Gene–Gene Interaction based Clustering method for Microarray Data

Norberto Díaz–Díaz, Francisco Gómez–Vela, Jesús Aguilar–Ruiz
School of Engineering. Pablo de Olavide University.
Seville, Spain
{ndiaz, fgomez, aguilar}@upo.es

Jorge García–Gutiérrez
Department of Computer Science, University of Seville
Seville, Spain
jgarcia@lsi.us.es

***Abstract*—In this paper, we propose a greedy clustering algorithm to identify groups of related genes and a new measure to improve the results of this algorithm. Clustering algorithms analyze genes in order to group those with similar behavior. Instead, our approach groups pairs of genes that present similar positive and/or negative interactions. In order to avoid noise in clusters, we apply a threshold, the neighbouring minimun index(λ), to know if a pair of genes have interaction enough or not. The algorithm allows the researcher to modify all the criteria: discretization mapping function, gene–gene mapping function and filtering function, and even the neighbouring minimun index, and provides much flexibility to obtain clusters based on the level of precision needed. We have carried out a deep experimental study in databases to obtain a good neighbouring minimun index, λ. The performance of our approach is experimentally tested on the yeast, yeast cell-cycle and malaria datasets. The final number of clusters has a very high level of customization and genes within show a significant level of cohesion, as it is shown graphically in the experiments.**

***Keywords*-clustering; microarray analysis;**

I. Introduction

In any biologic process, cells and genes in particular play an important role which can be measured by their different levels of expression. These levels depend on the type of process, on the stage, and on the experimental condition that is analyzed. The knowledge about these, under a specific situation, helps to understand the function that genes play in a particular biological process.

Current works accomplished by researchers in the Bioinformatic field, like SAGE [19] for measuring gene expression, or like [9], [13] to store this gene expression in structure denominated microarray, make possible the simultaneous study of numerous genes under different conditions. Many different approaches have been applied to analyze this structure, including principal component analysis [22] as well as supervised [2] and unsupervised [6], [11], [14], [15], [17] learning. In unsupervised learning, clustering techniques [7] are used to identify groups of genes that show the same expression pattern under different conditions.

Tavazoie et al. [17] applied the k–means algorithm to find clusters in yeast data. In Luo et al. [10], we can find many hierarchical clustering (HC) examples applied in genomic research. Lately, Wang et al. [20] developed a new technique to analyze methilation data. Wrobel et al. [21] used Pearson to obtain a HC based on a tree structure view from GO. In Sharan et al. [14] graph–theoretic and statistical techniques were used to identify tight groups of highly similar elements. In Speer et al. [15] a memetic algorithm is presented, i.e., a genetic algorithm combined with local search -based on a tree representation of the data - for clustering gene expression data. With this aim, in Jiang et al. [6] is explored a novel type of gene–sample–time microarray data sets, which records the expression levels of various genes under a set of samples during a series of time points. Recently, a graph based algorithm has been proposed to generate clusters of genes with similar expression profiles by Huttenhower et al. [5]. Hao et al. [4] have used Bayesian information criteria to estimate the correlation between gene expression levels. In this work a preprocess step has been carried out, the Gausian model is used to describe the data. Currently, the Gausian model is also applied in [12] to estimate the parameters of different clustering based on models which utilize the factor analysis covariance structure.

All of these methods are based on the idea of grouping those genes that show the same behavior. In this work, we propose a clustering algorithm to identify groups of related genes based on the idea of clustering pair of genes which present the same type of interaction. To avoid the posible noise when we work with this algorithm, we introduce a new measure, the neighbouring minimum index, extracted from different experiments on databases. In this way, we use this measure to decide if a pair of genes has enough interaction in early steps of our algorithm. We have tested our algorithm and our new measure on malaria [8], yeast cell-cycle [16] and yeast [17].

In broad outlines, the remainder of the paper is organized as follows. In section II, the characteristics of our approach are detailed. Later in Section III, we describe the results of our experiments. Finally, the most interesting conclusions are summarized in Section IV.

II. Description

The clustering process presented in this paper, named Interclus, can be divided into four steps: encoding of each gene expression (*segmentation*), representation of the interaction of every two genes (*gene–gene interaction*), filtering of most representative interactions (*filtering*), and clustering interactions (*neighborhood–based clustering*). These steps are depicted in Figure 1 and they are described in detail in the next subsections.

A. Segmentation

The first step addresses the segmentation of each gene expression level. Due to the fact these levels are represented by numerical values, the segmentation is done by discretizing the range of values obtaining a new matrix(discretized matrix) with the same dimension that stores segmented values. In this way, different labels are obtained according to the gene expression level under particular stimulus (experimental condition). However, the discretization is local, i.e., the same expression level for two different genes might transform into different labels.

