Analysis of genetic association using Hierarchical Clustering and Cluster Validation Indices

Inti A. Pagnuco[1,2,3], Juan I. Pastore[1,2,3], Guillermo Abras[1], Marcel Brun[1,2], and Virginia L. Ballarin[1]

[1]*Digital Image Processing Lab, ICyTE, UNMdP,* [2]*Department of Mathematics, School of Engineering, UNMdP,* [3]*CONICET*

Abstract

It is usually assumed that co-expressed genes suggest co-regulation in the underlying regulatory network. Determining sets of co-expressed genes is an important task, based on some criteria of similarity. This task is usually performed by clustering algorithms, where the genes are clustered into meaningful groups based on their expression values in a set of experiment.

In this work we propose a method to find sets of co-expressed genes, based on cluster validation indices as a measure of similarity for individual gene groups, and a combination of variants of hierarchical clustering to generate the candidate groups. We evaluated its ability to retrieve significant sets on simulated correlated and real genomics data, where the performance is measured based on its detection ability of co-regulated sets against a full search. Additionally, we analyzed the quality of the best ranked groups using an online bioinformatics tool that provides network information for the selected genes.

1 Introduction

New technologies for genomic analysis measure the expression of thousand of genes simultaneously. An important goal in some studies is to find sets of co-expressed genes, to study co-regulation and biological functions. This task is usually performed by clustering algorithms, where the whole family of genes, or a subset of them, are clustered into meaningful groups based on their expression values in a set of experiment. These techniques provide insight on the possible co-regulation between genes, under the hypothesis that co-expression may indicate evidence for co-regulation, and these hypotheses of co-regulation must eventually be corroborated or be rejected by further experiments. Existing clustering algorithms are based, usually, on a global approach [1, 2, 3, 4, 5], or on just pairwise analysis [6]. While some of them provide methods for the selection of the best number of clusters [2], they result in a very large number of clusters, depending on the number of groups required or determined by the algorithm. The large size of these sets makes their use as hypothesis generator improbable. A more comprehensive approach consist on the analysis of co-expression of all possible sub-sets of genes. While simple, this approach fails because of the need to analyze an exponential number of candidate sets. This search of meaningful sets would be impractical if the number of available genes for the analysis is large, which is usual in this field.

In this work we present a new algorithm, that combine the particularities of clustering hierarchical algorithm with individual cluster validation based on the Silhoutte and Dunn indices, to generate a ranked list of gene group, avoiding the exhaustive search, but providing high quality results. As in [2] we use the cluster validation index as a quality/homogeneity measure, but instead of using it to generate the partitions, we use it to select the best sets, from several partitions, obtained by different variants of hierarchical clustering. This approach permits the use of many different clustering algorithms, combining the best results of each one of them, avoiding the use of a full search approach, while still providing, as shown in the results, a good approach to the optimal results.

The Matlab code is available from https://subversion.assembla.com/svn/pdilab-gaa/tags/Tag.1.1.1

In the next section we present a) an introduction to pattern recognition tools, b) the proposed algorithm, c) the result in genomic data. The final conclusions show that these algorithms could result in a useful tool to the researchers as a preliminar technique to data analysis.

2 Approach

2.1 Pattern Recognition

Pattern Recognition techniques have been widely used to identify objects according their features [7]. Clustering algorithms is a collection of these techniques that groups unclassified patterns based on similarity.

In this work, the objects to be classified are genes. The data consists of a set of m samples $S_1, S_2, .., S_m$ and n genes $g_1, g_2, ..g_n$ that are normally represented by a matrix of two dimension M where $M(i, j)$ represents the expression of the gen g_i for the sample S_j. The expression of each gen g_i, across all samples, correspond to a row of the matrix M, and it is represented by a feature vector $X_i(x_{i1}, x_{i2}, .., x_{im})$, where each value x_{ij} represent the expression of the gen g_i for the sample S_j. The feature vector is then composed by m samples, so that the regions of the partition H belong to R_m.

Each gen can be assigned to one of k possible groups, and the result of a clustering algorithm is a partition of the set of genes, as a family $W = W_1, W_2, .., W_k$, where each W_i is a group of genes with a certain degree of similarity.

2.2 Hierarchical Clustering

Hierarchical Clustering is one of most used clustering algorithms in bioinformatics [8]. Given a set of N items to be clustered, and a $N \times N$ distance (or similarity) matrix, the basic process of hierarchical clustering [9] is described here: (i) Start by assigning each item to a different cluster, so that for N items, there are N clusters, each containing just one item. Let the distances between the clusters be

the same as the distances between the items they contain. (ii) Find the closest pair of clusters and merge them into a single cluster, so that now there is one cluster less. (iii) Compute distances between the new cluster and each of the old clusters. (iv) Repeat steps 2 and 3 until all items are clustered into a single cluster of size N. A more detailed description of the hierarchical clustering algorithms can be found in [9, 10].

