An Evolutionary and Visual Framework for Clustering of DNA Microarray Data

José A. Castellanos-Garzón1,∗ and Fernando Díaz1

1University of Valladolid, Department of Computer Science, Plaza Alto de los Leones, 1 - 40005 Segovia, Spain

Summary

This paper presents a case study to show the competence of our evolutionary and visual framework for cluster analysis of DNA microarray data. The proposed framework joins a genetic algorithm for hierarchical clustering with a set of visual components of cluster tasks given by a tool. The cluster visualization tool allows us to display different views of clustering results as a means of cluster visual validation. The results of the genetic algorithm for clustering have shown that it can find better solutions than the other methods for the selected data set. Thus, this shows the reliability of the proposed framework.

1 Introduction

The study of gene expression data from DNA microarrays is of great interest for Bioinformatics (and functional genomics), because it allows us to analyze expression levels in hundreds of thousands of genes in a living organism sample. This feature makes gene expression analysis a fundamental tool of research for human health. It provides identification of new genes that are key in the genesis and development of diseases. However, the exploration of these large data sets is an important but difficult problem. The use of evolutionary and visual analytics techniques can help to cope with this problem. Visual data exploration has high potential and many applications in data mining use information visualization technology for an improved data analysis.

Classifying DNA microarray data according to their similarity degree is one of the main goals of data mining applied to this domain. The organization of objects in affinity groups is one way of knowledge discovery, being a key factor in machine learning [1, 2]. Moreover, the application of genetic algorithms [3, 4] to data mining is still of great importance in classification problems. Particularly, we can highlight, the use of evolutionary strategies to unsupervised classification (cluster analysis) in the knowledge discovery process [5]. Cluster analysis is one component of exploratory data analysis, which means sifting through data to make sense out of measurements by whatever means are available, whereas genetic algorithms (GAs) are blind search methods, inspired by the natural selection mechanism and oriented to solve complex optimization problems. Therefore, the application of evolutionary, cluster analysis and visualization techniques to DNA microarrays can turn biological data into knowledge of biological systems, often requiring further experimentation from initial data [6].

According to all previously explained, this paper presents a clustering framework joining a GA with a visualization tool, both applied to cluster analysis of DNA microarray data. The goal of

∗To whom correspondence should be addressed. Email: jantonio_cu@ieee.org

this paper is to firstly show the research in which we have been working [7, 8] and secondly, to show the results of such a framework (particularly, for our GA-based hierarchical clustering method) for a new case study, showing the reliability of our proposal.

2 An Evolutionary and Visual Framework

This section presents details on the our framework oriented to clustering. Firstly, we have presented a general scheme of the proposed framework, where the main components coupled to it have been explained. Secondly, a GA-based hierarchical clustering method has been given and after that, a tool for visual cluster analysis has also been given by showing its functionalities on DNA microarray data and finally, details on the implementation of the framework are outlined. The clustering method and visualization tool are integrated and linked within our global framework.

Then, the goal of proposed framework is to create a mechanism where the results of the clustering methods can be evaluated and verified through visualizations provided by a tool. In this sense, the visualization components given in the tool will also be useful to evaluate the performance of the cluster validity measures applied to the clustering results. That is, clustering methods (including our method), cluster validity measures and visual components of cluster exploration have been integrated to improve the global process of cluster analysis for DNA microarray data.

Figure 1 shows a general scheme of our evolutionary and visual framework, where Figure 1-a represents the evolutionary model of clustering that setting its parameters, we obtain a specific clustering method on the data repository in Figure 1-c. This way, parameters and results are validated using the visual model in Figure 1-b. Moreover, both models in this figure are based on a well-known knowledge source. Note that our evolutionary framework has been extended to analyze any other hierarchical clustering method different from our evolutionary method in Figure 1-a.

2.1 A Genetic Algorithm for Hierarchical Clustering

A hierarchical clustering method is a procedure for transforming a proximity matrix (matrix of distances between the objects to be grouped) into a sequence of nested partitions (clusterings), and a clustering is a type of classification imposed on a finite data set [1]. Since the hierarchical clustering is very important in biological data analysis due to its graphic representation of the data in form of dendrogram, which provides knowledge of the data domain, we have considered dendrograms as the individuals of our GA for hierarchical clustering.