To carry out the discretization, we need to define an alphabet Ω, which is used to provide labels for the mapping, and a mapping function α, which is used to assign labels from Ω to the numerical values. The definition of Ω and α is provided by the user: characters for Ω and a discretization mapping table for α, in which the user can also make use of symbols ∞, μ and σ, standing for *infinite*, *mean* and *standard deviation*. Any expression that uses these special symbols is valid, together with arithmetical operators and numbers. For instance, in Figure 1, the first step transforms the gene expression level matrix into a discretized matrix by using the discretization mapping α, defined over a three–symbol alphabet $\Omega = \{I, M, E\}$. If the gene expression level is in $(-\infty, \mu - \sigma)$ then the label "I" is assigned (inhibited); if it is in $[\mu - \sigma, \mu + \sigma]$, then the label is "M" (medium); and finally, if it is in $(\mu + \sigma, +\infty)$, then "E" (expressed). An expression like $\mu + 0.5\sigma$ is also feasible, and any number of labels as well.

Note that although we use values like μ or σ, these values are different for each gene, so the discretization is local. A value of 0.6 for a gene can mean "expressed", and perhaps "inhibited" for another one, where both states translate further into labels.

B. Gene–Gene Interaction

Once each gene expression level has been labelled, we will focus on the interaction between every pair of genes. Firstly, another alphabet Π is needed to assign a label to any possible combination of gene pairs. For example, we might be interested in differentiating the interaction *inhibited–expressed* from the interaction *expressed–expressed*. In general, the size of the set Π is, at maximum, the square of the size of the set Ω, although usually should be lower. In Figure 1, it is shown in the first step that $|\Omega| = 3$, and in the second step, the gene–gene interaction mapping has exactly 9 combinations, but the size of the alphabet Π is 5, corresponding to {Z,S,P,N,Q}. In this example, Z stands for *null*, S for *similar*, P for *positive*, N for *negative*, and Q for *both expressed*. The interaction mapping function β is also defined by the user, as a mapping table, $\beta : \Omega \times \Omega \rightarrow \Pi$.

As the microarray has M genes and N experiments, for each gene, $M - 1$ interactions with the remaining genes are needed. In short, there will be $M \times (M - 1)$ interactions. The left–hand side of Figure 2 represents the discretized matrix obtained after the first step, in which rows mean experiments and columns mean genes. The values D_{ij} of a specific row and column are discrete, belonging to the alphabet Ω. To the right, any possible pair of different genes is enumerated in columns. In general, gene i can interact with other $M - 1$ genes. The value $I_{ij,k}$ of a row k and a column represents the symbol from the alphabet Π obtained after analyzing the two genes i and j involved in the interaction under the experiment k.

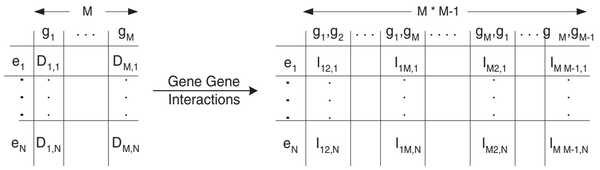

M
g1 ... gM
e1 D1,1 DM,1
eN D1,N DM,N
Gene Gene Interactions
M * M-1
g1,g2 ... g1,gM ... gM,g1 ... g M,gM-1
e1 I12,1 I1M,1 IM2,1 IM M-1,1
eN I12,N I1M,N IM2,N IM M-1,N