Step 3 can be done in different ways, which is what distinguishes single-linkage, complete-linkage and average-linkage clustering. In single-linkage clustering, the *distance* between two clusters is the shortest distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, the *similarity* between two clusters is the largest similarity from any member of one cluster to any member of the other cluster. In complete-linkage clustering, the distance between two clusters is the largest distance from any member of one cluster to any member of the other cluster. In average-linkage clustering, the distance between two clusters is the average distance from any member of one cluster to any member of the other cluster.

2.3 Quality measures

A common problem of clustering algorithms is the validation of results. In the next subsection we explain the two types of internal validation [11] used in this work.

2.3.1 Silhouette Index

One of the methods of internal validation used is Silhouette index, which measures the quality of clustering as the average quality of its elements [12]. The Silhouette value for an element ranges between -1 and 1. A high Silhouette value indicates that the element is closer to its own cluster elements than to the ones that do not belong to its own cluster.

To define the Silhouette index for the points, two measures need to be defined first. Lets N be the number of clusters. Let $X \in C_k$ a point (element) that belongs to the cluster C_k, and let n_k the number of elements of C_k, then the first measure $a(X)$, the average distance of X to the points of C_k is defined by:

$a(X) = \frac{1}{n_k - 1} \sum_{Y \in C_k, Y \neq X} d(X, Y)$

The second measure, $b(X)$, the average distance to the nearest cluster, for a point $X \in C_k$, is defined by:

$b(X) = \min_{h=1..N, h \neq k} \quad \frac{1}{n_h} \sum_{Y \in C_h} d(X, Y)$

Finally, the Silhouette index $S(X)$ is defined by:

$$S(X) = \frac{b(X) - a(X)}{max(a(X), b(X))} \tag{1}$$

Then, the Silhouette index for a cluster is defined as the average Silhouette of its points:

$$S(C_k) = \frac{1}{n_k} \quad \sum_{X \in C_k} S(X) \tag{2}$$

It can be seen that if $b(X) \gg a(X)$ (the point is closer to points in the same cluster than points in other clusters) then $S(X)$ is close to 1. If this happens for most of the points in a cluster C, then the Silhouette $S(C)$ is also close to one, indicating that it is isolated from other clusters.

If $b(X) \approx a(X)$, (there are points outside the cluster closer to X than points in the same cluster), then $S(X)$ is close to 0. If this happens for most of the points in a cluster C, then the Silhouette $S(C)$ is also close to zero, indicating that it is mixed with some other clusters.

2.3.2 Dunn Index

Another popular internal validation index is the Dunn index which, given a partition of the space of points in K clusters, is defined as the ratio between the minimum distance between two clusters and the size of the largest cluster [13]. If $C = \{C_1, ..., C_K\}$ is a partition of the n points into K clusters, then the Dunn index is defined by

$$D(C) = \frac{min_{h,k,h \neq k} d(C_k, C_h)}{max_{k=1,...,K} \Delta(C_k)} \tag{3}$$

where $d(C_k, C_h)$ is the distance between the two clusters and (ΔC_k) is the size of the cluster C_k . The value of $D(C)$ depends on the selection of the size and distance measures. Several measures for the distances between clusters (or linkage) are proposed in Ref [14].

3 Methods

3.1 Proposed algorithm

The goal of the proposed algorithm is to select subsets of genes highly correlated. In the expression profile space of genes (expression through all samples), provided by a distance function, where each genetic profile is represented with a point of R_n, this corresponds to finding compact sets that are separated from other points sets. A naive way to find these compact and separate groups would consist in *measuring* the compactness of all possible subset of the N profiles, which is not practical for large values of N. In the following subsection we present the measures used to evaluate the cluster compactness and the methodology used to find subsets highly correlated.

3.1.1 Evaluation of partition quality

A suitable measure to evaluate the cluster compactness may be the Silhouette Index [12], successfully used by the authors for other purposes [15, 16]. For our analysis, we use a slightly modified measure of cluster silhouette. In this case, for each set C_k of points, to compute its Silhouette we assume that all the points that do not belong to C_k belong to a second cluster C^c, and the *modified* silhouette index is computed based on this 2-clusters situation. We call it the *Cluster Specific Silhouette Index*, and it is defined by:

$$S(C_k) = \frac{1}{n_k} \quad \sum_{X \in C_k} S(X) \tag{4}$$

Another suitable measure is the Dunn Index. One of the issues about using the Dunn index is that it relates the size of all clusters against the distance between them, and does not provide a cluster specific measure for each cluster. We adapted the measurement to provide a measure of cluster quality for an individual cluster C_k, given by

$$D(C_k) = \frac{d(C_k, C^*)}{\Delta(C_k)} \quad (5)$$

where C^* is the set of points not in C_k (the same idea used with Silhoutte), $d(C_k, C^*)$ is the distance between the two clusters and (ΔC_k) is the size of the cluster C_k. We call this index a *Cluster Specific Dunn Index*.