Based then on the above, individuals (chromosomes) in our GA have been encoded as shown in Figure 2 (on the left side) for objects \( \{x_1, x_2, x_3, x_4, x_5\} \). Thus, individuals are built in an agglomerative way and each level is represented by its corresponding number. That is, an individual is a dendrogram which consists of a collection of clusterings, where each clustering is a partition of the universe of objects to be clustered. At first, each dendrogram of the initial population in our method (which we call EMHC, Evolutionary Method for Hierarchical Cluster) is

Figure 1: Evolutionary and visual framework for cluster analysis of DNA microarray data.

Figure 2: Example of a hierarchical clustering on the left side and its dendrogram on the right.

built up from the first level to the higher level by joining two clusters randomly chosen in the current level to create the next one [7].

2.1.1 Length of an Individual

The dendrogram length is defined as its number of levels (clusterings). If the size of a data set is $n$, then the dendrogram length is $n - 2$, assuming that the first and the last level are not included. But according to [7], in the best case, until the half of the dendrogram levels, there will be unitary clusters (one-element clusters) and that does not have practical meaning, hence, those levels can be removed. Following this reasoning, two or three-element clusters might not be also of interest and thus, a $\delta$ parameter can be introduced in order to remove the part of a dendrogram that does not give information from the user point of view. In other words, $\delta$ is the
proportion of levels (a fraction, in mathematical terms) in a dendrogram that we want to remove from the first level, because it is assumed as noise (or non-valuable information). Therefore, the length of a dendrogram is defined as follows:

Definition 1 Dendrogram length.
Let \( \mathcal{P}_n \) be a data set of size \( n \) and let \( \mathcal{G} \) be a dendrogram on \( \mathcal{P}_n \), then the length of \( \mathcal{G} \) is the clustering number of it and is defined as:

\[
|\mathcal{G}| = n - 2 - \left\lfloor \delta \cdot n \right\rfloor, \quad (1)
\]

where \( \delta \) is the level proportion of \( \mathcal{G} \) that is removed, assuming \( 1/2 \leq \delta < 1 \).

2.1.2 Fitness Function
To measure the goodness of candidate solutions (dendrograms) of EMHC, we have based on the concepts of homogeneity and separation given in [9]. In general, we have focused on the idea of the objects inside a cluster are very similar whereas the objects located in distinct clusters are very different. Therefore, the fitness functions for both, clustering and dendrogram have been defined as follows:

Definition 2 Clustering fitness function
Let \( D \) be the proximity matrix of a given data set, then the fitness function of a clustering \( \mathcal{C}_{i+1} \) in a dendrogram \( \mathcal{G} \) (i, the level in the dendrogram) according to \( \mathcal{H}_1^* \) and \( S_1^* \) (homogeneity and separation respectively) is defined as:

\[
g_c(\mathcal{C}_{i+1}) = \frac{S_1^*(\mathcal{C}_{i+1})}{g - k + 1} - \frac{\mathcal{H}_1^*(\mathcal{C}_{i+1})}{k - 1} + \max D, \quad (2)
\]

where \( S_1^*(\mathcal{C}_{i+1}) \) and \( \mathcal{H}_1^*(\mathcal{C}_{i+1}) \) have been defined in [7], \( k = |\mathcal{C}_i| \) and \( g = \binom{k}{2} \), being the number of distances among the clusters of \( \mathcal{C}_{i+1} \).

Definition 3 Dendrogram fitness function.
The fitness function of a dendrogram \( \mathcal{G} \), being \( \mathcal{C}_i \) (\( i \in [1, |\mathcal{G}| - 1] \)) clusterings of \( \mathcal{G} \):

\[
g_d(\mathcal{G}) = \frac{1}{|\mathcal{G}| - 1} \sum_{i=1}^{[\mathcal{G}]-1} g_c(\mathcal{C}_i). \quad (3)
\]

