), we can appreciate that the actual time and space requirements for real problems are not very demanding. In particular, as in any proximity matrix-based approach, there is no dependence on d after the proximity matrix has been computed. Therefore, from the standpoint of complexity, the technique becomes more and more appropriate as r is reduced, which corresponds for instance to microarray experiments on larger and larger sets of genes.

Pseudocode and a C language implementation are available at the web address http://mlsc.disi.unige.it/C/sfn/.

3.7 Properties of the proposed approach

In this section we highlight some features of the approach presented and of the resulting algorithm.

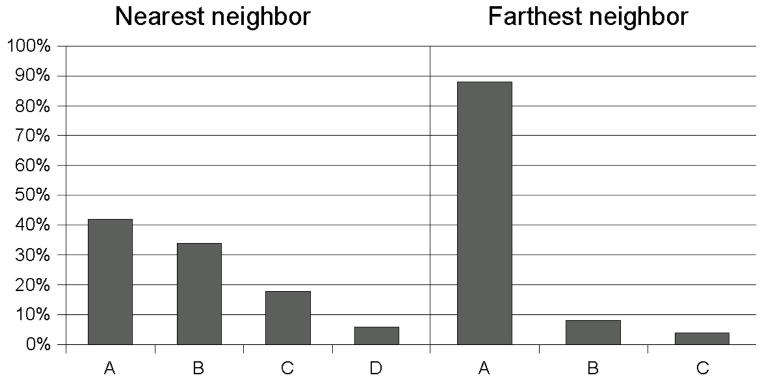


Fig. 3. Empirical stability analysis of the nearest- and farthest-neighbor criteria: histograms of obtained partitions on 50 random data sets.

The algorithm implemented according to the above description is of the "anytime" type, since it is divisive. We can decide to stop it when the hierarchy is partially built, and obtain a usable clustering result. Usually it is advisable to make use of this property, so that the result is more understandable (fewer larger clusters). It also makes little sense to split clusters into extremely small partitions when the data set is already scarce.

With respect to the position of points and to its perturbations, the hierarchy of dichotomies is more stable than in hierarchical agglomerative clustering algorithms. This is because clustering is based on the largest distances, over which the effect of small perturbations is usually negligible, rather than on the smallest. This is easily demonstrated by a simple experiments on a tiny problem with $d = 1$ and $n = 4$. The data set used is $\{2, 4, 6, 8\}$. Uniform noise in the interval $[-1, +1]$ was added to these points 50 times, and each time the partition resulting from applying the nearest neighbor and the farthest neighbor criteria have been evaluated. In all the experiments four different partitions were found with nearest neighbor and tree with farthest neighbor. We are not interested in the actual partitions, but only in the distribution of their frequency across the 50 random samplings. Therefore we have simply labeled the partitions with letters.

Figure 3 shows that the farthest neighbor criterion is much more stable, obtaining the same partition on 88% of the trials, whereas the nearest neighbor criterion does not go beyond 44% for the most frequent partition (the second most frequent is obtained 34% of the time).

Another feature of the SFN technique is the following. A cluster is not constrained to be separated in exactly two sub-clusters, and the clustering structure is therefore allowed to fit the natural structure of data (that can be non-dichotomic). According to this feature, SFN clustering is superior to agglomerative clustering. It is more similar to partitional clustering, although the ability to build a hierarchy is not found in standard partitional techniques.

After the proximity matrix D has been obtained, the algorithm operation (and computational complexity) is independent on data dimensionality. On the other hand, the dependence on the data cardinality (number of points) is not important, since by design we are in the case of small cardinality. Moreover, distances in the data space are used only for computing ranks and not for estimating densities or approximating region geometries.

An interesting property of the method is that very imbalanced clusters are possible. This is useful in the task of outlier detection. Due to the Points in Perspective principle, a point which is very far from other data will be put in a cluster on its own, since it will be the common farthest neighbor of all other points, and it will be the only one with a different farthest neighbor. Therefore, very imbalanced clusters at the top levels in the hierarchy are a signal of the presence of outliers.