A key aspect of this work is the selection of a cluster score that reflects the compactness and separateness of each set of profiles relative to the rest of the profiles. Many cluster validation measures can be found in the literature grouped in three categories: internal, relative and external [17, 18]. On this specific problem, external indices are not viable because we just don't know what is the expected partition. In the other hand, relative indices were proven to be unreliable regarding cluster quality in a probabilistic framework [19]. Finally, internal validation indices are well suited for the task of describing how close are the profiles of the groups, relative to the distance to other profiles. Inside this category we can also find many indices, including Silhouette, Dunn (and its many variants), correlation to the distance matrix, etc. In our case, we decided to use two of the most popular indices, Silhouette and Dunn, since they explicitly ponderate intra clusters compactness vs inter cluster distances, and can be adapted to measure the quality of a unique cluster, instead of the overall clustering (as done in section 3.1.1). They also proved to be reliable when we don't expect clusters with too extreme shape [19]. Both of them will be described in more detail in sections 2.3.1 and 2.3.2.

3.1.2 Search of good subsets

To solve this problem, in this paper we propose to limit the family of subsets where to search, using hierarchical clustering to generate a family of candidate subsets, and then evaluate the *modified* clustering validation indices only on those subsets. An application of the hierarchical clustering algorithm generates a total of $2 * N$ subsets to process. Because there are so many variants of hierarchical clustering, an important problem is to determine the best variants of hierarchical clustering for the task of detecting co-expressed genes on genomic data.

In a previous work [20] we studied the ability of variants of hierarchical clustering (complete-linkage, average-linkage, single-linkage) to detect the best subsets using Silhouette index as quality measure. To this purpose we used five different sets of simulated data with known result. Keeping the size of the artificial sets small, we were able to analyze all possible groups, sorting them by one of the clustering validation index, and use that information as gold standard to evaluate the performance of variants of hierarchical clustering to detect the best subset. The algorithm proceeds in the following way for a dataset of N profiles: (a) Apply hierarchical clustering to the profiles to obtain a complete dendrogram. (b) For $nc = 2$ up to $N - 1$: to partition the set in nc clusters based on the dendrogram. For each cluster, if the set size is above the minimum threshold (usually $nc >= 2$) then 1) Compute the score for the cluster, 2) Store the cluster and its score on a list. (c) Once a list of sets and scores is obtained, sort the list based on score (higher to lower).

As an additional approach, we combine all the variants, forming a large sets of candidate groups, still smaller than the maximum 2^N sets, and select the best groups from this set. This approach extends the search space, relative to the use of only one algorithm, maintaining the computational requirements still low. Another advantage of this approach is that new clustering algorithms may be added to the pool, generating more candidate groups, therefore improving the overall quality of the result. In addition, this approach does not depend on the selection of a specific combination. Every algorithm, or variant of an algorithm, used to partition the set of profiles, can be added to the combination, because it could only improve the results, by adding new sets not previously detected [21, 22, 23, 24]. The main steps of the proposed algorithm are: (a) Apply the previous algorithm for different hierarchical clustering parameters (linkage, distance); (b) For each algorithm, register in a list all the possible branches in the dendrogram tree. Combine all the previous lists into one list; (c) Compute the cluster specific validation index (score) for each group. (d) Sort the new list based on score index (higher to lower). Select those groups that have clustering validation index above a threshold.

The important step in this method, that sets it apart from previous methods, is that it does not use just one clustering (partition) of the genes, based on an algorithm, but a larger set of groups, defined by the dendrogram tree from the hierarchical clustering algorithms. Allowing the evaluation (via the modified indices) of intermediate groups (ones not showing on an *optimal* partition) avoids issues where a large group is not part of that partition, but would be a good candidate because of its compactness.

In the previous work [20], the analysis was aimed at finding which variant of hierarchical clustering detected a major proportion of the groups with highest Silhouette (determined by a full search in simulated contexts). In these analysis it was visible that no variant has better performance, and here we see that a combination of all the variants outperforms any of them.

In the next sections we apply this approach, based on the combination of variants to generate the candidate clusters and the rank based on the two modified indices, to search for correlated genes in five simulated and two real data situations.

4 Validation on Simulated Data

This section compares the variants of hierarchical algorithm relative to their individual performance on different cases. We define five synthetic datasets consisting in 10×30 profile matrices, where each row is a variable (gene) and each column represents a sample. With these small sizes, we are able to generate a *gold standard* by evaluating the clustering validation indices for the 2^{10} possible subsets of the 10 genes. The model for the data consist in a multivariate Gaussian distribution, the same means, and co-variance matrices adjusted to represent different situations of correlation between the 10 genes (see

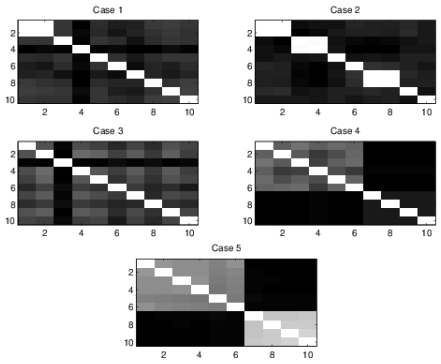


Figure 1: Covariance Matrices for the synthetic datasets. Case 1: the variables (1,2) are highly correlated. Case 2: (1,2), (3,4), and (7,8) are highly correlated. Case 3: variable 3 is uncorrelated to the other variables. Case 4: variables 1 to 6 show low correlation, plus four more uncorrelated genes. Case 5: two blocks with relatively high correlation.