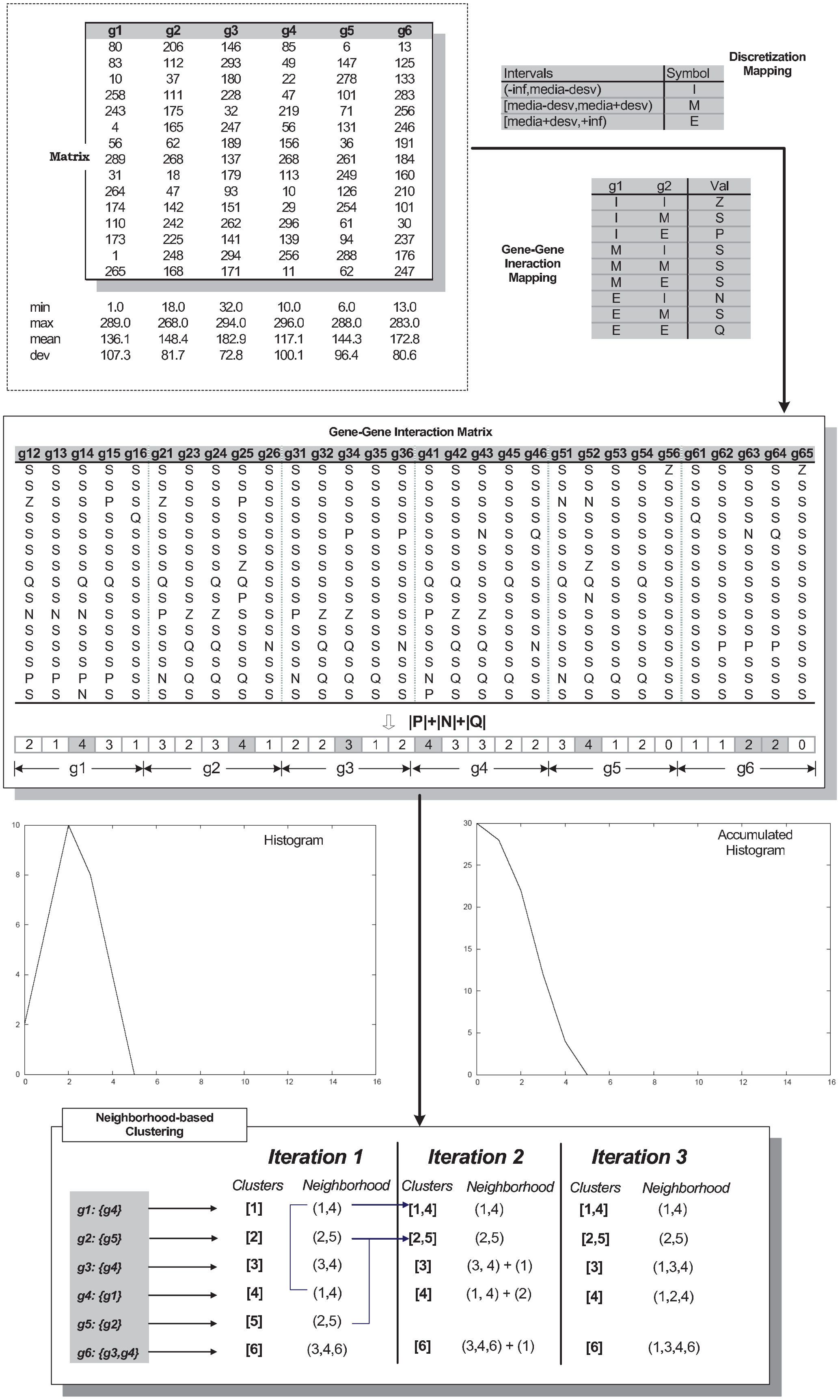
Figure 2. Gene-Gene Interactions

The new matrix M'' encodes the information of all possible interactions, although not every one might be interesting. For example, in Figure 1, we see in the table generated by the second step that many columns have only the symbol "S", which means similar, i.e., there is no significant up– or down–regulation in this case. The last column shows that genes 6 and 5 have similar behavior, so its interaction is not relevant. In this way, we might withdraw much irrelevant information if we were able to select the most interesting patterns in columns. That is the aim of the third step, described in the next subsection.

C. Filtering

The fact that two genes are inhibited under most or all of the experimental conditions, has no biological importance. Therefore, this situation can be easily ignored. When two genes are both expressed under most or all of the experimental conditions, that might have biological meaning. In fact, many studies only focus on this aspect: the interaction expressed–expressed. In this work, we are also interested in other cases: for example, when most of the time an inhibited gene is related to an expressed gene, and vice verse. And this situation is especially interesting when the complementary is true as well, i.e., if gene 1 is expressed then gene 2 is inhibited and if gene 1 is inhibited then gene 2 is expressed. The last situation is more difficult to detect and is one of the main goals in this work.

Another interesting issue is that what means "most of the time" for a pair of genes may not have the same meaning for another pair. This gives some clues about the



	g1	g2	g3	g4	g5	g6
	80	206	146	85	6	13
	83	112	293	49	147	125
	10	37	180	22	278	133
	258	111	228	47	101	283
	243	175	32	219	71	256
	4	165	247	56	131	246
	56	62	189	156	36	191
Matrix	289	268	137	268	261	184
	31	18	179	113	249	160
	264	47	93	10	126	210
	174	142	151	29	254	101
	110	242	262	296	61	30
	173	225	141	139	94	237
	1	248	294	256	288	176
	265	168	171	11	62	247
min	1.0	18.0	32.0	10.0	6.0	13.0
max	289.0	268.0	294.0	296.0	288.0	283.0
mean	136.1	148.4	182.9	117.1	144.3	172.8
dev	107.3	81.7	72.8	100.1	96.4	80.6

Discretization Mapping

Intervals	Symbol
(-inf,media-desv)	I
[media-desv,media+desv)	M
[media+desv,+inf)	E