The outlier detection property can be illustrated by looking again at Figure 1. Point 4 is clearly the farthest point for all other points in the data set. Accordingly, in the table, the last row (III Neighbor) provides a labeling that identifies 4 as an outlier, since it is the only one with a different label. Outlier analysis can therefore be based on the identification of a sufficiently imbalanced structure at the top level, with singletons or very small clusters along with other, reasonably sized clusters.

According to these features, the SFN technique is comparable to agglomerative hierarchical clustering, and therefore applicable to the same class of problems, especially tissue clustering in microarray experiments. However, in general, many bio-medical data analysis problems are characterized by low r, and the algorithm can be successfully applied.

4 Experimental validation

4.1 Experimental setup

We have validated the SFN algorithm on genomic data analysis and medical diagnosis problems, some of which are publicly available. The aim of the experiments is to demonstrate that the method performs comparatively well with respect to published results (we don’t aim at proving its superior performance, since this is not reasonable) while featuring the desirable properties that we have listed as design goals. The data sets used include the following.

The first problem is labeled *Genoa lung cancer*. It is included as a verification of consistency of the technique, since it is a real problem but has a very reduced number of instances. Five patients with lung cancer have been analyzed with a DNA microarray technique. These are preliminary results from an on-going study and are

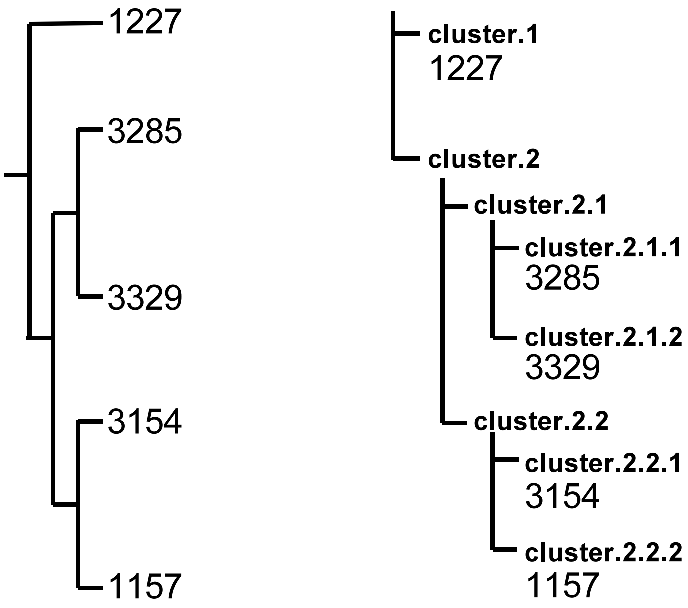


Fig. 4. Dendrogram obtained on the Genoa lung cancer problem by hierarchical agglomerative clustering and hierarchy obtained with the SFN algorithm.

not publicly available. Given the very small cardinality, these data have been used to validate the method against the results obtained with hierarchical agglomerative clustering. The problem has 1920 attributes and 5 instances ($r = 1.21 \cdot 10^{-3}$).

Please note that the Genoa lung cancer data are **not** the same as the Lung cancer data set available from the UCI repository.

We have applied the technique to a set of problems for which published results are available. They are described in the following, in order of growing dimensionality/cardinality ratio. The value of r is also expressed, and only in the first problem it is larger than 1.

(1) *Pima Indians diabetes* [51]. Pima Indians are affected by an endemic form of diabetes, which is found with much higher frequency than in other populations, and have agreed to be the subject of a study. The data collected have been put in the public access on the UCI repository of machine learning databases [52].

The problem has 768 instances, corresponding to patient descriptions, of which 500 classified as “Negative” and 268 as “Positive” by a test for diabetes. There are 8 numerical attributes ($r = 1.2$).

(2) *Wisconsin diagnostic breast cancer* (newer dataset) [53]. Samples of breast mass are microscopically analyzed. The data are obtained by digitizing an image from each sample. Features describe the cell nuclei present in the image. These data are from the UCI repository as above.

The problem has 30 attributes and 569 instances, of which 357 have been diagnosed as “Benign” and 212 as “Malignant” ($r = 0.3$).

(3) *Lyme disease* [54, 55]. A disease discovered in the relatively recent past. It has initial effects on skin, then it can reach the nervous system, heart, connective tissue (Lyme arthritis). In regions where it is not endemic, the diversity of signs can be confusing even to medical professionals trying to diagnose it, if they are not specifically trained. One of the authors has worked on this data set, which is currently not publicly available.