Figure 1).

As a *gold standard* to evaluate the performance of the approach, we computed all possible groups and its associated quality index, then we sorted them by the indices values, generating a *ranking* of sets, where the top ranked sets are the ones with best clustering validation index.

We compared the performance of three variants of hierarchical clustering: a) average linkage; b) complete linkage, and c) single linkage. To measure the similarity of profiles, we used two different similarity measures: a) Euclidean distance and b) correlation. The proposed approach combines all the previous variants.

The evaluation of the ability of each method to detect co-expressed genes is based on the detection of highly ranked genes, from the *gold standard*. We use two measures of quality: a) average index value, and b) average ranking. The average index value, for each method, is computed as the average index value of all the candidate groups produced by the method. The higher the average index value, the better the method. The average ranking is obtained by computing the position (or rank) of each candidate group, and averaging these values. If a method generates good results, the groups ranks would be small values (The optimal results would be the top ranked groups, with ranks 1,2,3,...). The lower the average ranking, the better the method. About the number of candidate groups, a classical hierarchical clustering algorithm would provide, for N genes, $N-2$ different candidate groups, since the one-gene groups, and the N genes group, are not processed. Supplementary Table S1 shows all the options for analysis.

Because we use a realistic number of samples (30 samples) for our simulations, the sample correlation and variance differs from the model ones. As an example, in Supplementary Table S2 we can see the computed correlation from the data generated randomly from the first model. where the correlation between the two first variables is hight (0.995) while other values, which should be very low, range between 0.004505 and 0.28178.

In the following subsections we present in detail the results obtained using Silhouette to evaluate the different approaches, and a comparison with the result of the analysis using Cluster Specific Dunn Index.

4.0.3 Case 1

In this case, the variables (genes) 1 and 2 are highly correlated, while the other variables show low correlation (figure 1 - Case 1). The algorithm should be able to detect groups containing genes 1 and 2 as top candidates. Supplementary Table S3 shows the detailed results for all the analyzed methods.

For example, as expected, the best group is formed by variables 1 and 2, with Silhouette index of 0.73, and it was properly detected by all the methods. The second best group, formed by the variables 1, 2 and 8, with Silhouette of 0.27, was not detected by any of the methods (and is not listed in the table). Finally, the third best group, formed by variables 1, 2 and 10, with a low Silhouette value of 0.25, was not detected by three of the methods.

4.0.4 Case 2

In the second case, the pairs of variables $(1,2)$, $(3,4)$, and $(7,8)$ are highly correlated, while the other variables show low correlation (figure 1 - Case 2). Supplementary table S4 shows the results for all the analyzed methods. As expected, the best three groups are formed by variables pairs $(1,2)$, $(3,4)$, and $(7,8)$, with Silhouette index of 0.91, 0.57 and 0.54 respectively. The fourth best group, formed by the variables 1, 2 and 10, with Silhouette of 0.38 was not detected by three of the methods. And the fifth best group, formed by the variables 1, 2 and 3 was not detected by any of the methods (and is not listed in the table).

4.0.5 Case 3

In the third case, variable 3 is uncorrelated to the other variables, while the other variables have a low correlation (figure 1 - Case 3). Supplementary table S5 shows the results for all the analyzed methods. As expected, variable 3 is not a member of the best groups. This variable only appeared in the last group detected, with Silhouette index of 0.06, a very low Silhouette index which by itself would discard this group. Due to the low correlation displayed between variables, even the top groups would be discarded based on a practical Silhouette threshold.

4.0.6 Case 4

In the fourth case, there is a six genes block with low correlation, variables 1 to 6, plus four more uncorrelated genes (figure 1 - Case 4). Supplementary table S6 shows the results for all the analyzed methods. As expected, the top groups include variables 1 to 6, from the first block, while lower ranked groups link variables 7 to 10, uncorrelated. Because the correlation level is low, the top level groups show low Silhouette indices, around 0.32.

4.0.7 Case 5

In the fifth case, there are two blocks with relatively high correlation. There is a six genes block (variables 1

Table 1: Average ranking of the top eight groups detected by each clustering strategy (Supplementary Table 1) using euclidean distance to evaluate Silhouette.