Gene-Gene Ineraction Mapping

g1	g2	Val
I	I	Z
I	M	S
I	E	P
M	I	S
M	M	S
M	E	S
E	I	N
E	M	S
E	E	Q

Gene-Gene Interaction Matrix

g12	g13	g14	g15	g16	g21	g23	g24	g25	g26	g31	g32	g34	g35	g36	g41	g42	g43	g45	g46	g51	g52	g53	g54	g56	g61	g62	g63	g64	g65
S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	Z	S	S	S	S	Z
S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Z	S	S	P	S	Z	S	S	P	S	S	S	S	S	S	S	S	S	S	S	N	N	S	S	S	S	S	S	S	S
S	S	S	S	Q	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	Q	S	S	S	S
S	S	S	S	S	S	S	S	S	S	S	S	P	S	P	S	S	N	S	Q	S	S	S	S	S	S	S	N	Q	S
S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
S	S	S	S	S	S	S	S	Z	S	S	S	S	S	S	S	S	S	S	S	S	Z	S	S	S	S	S	S	S	S
Q	S	Q	Q	S	Q	S	Q	Q	S	S	S	S	S	S	Q	Q	S	Q	S	Q	Q	S	Q	S	S	S	S	S	S
S	S	S	S	S	S	S	S	P	S	S	S	S	S	S	S	S	S	S	S	S	N	S	S	S	S	S	S	S	S
N	N	N	S	S	P	Z	Z	S	S	P	Z	Z	S	S	P	Z	Z	S	S	S	S	S	S	S	S	S	S	S	S
S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
S	S	S	S	S	S	Q	Q	S	N	S	Q	Q	S	N	S	Q	Q	S	N	S	S	S	S	S	S	P	P	P	S
S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
P	P	P	P	S	N	Q	Q	Q	S	N	Q	Q	Q	S	N	Q	Q	Q	S	N	Q	Q	Q	S	S	S	S	S	S
S	S	N	S	S	S	S	S	S	S	S	S	S	S	S	P	S	S	S	S	S	S	S	S	S	S	S	S	S	S

⇩ **|P|+|N|+|Q|**

2	1	4	3	1	3	2	3	4	1	2	2	3	1	2	4	3	3	2	2	3	4	1	2	0	1	1	2	2	0
g1					g2					g3					g4					g5					g6				

Neighborhood-based Clustering

	Iteration 1		Iteration 2		Iteration 3	
	Clusters	*Neighborhood*	*Clusters*	*Neighborhood*	*Clusters*	*Neighborhood*
g1: {g4}	[1]	(1,4)	[1,4]	(1,4)	[1,4]	(1,4)
g2: {g5}	[2]	(2,5)	[2,5]	(2,5)	[2,5]	(2,5)
g3: {g4}	[3]	(3,4)	[3]	(3, 4) + (1)	[3]	(1,3,4)
g4: {g1}	[4]	(1,4)	[4]	(1, 4) + (2)	[4]	(1,2,4)
g5: {g2}	[5]	(2,5)				
g6: {g3,g4}	[6]	(3,4,6)	[6]	(3,4,6) + (1)	[6]	(1,3,4,6)

Figure 1. INTERCLUS steps. First and second steps: definition of the discretization mapping function and the gene-gene interaction mapping to obtain the gene-gene interaction matrix. Third steps: selection of gene-gene interactions that satisfy the filtering criterion. Fourth step: neighborhood-based clustering

strength of interactions, and provides us a specific criterion for each gene regarding the remainder. Therefore, although the filtering function is global, the value provided by the filtering function might be different for each gene.

To explain the use of filtering, we have to define $C_{i,F}$. It denotes the conditions established for the g_i–interactions using the filter F, and $S_{C_{i,F}}$ represents the subset of genes whose interactions satisfy the condition $C_{i,F}$. As explained earlier, for the example in Figure 1, the condition $C_{1,F}$ would be $max(|P| + |Q| + |N|) = 4$, but $C_{3,F}$ would be $max(|P| + |Q| + |N|) = 3$.

In the filtering algorithm , L_F denotes the list of all the subsets $S_{C_{i,F}}$. That is, $L_F = \{S_{C_{1,F}}, S_{C_{2,F}}, ..., S_{C_{M,F}}\}$. After this process, the filtering algorithm will generate the list of subsets of genes related to each one, if exists. In Figure 1 is provided, in the third step, the list of four subsets of genes, each of them with only one gene, by using the filter $max(|P| + |Q| + |N|)$.