    The problem has 684 instances, corresponding to patient descriptions, of which 446 have been diagnosed as “Unaffected” and 238 as “Affected” by experts (according to criteria based on clinical and biological observations). Each instance has 54 numerical attributes ($r = 0.17$).

(4) *Molecular classification of leukemia* [2]. DNA microarray are used to characterize two forms of leukemia at the molecular level (acute lymphoblastic leukemia, labeled as “ALL”, and acute myeloid leukemia, labeled as “AML”) and within one of the two forms to separate two further sub-classes that are not distinguishable at the morphologic or serological level, but have dramatically different prognoses. There are a training set and a test set, both available from the web address `http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi`. The problem has 7192 attributes. The training set has 38 instances, of which 27 are classified as “ALL” and 11 as “AML” ($r = 0.73 \cdot 10^{-3}$), and the test set has 34 instances, of which 20 are “ALL” and 14 are “AML” ($r = 0.71 \cdot 10^{-3}$).

The last problem is included to test the performance of the technique on non-metric data. The problem is labeled *Splice-junction Gene Sequences* [56]. Splice junction sites are point in the genome where introns (non-coding sequences) and exons (coding sequences) are joined together. The task is to identify splicing sites. These data have been obtained from the UCI repository as above.

The problem has 60 categorical attributes, representing nucleotides in a DNA sequence, that can contain at its middle point an exon-intron boundary (labeled as “EI”), an intron-exon boundary (labeled as “IE”), or neither of the previous (labeled as “Neither”). There are 3190 instances, of which 767 are classified as “EI”, 768 as “IE”, and 1655 as “Neither”.

The first experiment consists in validating the clustering result on the Genoa lung cancer problem. This is to achieve a first indication that the clusters we get are reasonable. This problem has a very small data cardinality, so the number of possible clusterings is limited and, arguably, there is only one “correct” result.

Samples in the Genoa lung cancer dataset are individually identified by the following numeric labels: 1157, 1227, 3154, 3285, 3329.

Figure 4 show the dendrogram obtained with hierarchical agglomerative clustering (left) and the hierarchical tree obtained with the SFN algorithm (right). Proximity is defined as the correlation between data vectors. Note that a dendrogram retains proximity information in the length of the branches, while in the hierarchical tree on the right this information is not present. Labels on the SFN tree are formed by the prefix “cluster” followed by a list of numbers uniquely addressing a cluster within the hierarchy.

We obtain the same result in both cases. In the dendrogram on the left, leafs are sample labels. In the tree on the right, the cluster label of each leaf cluster (which in principle can contain more than one sample) is followed by the list of contained objects, in this case only one per cluster. By reading the dendrogram, we can examine the cluster containing samples 3285 and 3329 and the cluster containing 3154 and 1157. These two clusters are almost vertically aligned. This means that they are split at different hierarchical levels only on the basis of a difference that is probably non-significant. If the hierarchical agglomerative procedure were able to form clusters of more than 2 objects, these would probably be at the same level. On the other hand, SFN has this ability, so these two clusters are found at the same hierarchical level in the SFN tree on the right.

4.2 Evaluation of experimental results

To evaluate the quality of clustering, we adopt the approach of comparing the results to a “ground truth”. This is not a common approach in the general area of data clustering, but it is the standard way to proceed in the target application area of bioinformatics.

In general, the result of clustering is usually assessed on the basis of some external knowledge about how clusters should be structured. This may imply evaluating separation, density, connectedness, diameter, and so on. However, these are all evaluations of results against a given expectation, which may not translate into good performance when the method is applied to a problem [57]. More importantly, they allow clustering results to be validated against subjective hypotheses of the researcher. This should be avoided if at all possible. The problem is also discussed in a recent editorial of the present journal [58].

The only way to assess the usefulness of a clustering result is indirect validation, whereby clusters are applied to the solution of a problem and the correctness is evaluated against objective external knowledge. this procedure is defined by Jain and Dubes [17] as “validating clustering by extrinsic classification”, and has been followed in many other studies [2, 10, 39]. We feel that this approach is the only reasonable one if we don’t want to judge clustering results by some cluster validity index, which is nothing but a bias toward some preferred cluster property (e.g., compact, or well separated, or connected).