Silhoutte = 'EU'	Case 1	Case 2	Case 3	Case 4	Case 5
Combination	13.375	7.250	5.750	8.250	4.500
Av_ce	35.500	27.875	14.000	20.500	8.625
Av_eu	25.125	15.500	8.625	17.000	8.750
Co_ce	60.500	43.750	36.375	20.500	12.125
Co_eu	13.375	29.250	8.625	19.625	8.750
Si_ce	110.500	50.000	8.500	61.625	14.500
Si_eu	27.625	17.875	10.250	32.750	5.625

to 6) with low correlation in comparison with the another four genes block, variables 7 to 10, with higher correlation (figure 1 - Case 5). Supplementary table S7 shows the results for all the analyzed methods. As expected, the top groups include variables 7 to 10, from the second block, while lower ranked groups link variables 1 to 6, from the first block. It is interesting to note that the top ranked group is not actually the whole second block, but a subset of it. The whole second block is ranked 2nd. An additional simulation with 10.000 samples, where the sample covariance is closer to the model covariance, produced results where the two main groups (variables 1 to 6 and variables 7 to 10) are ranked 1st and 3rd in the list (results not included here). This shows that the small sample size affects also the results/performance of the algorithm.

The analysis was aimed at finding which variant of hierarchical method detected a major number of groups with high Silhoutte. Because no variant of hierarchical clustering showed better performance, we decided to use a new method, that combines the variants (third column in result tables). This new approach showed the best results compared with each variant of hierarchical clustering. Table 1 summarizes the average indices of the top eight groups detected by each strategy. Supplementary Table S8 shows the average Value Silhouette of the top eight groups detected by each strategy. In both cases we use Euclidean distance to compute the Silhouette Index.

Forward Search is another popular search algorithm, but it is oriented to find one best subset, while the proposed algorithm provides a ranked list of subsets. For this reason we considered that the proposed algorithm is better suited for the problem at hand.

4.0.8 Analysis using Cluster Specific Dunn Index

Although the Silhouette index is one of the most used cluster validation indices, and it is well adapted for the task proposed here to evaluate how compact is a cluster against all other data points, we compared the results, on simulated data, against the modified Dunn validation index, that was described in previous sections.

We repeated the previous analysis for all the simulated cases, using the Cluster Specific Dunn Index as quality index, and evaluated its ability to detect the most correlated variables.

The comparison was done, mainly, computing the average ranking (or position in the list) of the top eight detected clusters. In case of Silhouette, the ranking was defined by computing the Silhouette index for all subsets, and in the case of Dunn, the ranking was defined by computing the Dunn index for all subsets. Therefore, the main comparison resides on whether the sub-optimal search, based on combination of variants of the hierarchical clustering algorithms, can detect the top ranked subsets. There are not big differences when we compare the top eight detected clusters (See Supplementary material).

To evaluate the overall ability of the proposed combined method to detect good groups, we repeated the analysis, but computing the average rank for all the detected groups: 21 for Case 1, 17 for Case 2, 13 for Case 3, 18 for Case 4, and 15 for Case 5. Supplementary Table S9 shows the resulting average ranking for all five cases and both indices. Overall, except for Case 2, the average rank of the detected groups is lower for the Silhouette Index.

The previous analysis shows that the sub-optimal search by combination of clustering algorithms is better suited to detect good clusters (with low Silhouette Index, or high Dunn Index) when using the Silhouette Index than when using the Dunn Index. It is important to note that both indices are measuring the compactness and separation of a cluster in different ways. On the other hand, there is not a great difference between both methods regarding the detection of highly correlated sets. Some sets are ranked higher by Silhouette, and others are ranked higher by Dunn.

5 Application Examples

In this section we describe two application examples, where the algorithm is used to analyse two different sets of data: microarray based expression for diabetes, and QTLs for listeria. In both cases, we applied the algorithm using the Silhouette Index as a measure of quality to detect significant sets of genes/QTLs, and then analysed their significance based on existing knowledge about them.

5.1 Listeria

In the first application example, we used data of mouse susceptibility to monocytogenes listeria [25, 26]. This study consists in the analysis of the relation between QTLs (*Quantitative trait Loci*, stretches of DNA containing or linked to the genes that underlie a quantitative trait) and the *survival* time of 120 mouses that have been infected with listeria, where *survival* is defined by a survival time of more than 240 hours. The dataset consists of 35 surviving mice and 85 non-surviving, analysing 133 QTLs for each mouse. After filtering those QTLs with missing data, only 28 QTLs were retained.

It is important to note that here we are using QTLs instead of genes as target for our search of correlation. Because of the nature of the model used here, it is not restricted to expression information, the same analysis can be applied to discrete QTL studies.

To verify the performance of the algorithm in this context, considering the small amount of markers under study, we were able to perform a full search of QTLs

Table 2: Comparison of clustering results for listeria dataset. In the 1st column is detailed the group position for all possible groups (full search algorithm) in function of Silhouette value (2nd column). In the 3rd column is detailed the group position for the combination algorithm. In addition the number of elements and the members of groups.