Also, in this filtering process we establish a minimum threshold, named neighbouring minimun index (λ). This value will have been satisfied for each $C_{i,F}$, so that if the condition established for g_i-interactions do not satisfy it, $S_{C_{i,F}}$ will be empty and, therefore, it will not be part of L_F. In this way, we manage to give greater power to the filter function, since it is possible to select those gene interactions that fulfil the filtering criterion a minimum number of times. The neighbouring minimun index depends on the particular database and needs a preprocess. We can established its value using a histogram from accumulated frequencies and selecting the best value to apply as threshold. In Figure 3, we can see a graphical representation of the histogram. In X-axis, we have the possible values of the choosen filter. In Y-axis, we have the total number of times a given value corresponds a gen-gen comparision filter value. In section III, we carry out an exhaustive study that shows how to find a good λ.

D. Neighborhood–based clustering

Once the relevant interactions between each pair of genes have been obtained, it is time to cluster them. The clustering algorithm, named SNN (Similar Nearest Neighbor) [1], is based on the similarity of groups, instead of analyzing pairs of elements. It builds clusters by grouping genes whose neighbors are similar. SNN starts considering each gene as a separate cluster and at each step merges clusters which have exactly the same neighbors. Thus, the concept of neighborhood is redefined to handle correctly with clusters of neighbors.

Definition 1 (Neighborhood of a gene). *The neighborhood $N_g(i, F)$ of a gene g_i using the Filter F, is the set of genes whose amount of relevant interactions with regards to the gene i fulfils the condition $C_{i,F}$.*

$$N_g(g_i) = S_{C_i} \tag{1}$$

Algorithm 1 STEP–4 SNN

```
INPUT $L_F$: List of gene subsets
OUTPUT RSC: Set of Clusters
begin
  $SC := \theta$
  for all gene $g_i$ do
    $RSC[i] := \{g_i\}$
  end for
  repeat
    for all cluster $C_h \in RSC, 1 \leq h \leq |RSC|$ do
      $NSC[h] := N_c(C_h)$
    end for
    $SC := RSC$
    $RSC := Reduction(SC, NSC)$
  until $SC = RSC$
end
```

Algorithm 2 Reduction

```
INPUT C: Set of Cluster
  NSC: Neighbor Set of Cluster
OUTPUT R: Reduced set of clusters
begin
  $R := C$
  for all pair $(i, j)$, with $1 \leq i \leq j \leq |C|$ do
    if $S[i] = S[j]$ then
      $R[i] := R[i] \bigcup C[j]$
      remove $R[j]$
    end if
  end for
end
```

Definition 2 (Neighborhood of a cluster). *The neighborhood $N_c(C, F)$ of a cluster c (*cluster neighborhood*) using the Filter F, is the set formed by all the neighborhoods of each gene belonging to the cluster C.*

$$N_c(C) = \bigcup_{g \in C} N_g(g) \tag{2}$$

Once every necessary definition to support the algorithm at this step have been presented, we will describe the code depicted in Algorithm 1. The input parameter is L_F, containing in each position i the neighbors of g_i. And the output parameter is RSC, the reduced set of clusters, where each one comprises a group of genes. SC is an auxiliary set of clusters and RSC is initially set with clusters containing only one gene. The process is repeated until RSC has no change at an iteration. The neighborhood of every cluster is calculated in order to analyze the possible reduction of the set of cluster, task done by the Reduction function (Algorithm 2). The reduction of a set of cluster follows the next criterion: two clusters are joined if both have exactly the same neighborhood. We are aware of the restrictive character of this criterion and a relaxation of it is considered among our future research directions.

III. EXPERIMENTS

In this section, we address the evaluation of the performance of our approach, which is experimentally tested on the yeast [17], yeast cell-cycle [16] and malaria [8] datasets.

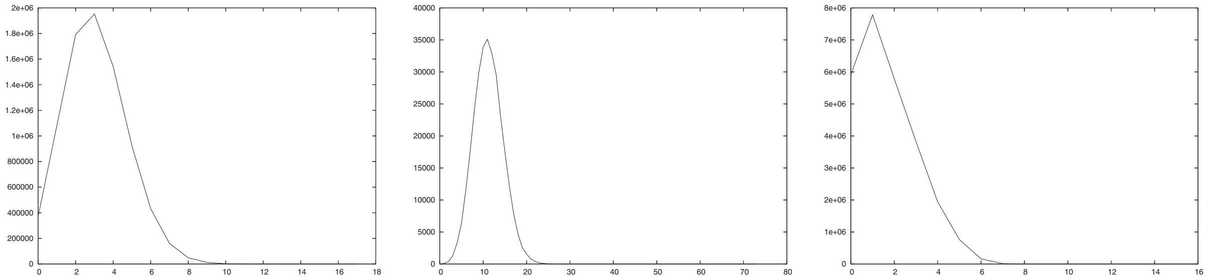


Figure 3. Resulting histograms to obtain neighbouring minimun indexes for yeast, yeast cell-cycle and malaria datasets for Q-filter.