Therefore, to adopt this approach we need labeled data sets, where the external (extrinsic) knowledge is the class information provided by labels. The experiments are all performed on supervised problems.

We expect that, if the algorithm finds significant structures in the data, these will be reflected by the distribution of classes. Therefore we operate a “calibration” [7] step for clusters and compare them to the behavior of *supervised* methods from the literature.

The so-called calibration step consists in the following.

For each cluster j:

- Count the number of patterns of each class k (call it n_{jk}).
- Count the total number of patterns (call it N_j).
- Compute the proportion of patterns of each class (call it $p_{jk} = n_{jk}/N_j$).
- Assign to the cluster the label of the most represented class (c_k such that $k = \underset{k}{\operatorname{argmax}} \{p_{jk}\}$).

A cluster j for which $p_{jk} = 1$ for some k is usually termed a “pure” cluster, and a purity measure can be expressed as the percentage of elements of the assigned class in a cluster. During this procedure we can also obtain confidence estimates, on the basis of cluster cardinalities N_j. The experimental results are then expressed as the fraction of points falling in clusters which are labeled with a class different from that of the point. This quantity is expressed as a percentage and termed “error percentage” (indicated as “Error %” in the results).

Adopting this strategy, we cannot obtain a direct assessment of the goodness of clusters per se; in exchange, we obtain valuable information about how these clusters map on the natural structure of the problem, something that may be more interesting than evaluating a single or few indirect performance parameters.

Regarding the evaluation method, we choose not to perform cross-validation or similar procedures, considering that the algorithm is “trained” in a completely unsupervised manner, and calibration already occurs (in a sense) on an external validation data set, that is the set of class labels. Cross-validation or resampling methods, however, could be very useful to assess the stability of the proposed method, by comparing clustering structures in repeated experiments.

Table 1 lists the published results of machine learning techniques, available at the respective sources of the datasets.

The experimental results reported on Table 2 are obtained on problems *Pima diabetes*, *Wisconsin breast cancer*, *Lyme disease*, and *Leukemia*, all with Euclidean distance.

Table 3 and Figure 6 show results for the splicing junction sites problem.

Table 1
Results from the literature for the experimental problems (supervised methods)

Problem	**Error %**
Pima diabetes	24%
Wisconsin breast cancer	0%
Lyme disease	7.2%
Leukemia (training set)	0%
Splice-junction sites	6.3%

Table 2
Experimental results on problems *Pima*, *Breast cancer*, *Lyme*, *Leukemia*

Problem	**Preprocessing**	**Error %**
Pima	Normalized with respect to average/stdev	12.40%
Wisconsin	Normalized with respect to average/2*stdev	5.60%
Lyme	Normalized with respect to average/2*stdev	6.00%
Leukemia (training set, $n = 38$)	None	0.00%
Leukemia (training+test sets, $n = 72$)	None	6.90%

Table 3
Performance on the *Splicing-junction sites* problem.

Label	**Cardinality**	**Class**	**Purity**
Cluster.1	2495	Splicing	63.8%
Cluster.1.1	902	Non-Splicing	100.0%
Cluster.1.2	1593	Splicing	100.0%
Cluster.2	695	Non-Splicing	100.0%

4.3 Comments to the results

The results we achieve may be compared with those obtained by supervised approaches proposed in the literature (see Table 1). We may observe that the results are generally similar, although usually better, with the single exception of the Wisconsin diagnostic breast cancer problem for which a perfect classification was not achieved.

This should be a confirmation of the validity of the method. Since clustering is done

Table 4
Details of clusters for the Leukemia problem

Cardinality	Clusters	Class
10	1	AML
5	1	ALL
4	1	ALL
2	5	ALL
1	4	ALL

in a completely unsupervised manner, finding that the cluster structure is reasonably mapped onto the true classes supports the hypothesis that the algorithm is capable of discovering the “true” structure, the one that is inherent in data.

In particular, the results on the Leukemia dataset show that the method compares favorably with the approach by Golub et al. [2]. For instance, when comparing unsupervised methods, performance on the training set of 38 samples is errorless in our case, whereas the original Self-Organizing Map (SOM) approach yielded 4 misclassified samples.