Group	Sil	Id_comb	Num_elem	Groups Member
1	0.90661	1	2	8,9
2	0.86923	2	2	3,4
3	0.86788	3	2	1,2
4	0.814008	4	2	25,26
5	0.736231	5	3	8, 9, 10
6	0.701954	6	3	24, 25, 26
7	0.672453	7	2	11,12
8	0.66783	8	4	1,2,3,4
18	0.592628	9	2	14,15
21	0.563286	10	2	17,18
24	0.554229	11	5	8,9,10,11,12
37	0.49119	12	3	16,17,18
44	0.452103	13	6	8,9,10,11,12,13
69	0.393421	14	5	14,15,16,17,18
71	0.387993	15	4	23,24,25,26
327	0.293433	17	6	14,15,16,17,18,20
346	0.290286	18	6	6,14,15,16,17,18

Table 3: Groups detected with higher Silhouette using the combination algorithm on diabetes dataset. In the 1st column is detailed the group position for the combination algorithm result in function of Silhouette value (2nd column). The 3rd column detailed the group size, the 4th column the Affymetrix reference. In the 5th column is reported the similar predicted biological functions for the genes found on these groups.

Group	Silhouette	Size	Probes (Affy Ids)	Analysis
1	0.9548	2	204550_x_at 215333_x_at	Unique gene
2	0.9429	3	204418_x_at 204550_x_at 215333_x_at	2 Genes from the same family.
3	0.82689	2	207831_x_at 207907_at	No relationship found.
4	0.80904	2 2	201639_s_at 201904_s_at	No relationship found.
5	0.79401	2	205175_s_at 213670_x_at	Indirect metabolic relationship
6	0.79394	2	200966_x_at 214687_x_at	Unique gene
7	0.79345	2	200991_s_at 202676_x_at	Common function: Protein binding
8	0.79343	3	201379_s_at 207831_x_at 207907_at	Common function: Protein binding

groups (up to 7 elements), computing the Silhouette index of all of these subsets. The performance of the search algorithm (based on clustering) is measured based on the number of best subsets detected.

Table 2 shows the comparison of results between full search (column 1) and combination (column 3) of variants of hierarchical clustering. We can see in the table that the 8 most significant groups were properly detected by the algorithm. Additionally, the top groups of 2, 3, 4, 5 and 6 elements, were successfully detected by the algorithm.

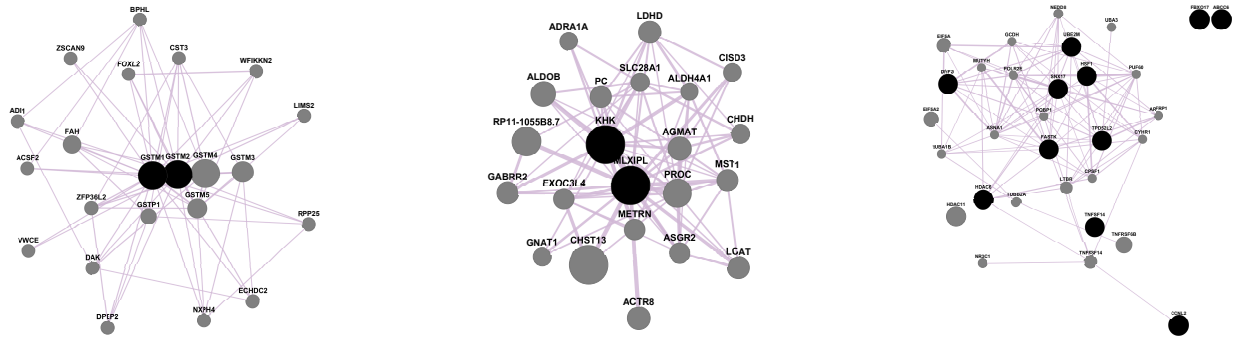
5.2 Diabetes

In this second example of biological application we ran the algorithm on microarray data. This data was obtained from a previous study that analyzed the expression profiles of obese and skinny subjects [27]. This study presents the expression profile of 18 subjects, 13 with obesity and 5 without, using a U133A chip of Affymetrix. From the original dataset, with 22283 genes, only the most variable 1000 genes were preselected for the analysis. In this case, it is not possible to compare the resulting groups with a full search, since there are 2^{1000} possible subsets. Therefore, the resulting groups/sets were studied based on actual biological knowledge about them. We restricted the analysis to the top eight ranked sets, described in Table 3. We verified if the genes found on these groups, have similar predicted biological functions. It should be noted that in group 1, even if there are two different Affymetrix probes 204550_x_at and 215333_x_at, they make reference to the same gene. The same situation is repeated for group 6.