Once defined the fitness function for the individuals, our goal is to maximize \( g_d \) by obtaining small values for \( \mathcal{H}_1^* \) and large values for \( S_1^* \) on the clusterings of \( \mathcal{G} \). Based then on the previous definition, an agglomerative coefficient (ac) can be used in order to estimate the level of a dendrogram \( \mathcal{G} \) where carrying out a cut-off, that is:
Definition 4 Agglomerative coefficient.
Let $\mathcal{G}$ and $\mathcal{C}_i$ be a dendrogram on $\mathcal{P}_n$ and a clustering of $\mathcal{G}$, respectively. The agglomerative coefficient of $\mathcal{G}$ is defined as:

$$ac(\mathcal{G}) = \arg_{i \in [1,|\mathcal{G}|]} \max g_c(\mathcal{C}_i),$$

that is, level $i$ whose clustering has the maximum fitness of the whole dendrogram.

2.1.3 Mutation Operator

We have used the mutation operator (MO) defined in [7], which is a unitary alteration operator which only is applied on a single individual. It only transforms a part of a dendrogram, exploring different branches and returning a new dendrogram that replaces the previous one. Hence, the MO carries out an in-depth search. Only a part of the transformed dendrogram is modified with this operator and the another part is kept unchangeable. Indeed, since a dendrogram is special kind of tree (see Figure 2), this MO performs similar as moving a cluster associated to a branch of the dendrogram to another branch in the same dendrogram.

Figure 3-(A) summarizes, through a hypothetical example, the steps the MO. As shown, the dendrograms have been built on data set \{a, b, c, d, e\}. Dendrogram #1 shows (in red color) the clusters selected to be joined in order to build the next upper level. In this case, level 3 is selected to carry out the mutation process and create Dendrogram #2. Then, Dendrogram #2 is created from level 3 in Dendrogram #1 by making a different selection of clusters on this level (clusters in red color, Dendrogram #2). The process is repeated on the new level (in this case, level 4) until reaching the last level. As shown in Dendrogram #2, only levels from 3 to 5 in Dendrogram #1 have been modified. Note that Figure 3-(B) shows the same process as in Figure 3-(A) but in form of dendrogram graphs.

Apart of this MO, we can define a new genetic operator that supports the above mutation process. That is, an operator that can be used in combination with the above and acting as an intermediate genetic operator between the mutation and the crossover operator. Namely, the second mutation operator (SMO) on a dendrogram $\mathcal{G}$ has been defined as follows:

1. Choosing a random level $i \in [1,|\mathcal{G}| - 1]$.
2. Choosing two random cluster positions $(s, t)$ in the clustering of level $i$.
3. Choosing an element $s_\alpha$ in cluster $s$ and an element $t_\beta$ in cluster $t$ randomly.
4. Exchanging elements $s_\alpha$ and $t_\beta$ of clusters $s$ and $t$ in level $i$, respectively.
5. Spreading the modification of the above step to clusters of the remaining levels different of level $i$ for which $s_\alpha$ and $t_\beta$ belong.

2.1.4 Crossover Operator

The crossover operator (CO) has also been defined in [7] and performs by recombining valuable information of two individuals in order to build a new individual, which inheres the genetic
code of their ancestors. Thus, this carries out a wide search in the dendrogram space, taking two parent dendrograms as its input to obtain a single child dendrogram. On the one hand, this CO recombines clusters of the parent dendrograms and on the other hand, it also remains other clusters unchangeable of the parents to form the child dendrogram. In general terms, the CO randomly chooses the same level in both parents to form a new clustering (which is called seed clustering) by selecting the best clusters of the parents in the chosen level. After that, the child dendrogram is built by applying the above MO on the seed clustering to achieve the upper levels of the child and finally, a divisive strategy (for clusters) is also applied on the seed clustering to build the remaining lower levels.