It is not easy to compare the deeper trees obtained by standard agglomerative hierarchical clustering to those obtained with the proposed method, that may be much shallower and still convey significant structure, since they have no constraint on the number of sub-clusters. In the case of Leukemia data, the tree depth for standard hierarchical clustering is at least 6 (for instance, with the average linkage method we obtain a tree depth of 9). For SFN, splitting stopped at level 4, although only 1 cluster was split up to the fourth level, whereas 11 clusters with no further substructure were present at level 1. Calibration itself is not a well-defined process for a binary tree, since the structure of clusters is not related to the depth of the tree, but rather to the linkage value. The tree should therefore be cut to a given linkage value before assigning class labels and computing performance indexes (e.g. cluster purity). As already noted, to find this value we need to use criteria that are, at least to a certain extent, arbitrary.

We can comment further on the clusters obtained by taking also into account the class labels, that are “ALL” for 27 acute lymphoblastic leukemia patients and “AML” for 11 acute myeloid leukemia. The distribution of cardinality among the clusters at level 1 is as detailed in Table 4.

To allow for a comparison with the originally presented results (obtained with a SOM, therefore non-hierarchical), we have also plotted the top-level clusters using the same conventions as in Reference [2]. See Figure 5.

All leaf clusters (those which are not further split) are pure, that is, homogeneous

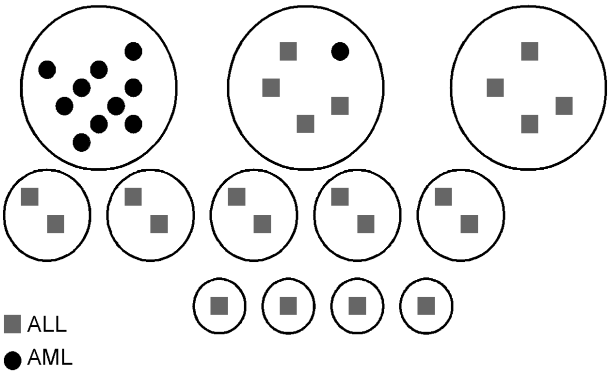


Fig. 5. Top-level clusters obtained for the Leukemia problem, presented with the same conventions as in Reference [2]. Note that the first cluster, that is not homogeneous, is further split into homogeneous sub-clusters (not shown).

with respect to the diagnosis. The single cluster having deeper structure has cardinality 5 and contains one data object of class AML; its purity level as defined in Subsection 4.2 is therefore 80%. (Note that its sub-clusters at the leaf level are all pure.) All other AML are in the largest level 1 cluster, the one with cardinality 10.

This suggests a structure in data whereby AML profiles are better characterized than ALL profiles. This is clearly true when we notice that there are two sub-classes of ALL, which are T-cell ALL and B-cell ALL.

The distribution in general is well represented by a partitional clustering (this is a confirmation of the already good result obtained by Golub et al. with the SOM approach), however there is a subset of the data that needs a deeper structure for adequate representation. After the calibration step, we see that this subset contains a sample diagnosed as AML that is correctly separated from the other samples. Cluster structure is again confirmed by the class labels.

The splice-junction sites problem is of a different nature, in that it involves non-metric data, i.e., strings of DNA sequences, 60 bases long and centered around the candidate splicing site. We use the (generalized) Hamming distance, defined as the number of mismatches between bases in corresponding positions (only the 40 central bases have been considered). We also simplify the problem by discriminating splicing/non-splicing sites, without distinction between EI and IE boundaries, obtaining a dichotomic classification problem. However this does not affect our ability to compare the results with those from the literature, since these are reported with error percentages class by class, and therefore it is possible to aggregate them.