The yeast dataset has information on 2884 genes under 17 different experimental conditions. The yeast cell-cycle dataset has information on 546 genes under 73 condtions. The malaria dataset has information on 5118 genes and 16 conditions. We have used Expander [18] to validate our results and to show the relevance of the clusters obtained.

In Table I it is shown the discretization mapping we have used in the experiments. The symbols μ_i and σ_i denote the mean and the standard deviation, respectively, of the expression levels of g_i under the whole set of experiments. Thus, the g_i expression level under e_k will be labeled as **I** (inhibited) if it belongs to $(-\infty, \mu_i - 0.25\sigma_i]$, as **M** (middle) if it belongs to $(\mu_i - 0.25\sigma_i, \mu_i + 0.25\sigma_i)$, or as **E** (expressed) if it belongs to $[\mu_i + 0.25\sigma_i, +\infty)$.

Table I
DISRETIZATION MAPPING α

Intervals	Ω
$(-\infty, \mu_i - 0.25\sigma_i]$	I
$(\mu_i - 0.25\sigma_i, \mu_i + 0.25\sigma_i)$	M
$[\mu_i + 0.25\sigma_i, +\infty)$	E

The alphabet Π, used in the experiments to encode each pair of gene–gene interaction, and the interaction mapping function β are shown in Table II. Highly relevant interactions are those where genes change their state from inhibited to expressed (P) or from expressed to inhibited (N) or from expressed to expressed (Q).

Table II
GENE–GENE INTERACTION MAPPING FUNCTION β.

Ω	Ω	Π	Ω	Ω	Π	Ω	Ω	Π
I	I	Z	I	M	S	I	E	P
M	I	S	M	M	S	M	E	S
E	I	N	E	M	S	E	E	Q

Making diverse experiments with datasets we have realized when we change λ value, the biggest clusters do not change in most of times. In opposite of that, the smallest clusters, with size two or three, have a very high variability. This makes us think the biggest clusters in any biological dataset and the most interesting too, have a very high level of interaction and it will be studied in future. We use graphics as in Figure 3 to discover the most interesting value for the neighbouring mininum index, λ. In Table III, we can see the kind of results we obtain when we vary the λ threshold for yeast dataset. These clusters are ordered decreasingly according to their sizes. The dimension of each cluster will be shown at column 'Size'. The column 'Number' represents the number of cluster which have been obtained with that size. For example, the size of the biggest cluster obtained using $\lambda = 9$ is 31 and two clusters exist with that dimension. In this case, the clusters with size 31 do not change in any gen. It is very important the fact that the three studied datasets show the same behavior. The most accurate value for λ will be always the lowest value, but it increases computational costs in comparison with higher values. If we are interested only in the biggest clusters, we will always choose a high value. With the values in Table III, we can see the best value is 6 because is the most well-balanced threshold.

Table III
THE BIGGEST CLUSTERS FROM THE OBTAINED RESULTS USING THE YEAST DATASET WITH $\lambda = 3$, $\lambda = 6$ AND $\lambda = 9$.

	$\lambda = 3$		$\lambda = 6$		$\lambda = 9$	
	Number	Size	Number	Size	Number	Size
1^{st}	2	31	2	31	2	31
2^{nd}	2	17	1	17	1	8
3^{th}	1	16	1	16	1	7
4^{th}	1	13	1	13	1	6
5^{th}	1	11	1	8	3	5

The result of our approach with filter Q is summarized in Table IV. The used λ is 6 for malaria, 20 for yeast cell-cycle and 6 for yeast. Note that the λ values have been obtained in the same way as it was explained before.

Next, the results obtained are validated. Because we do not have enough space in this paper, we have chosen only one image to show the quality of our method. We could have chosen any other datasets or filtering criterion. The other clusters obtained with rest datasets, summarized in Table III, have the same quality.

The results on yeast dataset are shown in Figure 4. The figures were created using Expander [18] with a preprocessing (standardization and normalization). The expression level is represented with colors from green(inhibited) to