Figure 4-(A) summarizes, through a hypothetical example, the steps of the CO. As shown, the used data set is \{a, b, c, d, e, f, g\}. From two hypothetical dendrograms (the parents) on this data set, a level is randomly selected (level 4 for this example) and the two clusterings (i and ii) of this level in both dendrograms are taken out as shown in this figure (Step #1). The most homogenous clusters (represented in green color, clustering i and ii) are selected for each clustering in order to create an intermediate clustering, which has repeated and missing object data (Step #2). Thus, this clustering is repaired to obtain a new clustering being the seed of the child dendrogram. With the seed clustering, the child dendrogram is built (Step #3) by applying the MO to form the upper levels to level 4 (starting by joining clusters \(f\) and \(b\) from the seed level). The lower levels to level 4 of the child dendrogram are formed by applying a divisive strategy (starting by splitting cluster \(d, e, g\) into clusters \(d, e\) and \(g\) from the seed level). Note that in Figure 4-(B) has shown both parent-dendrograms and the result of the crossover (child-dendrogram) in form of dendrogram graphs.
This CO can also optionally be improved by introducing an strategy performing on the seed clustering. That is, the information that could be lost in the clustering reconstruction process of the CO can be retrieved using an evolutionary search strategy. First, we define two operators to modify a clustering:

- **ALT1 operator:** given a clustering, it exchanges two elements between two clusters. Both clusters and elements are chosen randomly.
- **ALT2 operator:** given a clustering, it moves one element from a cluster to another cluster. Both clusters and the element are chosen randomly.

The evolutionary strategy to improve the seed clustering of the CO has been defined as follows:

### 2.1.5 EvolCluster Algorithm

**Input:** \( C \), a clustering. OP, a operator performing on \( C \), which can be ALT1 or ALT2. MaxGeneration, the number of iterations. EvalFitness evaluates the fitness of a clustering \( C \).

**Output:** \( C \), improved.

1. % Computing the fitness of \( C \).
2. \( f := \text{EvalFitness}(C); t := 0; \)
3. while \( t <= \text{MaxGeneration} \) do
4. \( t := t + 1; \)
5. % Applying an alteration.
6. \( C' := \text{OP}(C); \)
7. % Evaluating the new clustering.
8. \( f' := \text{EvalFitness}(C'); \)
9. % Updating of the improved clustering.
10. if \( f' > f \) then
11. \( f := f'; C := C' \)
12. endif
13. endwhile
14. end.

### 2.2 A Tool for Visual Cluster Analysis

Complementing our clustering framework, we have developed tool 3D-VisualCluster (3D-VC for short) [8], which loads the results of EMHC (and in general, the results of other methods) and displays them on different visualization components as a means of result visual comparison. Therefore, 3D-VC implements exploration of dendrograms (as input) in different ways: views
Figure 4: A hypothetical example summarizing the running of the crossover operator defined for the EMHC method. (A) shows the crossover process from the internal structure of the dendrograms whereas (B) shows the same, but from the graph of the dendrograms.

of microarrays (heatmap) including dendrogram and parallel coordinates for clusters, 3D scatter plot (dimensionality reduction based on principal component analysis by using the covariance matrix) with the clusters of the data as points, which can show boundary points and 3D-shapes for clusters and additionally, a reference partition can be loaded by tool to be compared with the clusterings of the dendrogram. For this case, the clusters in the reference partition (called r-clusters) are displayed through their 3D-surfaces.

A general view of 3D-VC has been shown in Figure 5 displaying a sketch of eight linked and interactive views and six tasks for visual cluster analysis. Note that the order of tasks (six tasks) involved in this figure states a methodology to follow in the visual analysis and validation of the results for a clustering method. Therefore, from the input of clustering results to 3D-VC, the user can follow the order of the tasks defined in this figure to make a process of visual analytics leading to knowledge discover from the interaction with the views provided by each task (see [8] for more details).

Note that representing reference partitions from r-cluster boundary points is one of the main contributions of this tool. This is because there are several statistical indicators for comparing a
Figure 5: General view of the tool. There are eight linked views: microarray, dendrogram and parallel coordinates views at the top of the figure; 3D scatter plot views at the bottom; cluster boundary points, reference partition surfaces and cluster surface reconstruction.
clustering with a reference partition, but there is no visual approach to validate this comparison. Each r-cluster in the reference partition is represented by our tool in a 3D surface. Thus, the gene-points at the intersection of a cluster with a r-cluster fall within the surface of such a r-cluster (in the space). In such case, one can check the degree of agreement between a clustering and a reference partition, or verify the results of statistical measures. Thus, it is possible to visually choose the level of the dendrogram that best approximates the reference partition.