Here the result is very good: Figure 6 illustrates the hierarchy obtained (graphics from a program by the authors). Fixing the maximum level at 2, the structure is very simple, with a cluster further split into two sub-clusters and another cluster

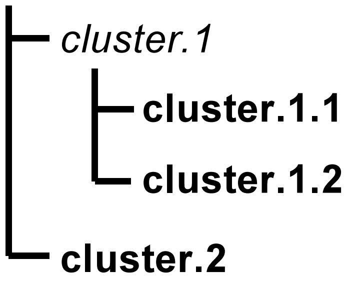


Fig. 6. Hierarchy obtained on the splice-junction sites problem. The cluster in slanted font is further split into sub-clusters.

without sub-clusters. Clusters at the deepest level (leaf clusters) are all pure, and the resulting classification, after performing the calibration step, is errorless, as indicated in the figure.

These data should be compared to results of other methods. The results reported in the accompanying documentation to the data set are all from supervised techniques. No supervised method is reported as capable of errorless performance.

Comparison with centroid-based clustering methods (k-Means) is not possible, since a proper centroid (barycenter) is not obtainable from non-metric data. It is also difficult to compare the obtained tree to that given by the standard agglomerative hierarchical methods, since, in contrast to the Genoa lung cancer problem, here the cardinality is high as an absolute value (although still very low when related to the dimensionality). Trees obtained with these methods will be much deeper; they may or may not be comparable to the one presented, and, if so, only after extracting significant clusters by pruning the tree at an appropriate level, as already indicated.

5 Conclusion

The clustering algorithm presented here is based on a novel principle of operation, and as such has properties not found in other more commonly used methods. With respect to standard hierarchical agglomerative clustering methods:

- top-down, rather than bottom-up operation; hence the ability of stopping clustering at a given level in the hierarchy;
- clusters are non-dichotomic, so that the resulting tree depth may be much lower.

With respect to partitional methods:

- it is not centroid-based;
- complexity is independent of data dimensionality;
- local anomalies found in the Shared Near Neighbors approach, and coped with by setting user parameters, here are not present due the operation based on far-

thest points.

It is especially designed for the analysis of data sets with high dimensionality-to-cardinality ratio, and is therefore well suited to DNA microarray data analysis, as demonstrated by the experiments. However it is more generally applicable in the field of biomedical data analysis, where these conditions are often met, and this was also experimentally shown in the present work.

We have observed that, similarly to the Jarvis-Patrick algorithm, the method presented may yield small or singleton clusters. This happens especially when data cardinality grows. Future developments include criteria for controlling the proliferation of singletons (cluster validity), but also applications of this property to outlier detection.

Acknowledgment

Work funded by the Italian Ministry of Education, University and Research (2004 “Research Projects of Major National Interest”, code 2004062740).

References

[1] F. Azuaje, Clustering-based approaches to discovering and visualising microarray data patterns, Briefings in Bioinformatics 4 (1) (2003) 31–42.

[2] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, E. Lander, Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.

[3] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitarewan, E. Dmitrovsky, E. S. Lander, T. R. Golub, Interpreting patterns of gene expression with self-organizion maps: Methods and application to hematopoietic differentiation, Proceedings of the National Academy of Science USA 96 (1999) 2907–2912.

[4] R. Homayouni, K. Heinrich, L. Wei, M. W. Berry, Gene clustering by latent semantic indexing of MEDLINE abstracts, Bioinformatics 21 (1) (2005) 104–115.

[5] M. B. Eisen, P. T. Spellman, P. O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, Proceedings of the National Academy of Sciences 95 (25) (1998) 14863–14868.

[6] A. Jain, M. Murty, P. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.

[7] T. Kohonen, Self-Organizing Maps, Springer, 2001.

[8] R. R. Sokal, P. H. Sneath, Principles of Numerical Taxonomy, Freeman, San Francisco, USA, 1963.

[9] R. Kannan, S. Vempala, A. Vetta, On clusterings: Good, bad and spectral, Journal of the ACM 51 (3) (2004) 497–515.
URL `http://doi.acm.org/10.1145/990308.990313`

[10] A. Paccanaro, J. A. Casbon, M. A. S. Saqi, Spectral clustering of protein sequences, Nucleic Acids Research 34 (5) (2006) 1571–1580.

[11] N. Arshadi, I. Jurisica, Data mining for case-based reasoning in high-dimensional biological domains, IEEE Transactions on Knowledge and Data Engineering 17 (8) (2005) 1127–1137.