The most interesting case identified is Group 2. In this group there are 3 probes, 2 of them reference to the same gene called GSTM1 (Glutathione S-transferase mu 1). The relation between these 2 probes and the third one, which references to gene GSTM2 (Glutathione S-transferase mu 2) is that both of them are members of the same family, and they are involved in a metabolic process. We analized these genes with the GeneMania bioinformatic tool, using the Co-expression option. GeneMania searches large biological datasets to find related genes, including protein-protein, protein-DNA and genetic interactions, pathways, reactions, gene and protein expression data, protein domains and phenotypic screening profiles [28]. Figure 2 shows the result of the analysis of co-expression for groups 2, 5 and 66. Figure 2 shows a large amount of linkage between the genes of the selected groups. Most of the associations are obtained from the Gene Expression Omnibus database (GEO), since GeneMania only collects data associated with publications. For group 2 (Figure 2(a)) some functions associated are glutathione transferase activity, glutathione derivative metabolic process, peptide metabolic process, peptide binding, modified amino acid binding, etc.

The third and fourth group have not documented relationships. From this analysis, they could be good candidates for further analysis of co-regulation, or other biological relationship.

The fifth group shows interesting relationships. The first Affymetrix Identifier (205175_s_at) references to a locus NSUN5P1. This locus represents a transcribed pseudo gene of a nearby locus on chromosome 7, which encodes a putative methyltransferase. Diseases associated with NSUN5P1 include Williams-Beurem Syndrome. The other group member references to the KHK gene (213670_x_at). The KHK gene encodes ketohexokinase that catalyzes conversion of fructose to fructose-1-phosphate. The product of this gene is the first enzyme with a specialized pathway that catabolizes dietary fructose (GeneCard information [29]). Due to the fact that NSUN5P1 is a pseudo gene, we can not do an analysis of relation with the KHK gene (probe



(a) (b) (c)

Figure 2: Results from GeneMania tool for genes group of diabetes dataset. Black spots are the group members, the pale lines and spot make references to biological functions reported in literature. a) Graph analysis for group 2, where the members are closely related; b) Group 5; c) Group 66.

213670_x_at). For this reason we search genes associated to Williams-beuren disease with pathways related to the KHK gene. In this search we found the MLXIPL gene that encodes a basic helix-loop-helix leucine zipper transcription factor. This protein forms a herodimeric complex and binds and activates, in a glucose-dependent manner, carbohydrate response element (ChoRE) motifs in the promoters of triglyceride synthesis genes. This gene is deleted in Williams-Beurem Syndrome. Both genes are involved in the metabolism process. Figure 2(b) shows the result of a co-expression analysis between MLXIPL and KHK gene, using the GeneMania bioinformatic tool.

For groups 7 and 8 we used GeneCards for analysis. GeneCards is a searchable, integrated database of human genes that provides information on all known and predicted human genes [29]. For both groups their members have a common function, that is protein binding, according to GeneCards.

Another case analyzed was a big group with eleven elements, this group had been ranked in the position 66, and its Silhouette index is 0.7044, a large value based on our past experience with the Silhouette index. We searched the elements and ran a co-expression analysis with GeneMania. Figure 2(c) shows the relation between some elements. To verify the quality of the result we select randomly 11 genes and ran the same analysis, the result obtained was a graph more dispersed (Graphic not included here).

6 Discussion

In this work we applied a simple but powerful method to identify groups of genes/markers that are compact and well separated from other genes/markers, using two clustering validation indeces to rank the sets, and a combination of several variants of hierarchical clustering to search the best sets.

We evaluated the ability of this method to detect correlated groups with simulated data, where different situations were created, and due to the small number of variables, a full search was possible. In these examples, the algorithm performed well on detecting the correlated blocks from the simulated data.

The analysis determined that the sub-optimal search by combination of clustering algorithms is better suited to detect good clusters when using the Silhouette Index than when using the Dunn Index. However, there is not great difference between both methods regarding the detection of highly correlated sets. Some sets are ranked higher by Silhouette, and others are ranked higher by Dunn.

On the other hand we evaluated its performance on real data. In one case, due to the small size, the resulting groups were also compared to the results from full search. In the second case, the resulting groups were analyzed using standard bioinformatics tools, verifying strong relationship between the genes in the top groups.

The algorithm provides a balance between search time and detection rate. It avoids the full search, which can be impractical for large number of genes, with the cost of missing some good sets, but it is able to detect most of the top ranked sets, which is not usually possible by using only one clustering algorithm.

The effectiveness of this algorithm is related to the ability of the validation clustering index to score properly compact groups, which are at the same time separated from other groups of genes.

7 Conclusion

With this analysis we verified the suitability of the proposed tool for the detection of sets of co-expressed genes, and we considered that it is useful for biologists or researchers in computational biology interested in generating new hypotheses about the co-expression of genes, or genomic markers like QTLs, which are not provided in most standard tools. This algorithm will quickly generate a set of good groups based on a clustering validation index, and combines the advantages of each variant of hierarchical clustering algorithm. Future work includes the analysis of other indices of group quality, besides Silhouette and Dunn, and the application to new datasets, including SNPs.

Acknowledgement

This work was partially supported by CONICET and FONCYT.

References

[1] L. Heyer, S. Kruglyak, and Shibu Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9:1106–1115, 1999.