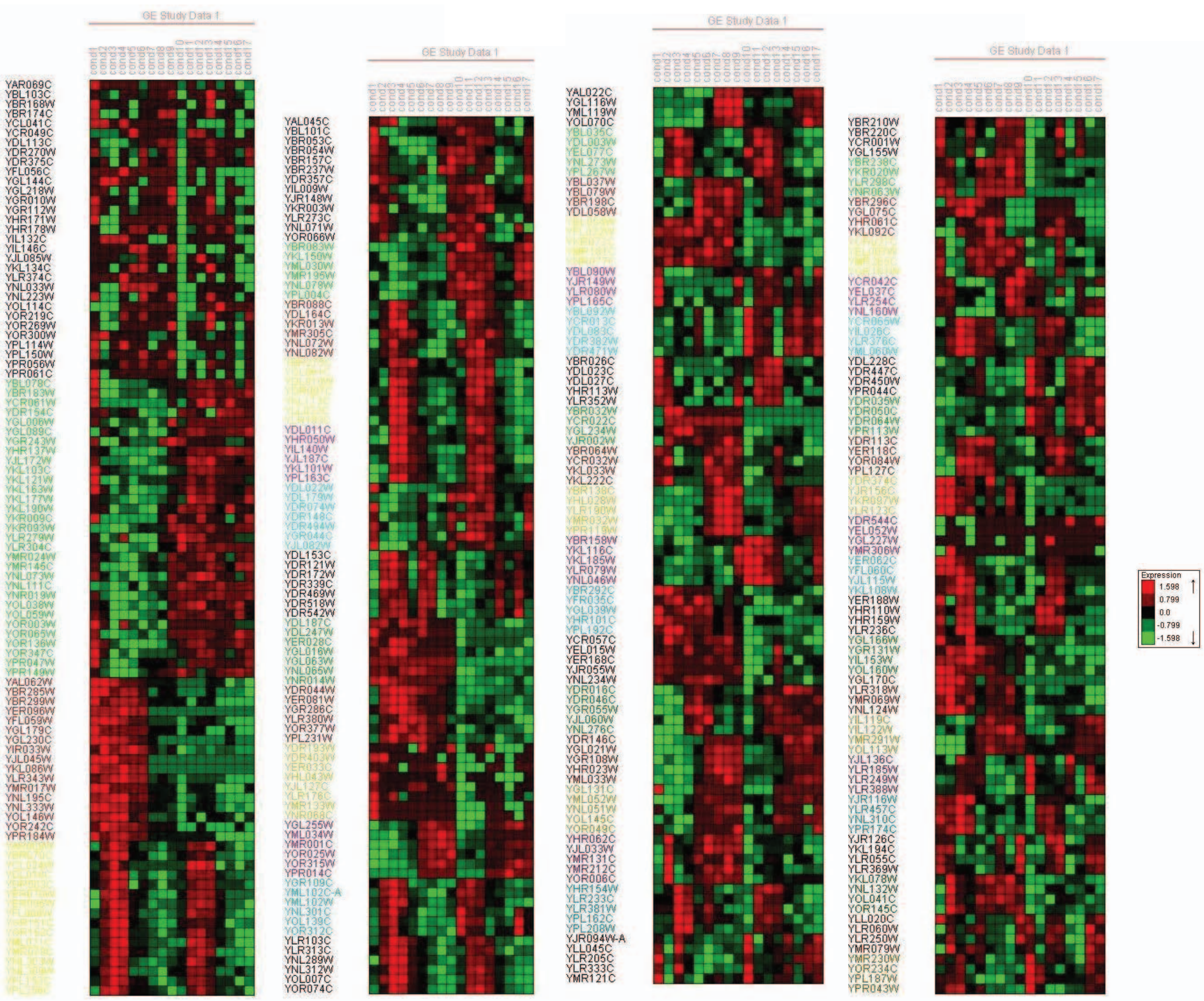


Figure 4. Resulting clustering for yeast with Q filter and $\lambda = 6$ and bigger size than 3.

Table IV
THE BIGGEST CLUSTERS FROM THE OBTAINED RESULTS USING THE YEAST, YEAST CELL-CYCLE AND MALARIA DATASETS WITH Q FILTER.

	yeast		malaria		yeast cell-cycle	
	Number	Size	Number	Size	Number	Cluster
1^{st}	2	31	1	364	1	12
2^{nd}	1	17	1	311	2	6
3^{th}	1	16	1	31	5	5
4^{th}	1	13	1	30	3	4
5^{th}	1	8	1	27	8	3

red(expressed) and the genes in a same cluster present the same color in their name. The figure shows that our clustering technique groups genes with very similar behaviour, as the colors are very alike. So, our approach can be validated in a positive way [3], since it presents a high level of compactness among genes within the same cluster, and a great separation among the different clusters obtained.

We have not given a comparative study with other clustering techniques because of our approach has a high level of customization and versatility and it could not be possible to make a fair comparison with other approaches that do not have or are not similar as ours.