[12] J. Qin, D. P. Lewis, W. Stafford Noble, Kernel hierarchical gene clustering from microarray expression data, Bioinformatics 19 (16) (2003) 2097–2104.
URL
`http://bioinformatics.oxfordjournals.org/cgi/content/abstract/19/16/2097`

[13] R. A. Jarvis, E. A. Patrick, Clustering using a similarity measure based on shared near neighbors, IEEE Transactions on Computers C22 (1973) 1025–1034.

[14] T. Cover, Geometrical and statistical properties of inequalities with application in pattern recognition, IEEE Trans. Electron. Comput. 14 (1965) 326–334.

[15] L. G. Valiant, A theory of the learnable, Communications of the ACM 27 (1984) 1134–1142.

[16] V. N. Vapnik, Statistical learning theory, Wiley, New York, 1998.

[17] A. K. Jain, R. C. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1988.

[18] L. Kaufman, P. J. Rousseeuw, Finding Groups in Data, John Wiley & Sons, New York, USA, 1990.

[19] R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, New York (USA), 1973.

[20] J. D. Banfield, A. E. Raftery, Model-based gaussian and non-gaussian clustering, Biometrics 49 (1993) 803–821.

[21] J. C. Bezdek, Pattern recognition with fuzzy objective function algorithms, Plenum, New York, 1981.

[22] E. E. Gustafson, W. C. Kessel, Fuzzy clustering with a covariance matrix, in: Proc. IEEE Conf. Decision Contr., San Diego, USA, 1979, pp. 761–766.

[23] J. C. Bezdek, Detection and characterization of cluster substructure, II: fuzzy c-varieties and convex combinations thereof, SIAM J. Appl. Math. 40 (2) (1981) 358–372.

[24] R. Krishnapuram, H. Frigui, O. Nasraoui, Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation – part ii, IEEE Transactions on Fuzzy Systems 3 (1) (1995) 44–60.

[25] K. Rose, E. Gurewitz, G. Fox, A deterministic annealing approach to clustering, Pattern Recognition Letters 11 (1990) 589–594.

[26] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: L. L. Cam, J. Neyman (Eds.), Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. I, University of California, 1967, pp. 281–297.

[27] T. Martinetz, S. Berkovich, K. Schulten, 'Neural gas' network for vector quantization and its application to time-series prediction, IEEE Transactions on Neural Networks 4 (4) (1993) 558–569.

[28] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.

[29] C. C. Aggarwal, P. S. Yu, Redefining clustering for high-dimensional applications, IEEE Transactions on Knowledge and Data Engineering 14 (2) (2002) 210–225.

[30] R. R. Sokal, C. D. Michener, A statistical method for evaluating systematic relationships, University of Kansas Science Bulletin 38 (1958) 1409–1438.

[31] S. C. Johnson, Hierarchical clustering schemes, Psychometrika 2 (1967) 241–254.

[32] J. H. Ward, Hierarchical grouping to optimize an objective function, Journal of American Statistical Association 58 (301) (1963) 236–244.

[33] J. Felsenstein, Confidence limits on phylogenies: An approach using the bootstrap, Evolution 39 (1985) 783–791.

[34] S. Monti, P. Tamayo, J. Mesirov, T. Golub, Consensus clustering: A resampling-based method for class discovery andvisualization of gene expression microarray data, Machine Learning 52 (1-2) (2003) 91–118.
URL
`http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1023/A:102`

[35] E. Kuchinskaya, M. Heyman, D. Grander, M. Linderholm, S. Soderhall, A. Zaritskey, A. Nordgren, A. Porwit-MacDonald, E. Zueva, Y. Pawitan, M. Corcoran, M. Nordenskjold, E. Blennow, Children and adults with acute lymphoblastic leukaemia have similar gene expression profiles, European Journal of Haematology 6 (74) (2005) 466–480.

[36] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, When is nearest neighbor meaningful?, in: 7th International Conference on Database Theory Proceedings (ICDT'99), Springer-Verlag, 1999, pp. 217–235.

[37] M. Jirina, M. j. Jirina, Boundary phenomenon in multivariate data, in: Proceedings of the IEEE Applied Electronics 2004, 2004, pp. 97–100.