2.3 Implementing the Framework

The clustering framework has been implemented in R-Project (R Development Core Team) and Java (Java-3D). The evolutionary part of our framework (method EMHC) has been developed on R and is freely distributed at http://cran.r-project.org/web/packages/clustergas. On the other hand, tool 3D-VC has been developed on Java and is linked to R in such a way that the results of clustering methods implemented on R can be read by the tool for subsequent visual analysis. 3D-VC is publicly available at http://eden.dei.uc.pt/~jgarzon/3DVisualCluster/3D-VisualCluster.

3 Analyzing a Case Study

This section outlines two comparative studies of the proposed framework on public data set lung [10], which comprises 73 lung tissues including 67 lung tumors for 916 gene observations for each lung tissue, namely, a gene expression matrix of 916 genes × 73 samples. 20-nearest neighbors have been used to estimate missing values of this data set and it has been published at http://genome-www.stanford.edu/lung_cancer/adeno. The goal of this case study is to show that EMHC can find better solutions than other methods for a new data set as lung, this way, proving the reliability of the global framework. Thus, the experiments are focused on the comparison of EMHC with five hierarchical clustering methods, that is, Agnes, Diana, Eisen, HibridHclust and TSVQ [2, 11, 12, 13], under a set of internal measures of cluster validity as Homogeneity (Homog), Separation (Separ) and Silhouette Width (SilhoW), all explained in [2, 7]. Note that the values of homogeneity decrease when the clustering quality (or dendrogram quality) increases, whereas the values of separation and silhouette width increase when the clustering quality (or dendrogram quality) increases. The experiments have been carried out on a 3GHz computer of 2GB of memory, using Debian GNU/Linux. Then, based on all the above, EMHC has been initialized according to Table 1, any other parameter of EMHC has been assigned as in [7].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value (or interval)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossover probability</td>
<td>[0.60, 0.75]</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>[0.10, 0.20]</td>
</tr>
<tr>
<td>Number of individuals</td>
<td>20</td>
</tr>
<tr>
<td>Number of generations</td>
<td>[10^3, 10^6]</td>
</tr>
<tr>
<td>Metric on data</td>
<td>Euclidean</td>
</tr>
</tbody>
</table>
3.1 Global and Local Comparison of Clustering Results

The performance of EMHC with respect to other methods is shown by means of global and local quality of the found solutions. Thus, by the first case (global quality), the measures of cluster validity are applied to the whole dendrogram for each compared method. Table 2 lists the values scored by each validity measure applied to the dendrogram returned by each method, the best values reached in each case have been underlined. The second case, that is, local quality, Table 3 lists the measure values applied to the best clustering\(^1\) of each output dendrogram. Column \#Clusters has the number of clusters of the clustering selected for each method and the best values for each measure have also been underlined. Finally, the runtimes of EMHC defined for this experiment were between 0.30 and 2 hours.

Table 2: Global cluster validity of EMHC vs. five hierarchical clustering methods from the lung data set.

<table>
<thead>
<tr>
<th>Method</th>
<th>Homog</th>
<th>Separ</th>
<th>SilhoW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agnes</td>
<td>14.25</td>
<td>20.84</td>
<td>0.14</td>
</tr>
<tr>
<td>Diana</td>
<td>12.79</td>
<td>15.31</td>
<td>0.05</td>
</tr>
<tr>
<td>Eisen</td>
<td>13.27</td>
<td>15.17</td>
<td>-0.03</td>
</tr>
<tr>
<td>HybridHclust</td>
<td>12.15</td>
<td>15.03</td>
<td>0.05</td>
</tr>
<tr>
<td>TSVQ</td>
<td>12.13</td>
<td>15.03</td>
<td>0.04</td>
</tr>
<tr>
<td>EMHC</td>
<td>14.38</td>
<td>21.60</td>
<td>0.14</td>
</tr>
</tbody>
</table>

Table 3: Local cluster validity of EMHC vs. five hierarchical clustering methods based on the best clustering of each output dendrogram from the lung data set.