IV. Conclusions

In this work, we propose a greedy clustering algorithm to identify groups of related genes and a new measure (λ) to improve its results. The approach is based on neighborhood of gene–gene interactions instead of on expression levels. One of the main features is that the algorithm allows the researcher to modify all the criteria: discretization mapping function, gene–gene mapping function and filtering function, and provides much flexibility to obtain clusters based on the level of precision needed. We have carried out a deep study in order to obtain good values for λ and the performance of our approach is experimentally tested on the yeast, yeast cell-cycle and malaria datasets and it has been validated with Expander. The final number of clusters is customizable and genes within show a significant level of cohesion, as it is shown graphically in the experiments.

References

[1] Jesus S. Aguilar, Roberto Ruiz, Jose C. Riquelme, and Raul Giraldez. Snn: A supervised clustering algorithm. *In Proceedings of IEA/AIE*, 2001.

[2] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science (New York, N.Y.)*, 286(5439):531–537, October 1999.

[3] Julia Handl, Joshua Knowles, and Douglas B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[4] Xiaolin Hao, Rui Jiang, and Ting Chen. Clustering 16S rRNA for OTU prediction: a method of unsupervised Bayesian clustering. *Bioinformatics*, 27(5):611–618, January 2011.

[5] Curtis Huttenhower, Avi Flamholz, Jessica Landis, Sauhard Sahi, Chad Myers, Kellen Olszewski, Matthew Hibbs, Nathan Siemers, Olga Troyanskaya, and Hilary Coller. Nearest Neighbor Networks: clustering expression data based on gene neighborhoods. *BMC Bioinformatics*, 8(1):250+, 2007.

[6] Daxin Jiang, Jian Pei, Murali Ramanathan, Chun Tang, and Aidong Zhang. Mining coherent gene clusters from gene-sample-time microarray data. In *Proc. Tenth ACM SIGKDD international*, pages 430–439, 2004.

[7] G. Kerr, H. J. Ruskin, M. Crane, and P. Doolan. Techniques for clustering gene expression data. *Computers in biology and medicine*, 38(3):283–293, March 2008.

[8] Karine G. Le Roch, Yingyao Zhou, Peter L. Blair, Muni Grainger, J. Kathleen Moch, J. David Haynes, Patricia De La Vega, Anthony A. Holder, Serge Batalov, Daniel J. Carucci, and Elizabeth A. Winzeler. Discovery of gene function by expression profiling of the malaria parasite life cycle. *Science (New York, N.Y.)*, 301(5639):1503–1508, September 2003.

[9] R. J. Lipshutz, S. P. Fodor, T. R. Gingeras, and D. J. Lockhart. High density synthetic oligonucleotide arrays. *Nat Genet*, 21(1 Suppl):20–24, January 1999.

[10] F Luo, K Tang, and L Khan. Hierarchical clustering of gene expression data. In *Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering.*, 2003.

[11] Patrick C. H. Ma and Keith C. C. Chan. Discovering clusters in gene expression data using evolutionary approach. In *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '03, pages 459–466, Washington, DC, USA, 2003. IEEE Computer Society.

[12] Paul D. McNicholas and Thomas B. Murphy. Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics*, 26(21):2705–2712, November 2010.

[13] M. Schena. Genome analysis with gene expression microarrays. *Bioessays*, 18(5):427–31, 1996.

[14] Roded Sharan and Ron Shamir. CLICK: A clustering algorithm with applications to gene expression analysis. In *Proceddings of the eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 307–316. AAAI Press, Menlo Park, CA, 2000.

[15] N. Speer, C. Spieth, P. Merz, and A. Zell. Clustering Gene Expression Data with Memetic Algorithms based on Minimum Spanning Trees. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, volume 3, pages 1848–1855. IEEE Press, 2003.

[16] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, 9(12):3273–97, Dec 1998.

[17] S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.

[18] Igor Ulitsky, Adi Maron-Katz, Seagull Shavit, Dorit Sagir, Chaim Linhart, Ran Elkon, Amos Tanay, Roded Sharan, Yosef Shiloh, and Ron Shamir. Expander: from expression microarrays to networks and functions. *Nature Protocols*, 5(2):303–322, January 2010.

[19] VE Velculescu, L Zhang, W Zhou, J Vogelstein, MA Basrai, DE Jr Bassett, P Hieter, B Vogelstein, and KW Kinzler. Characterization of the yeast transcriptome. *Cell*, 88(2):243–51, 1997.

[20] Z. Wang, P. Yan, D. Potter, C. Eng, T. Huang, and S. Lin. Heritable clustering and pathway discovery in breast cancer integrating epigenetic and phenotypic data. *BMC Bioinformatics*, 8(38), 2007.

[21] Gunnar Wrobel, Frederic Chalmel, and Michael Primig. goCluster integrates statistical analysis and functional interpretation of microarray expression data. *Bioinformatics*, 21(17):3575–3577, 2005.

[22] K. Y. Yeung and W. L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, September 2001.