[38] A. L. N. Fred, J. M. N. Leitao, A new cluster isolation criterion based on dissimilarity increments, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (8) (2003) 944–958.
URL `http://doi.ieeecomputersociety.org/10.1109/`

[39] G. Getz, E. Levine, E. Domany, Coupled two-way clustering analysis of gene microarray data, Proceedings of the National Academy of Science USA 97 (22) (2000) 12079–12084.

[40] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.

[41] E. Pękalska, P. Paclík, R. P. W. Duin, A generalized kernel approach to dissimilarity-based classification, Journal of Machine Learning Research 2 (2001) 175–211.

[42] F. Korn, B.-U. Pagel, C. Faloutsos, On the “dimensionality curse” and the “self-similarity blessing”, IEEE Transactions on Knowledge and Data Engineering 13 (1) (2001) 96–111.

[43] E. Pękalska, Dissimilarity representations in pattern recognition. concepts, theory and applications, Ph.D. thesis, Delft University of Technology (2005).

[44] C. Spearman, ‘General intelligence,’ objectively determined and measured, American Journal of Psychology 15 (1904) 201–293.

[45] M. Kendall, J. D. Gibbons, Rank Correlation Methods, 5th Edition, Oxford University Press, Oxford (UK), 1990.

[46] L. A. Goodman, W. H. Kruskal, Measures of association for cross-classifications, Journal of the American Statistical Association 49 (1954) 732–764.

[47] L. J. Hubert, J. Schultz, Quadratic assignment as a general data-analysis strategy, British Journal of Mathematical and Statistical Psychology 29 (1976) 190–241.

[48] R. Baumgartner, R. Somorjai, R. Summers, W. Richter, Assessment of cluster homogeneity in fMRI data using Kendall’s coefficient of concordance, Magnetic Resonance Imaging 17 (10) (1999) 1525–1532.

[49] S. Perrey, H. Brinck, A. Zielesny, Iterative rank based methods for clustering, in: R. Kasturi, D. Laurendeau, C. Suen (Eds.), Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB’03), IEEE Computer Society Press, Los Alamitos, USA, 2003, pp. 478–479.
URL
http://doi.ieeecomputersociety.org/10.1109/CSB.2003.1227379

[50] C. Devauchelle, A. W. M. Dress, A. Grossmann, S. Grünewald, A. Henaut, Constructing hierarchical set systems, Annals of Combinatorics 8 (2004) 441–456.
URL
http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s0002

[51] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, R. S. Johannes, Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, in: Proceedings of the Symposium on Computer Applications and Medical Care, Computer Society Press, 1988, pp. 261–265.

[52] C. Blake, C. Merz, UCI repository of machine learning databases, URL: http://www.ics.uci.edu/∼mlearn/MLRepository.html (1998).
URL http://www.ics.uci.edu/∼mlearn/MLRepository.html

[53] W. N. Street, W. H. Wolberg, O. L. Mangasarian, Nuclear feature extraction for breast tumor diagnosis, in: IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, Vol. 1905, 1993, pp. 861–870.

[54] C. Moneta, G. Parodi, S. Rovetta, R. Zunino, Automated diagnosis and disease characterization using neural network analysis, in: Proceedings of the 1992 IEEE International Conference on Systems, Man and Cybernetics - Chicago, IL, USA, 1992, pp. 123–128.

[55] G. Bianchi, L. Buffrini, P. Monteforte, G. Rovetta, S. Rovetta, R. Zunino, Neural approaches to the diagnosis and characterization of lyme disease, in: Proceedings of the 7th IEEE Symposium on Computer-Based Medical Systems, Winston-Salem, NC, 1994, pp. 194–199.

[56] M. O. Noordewier, G. G. Towell, J. W. Shavlik, Training knowledge-based neural networks to recognize genes in DNA sequences, in: Advances in Neural Information Processing Systems III, Vol. 3, Morgan Kaufmann, 1991, pp. 530–536.

[57] D. B. Allison, X. Cui, G. P. Page, M. Sabripour, Microarray data analysis: from disarray to consolidation and consensus, Nature Reviews Genetics 7 (1) (2006) 55–65.
URL http://dx.doi.org/10.1038/nrg1749

[58] E. R. Dougherty, The fundamental role of pattern recognition for gene-expression/microarray data in bioinformatics, Pattern Recognition 38 (12) (2005) 2226–2228.