<table>
<thead>
<tr>
<th>Method</th>
<th>#Clusters</th>
<th>Homog</th>
<th>Separ</th>
<th>SilhoW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agnes</td>
<td>7</td>
<td>14.45</td>
<td>22.30</td>
<td>0.23</td>
</tr>
<tr>
<td>Diana</td>
<td>29</td>
<td>11.98</td>
<td>15.10</td>
<td>0.05</td>
</tr>
<tr>
<td>Eisen</td>
<td>9</td>
<td>13.40</td>
<td>15.19</td>
<td>-0.01</td>
</tr>
<tr>
<td>HybridHclust</td>
<td>31</td>
<td>11.27</td>
<td>14.90</td>
<td>0.03</td>
</tr>
<tr>
<td>TSVQ</td>
<td>31</td>
<td>11.31</td>
<td>14.90</td>
<td>0.03</td>
</tr>
<tr>
<td>EMHC</td>
<td>3</td>
<td>14.61</td>
<td>23.11</td>
<td>0.30</td>
</tr>
</tbody>
</table>

As these two tables show, the best values for separation and silhouette width have been achieved for the EMHC and Agnes method. The best values for homogeneity have been achieved for methods TSVQ and HybridHclust. Figure 6 shows the above by representing each method as a curve, where the methods in groupings \{EMHC, Agnes\} and \{HybridHclust, TSVQ\} have similar behavior. Summarizing on EMHC, we can say that it performs better than the other methods on separation measures and on measures that combine homogeneity and separation (such as, silhouette width), but it does not have the same performance on homogeneity. Note that as a part of our framework, the given results can visually be analyzed by tool 3D-VC to validate the used measures [8].

\(^1\)The best clustering according to \(ac\), Definition 4.
3.2 Visual Analysis of the Results

The results obtained by the clustering methods can be displayed with the 3D-VisualCluster tool. This allows us to visually verify the quality of the returned dendrograms and extract conclusions on the performance of the used methods, since there is not a reference partition to externally validate the results. To do that, we have chosen the two clustering methods with best performance on the data set. That is, the method that better has performed on homogeneity and the one on separation have been selected to display their dendrograms. TSVQ has been the method whose dendrogram has the best homogeneity and the dendrogram with better separation has been for EMHC.

The goal of this experiment is to provide visual results as a means of comparison with the results given by the validity measures in the previous tables. This way, to also show that some attributes (as for example, the number of clusters) in the results given by the validity measures in those comparative tables do not match their visualizations. This will prove that for deciding on a clustering result, we need to apply more than one approach of validation and so to decide on the global result from all applied approaches, which will improve the process of cluster analysis.
Then, Figures 7 and 8 show the dendrograms (on the lung microarray) given by TSVQ and EMHC respectively. Figure 9 shows the data distribution of the lung data set in a 3D space. The gene distribution of lung shown on the scatter plot in Figure 9 shows that lung is a very compact data set. A consequence of this compactness is that, the homogeneity concept applied in the previous tables could fail since it is hard to separate (extract) clusters. Since TSVQ just checks homogeneity and does not separation, it has found good compact clusters but it is not able to separate the small clusters located on the exterior region of the data set from the ones located in its compact region on the center. Figure 9. In contrast, EMHC has been able to detect both types of clusters. Hence, it is better fitted the distribution of the data set given in Figure 9. Note that the small clusters located on the exterior region of the data set have been found by EMHC in the construction final part of the dendrogram given in Figure 8. On the left and right side of the microarray and dendrogram of this figure can be seen how are added small clusters.
4 Conclusions

The goal of this paper has been to show the results of our research on a new case study. According to this, we have presented a clustering framework joining a genetic algorithm (EMHC) and a tool (3D-VC) for the analysis of hierarchical clustering. The results shown on the evolutionary part of the framework, that is EMHC, prove that it has performed well for the new used data set (lung), finding better solutions than the other methods. Moreover, we have detected that EMHC performs better on separation and measures that combine homogeneity and separation. On the other hand, the functionalities shown by 3D-VC can be very useful for the visual validation of clustering results. Thus, our global proposal has proven its reliability for the cluster analysis of DNA microarray data.

Acknowledgements

This work has partially been funded by the Spanish Ministry of Science and Innovation, the Plan E from the Spanish Government, the European Union from the ERDF (TIN2009-14057-C03-02).

References