[2] K. S. Pollard and M. J. van der Laan. New methods for identifying significant clusters in gene expression data. *Joint Statistical Meetings - Biometrics Section-to include ENAR & WNAR*, page 2714 – 2719, 2002.

[3] H. Peng, F. Long, M. B. Eisen, and E. W. Myers. Clustering gene expression patterns of fly embryos. *IEEE, ISBI*, page 1144 – 1147, 2006.

[4] T. Nguyen, J. Mattick, Q. Yang, M. Orman, M. Ierapetritou, F. Berthiaume, and I. Androulakis. Bioinformatics analysis of transcriptional regulation of circadian genes in rat liver. *BMC Bioinformatics*, 15(83), 2014.

[5] W. De Mulder, M. Kuiper, and R. Boel. Clustering of gene expression profiles: creating initialization-independent clusterings by eliminating unstable genes. *Journal of Integrative Bioinformatics*, 7(3)(134), 2010.

[6] A. Feltus, S. Ficklin, S. Gibson, and M. Smith. Maximizing capture of gene co-expression relationships through pre-clustering of input expression samples: an arabidopsis case study. *BMC System Biology*, 7(44), 2013.

[7] R. Duda, P. Hart, and D. Stork. *Pattern Clasification and Scene Analysis*. Wiley-Interscience; 2 edition (November 9, 2000), 2 edition, November 2000.

[8] D. Chiang, P. Brown, and M. Eisen. Visualizing associations between genome sequences and gene expression data using genome-mean expression profile. *Bioinformatic*, 17, 2001.

[9] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3), 1967.

[10] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354, 1983.

[11] L. Dalton, V. Ballarin, and M. Brun. Clustering algorithms: On learning, validation,performance, and applications to genomics. *Current Genomics*, 10(6):430–445, 2009.

[12] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987.

[13] F. Azuaje. A cluster validity framework for genome expression data, bioinformatics. *Bioinformatics*, 2002.

[14] N. Bolshakova and F. Azuaje. Cluster validation techniques for genome expression data. *Signal Processing - ELSEVIER*, 2003.

[15] J. Pearson and et al. Identification of the genetic basis for complex disorders by use of pooling-based genomewide single nucleotide polymorphism association studies. *The American Journal of Human Genetics*, 80:126–139, 2007.

[16] J. Hua, D. Craig, M. Brun, J. Webster, W. Tembe, K. Joshipura, M. Huentelman, E. Dougherty, and D. Stephan. Sniper-hd: improved genotyped calling accuracy by an expectation-maximization algorithm for high-density snp arrays. *Bioinformatic*, 23(1):57–63, 2007.

[17] F. Azuaje and N. Bolshakova. *Clustering genomic expression data: design and evaluation principles*, chapter 13. London: Springer Verlag, berrar, dubitzky edition, 2002.

[18] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, 2010.

[19] M. Brun, C. Sima, J. Hua, J. Lowey, B. Carroll, E. Suh, and E. Dougherty. Model-based evaluation of clustering validation measures. *Pattern Recognition*, 2007.

[20] G. Abras, J. Pastore, M. Brun, and V. Ballarin. Detección de conjuntos significativos de genes via silohuette. *CAIS 2010, 1er Congreso Argentino de Informática y Salud, 38 JAIIO*, August 30 to September 3, Buenos Aires, Argentina 2010.

[21] A. Fred and A. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Trans Pattern Anal Mach Intell.*, 27(6):835–50, 2005.

[22] A. Topchy, A. K. Jain, and W. Punch. Combining multiple weak clusterings. *Third IEEE International Conference on Data Mining*, pages 331–338, 2003.

[23] Z. Lu, Y. Peng, and J. Xiao. From comparing clusterings to combining clusterings. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. Institute of Computer Science and Technology, Peking University, Beijing.*, 2008.

[24] J. Ghosh A. Strehl. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[25] V. L. Boyartchuk, K. W. Broman, R. Mosher, S. D'Orazio, M. Starnbach, and W. F. Dietrich. Multigenetic control of listeria monocytogenes susceptibility in mice. *Nature Genetics*, 27:259–260, 2001.

[26] K. W. Broman, V. L. Boyartchuk, and W. F. Dietrich. Mapping time-to-death quantitative trait loci in a mouse cross with high survival rates. *Technical Report MS00-04, Department of Biostatistics, Johns Hopkins University*, Mayo 2000.

[27] J. Pihlajamäki, T. Boes, and Eun-Young Kim et al. Thyroid hormone-related regulation of gene expression in human fatty liver. *J Clin Endocrinol Metab*, 94(9):3521–3529, Septiembre 2009.

[28] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, and Q. Morris. Genemania: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, 9, 2008.

[29] M. Rebhan, V. Chalifa-Caspi, J. Prilusky, and D. Lancet. Genecards: A novel functional genomics compendium with automated data mining and query reformulation support. *Bioinformatics*, 14:656–664, 1998.