IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:

IRIDIA, *Institut de Recherches Interdisciplinaires*
et de Développements en Intelligence Artificielle
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2007-016

The information provided is the sole responsibility of the authors and does not necessarily reflect the opinion of the members of IRIDIA. The authors take full responsability for any copyright breaches that may result from publication of this paper in the IRIDIA – Technical Report Series. IRIDIA is not responsible for any use that might be made of data appearing in this publication.

MetaClustering: Discovery of The Different Sample Clusterings in Gene Expression Data

David Venet[1] davenet@iba.k.u-tokyo.ac.jp

Hugues Bersini[2] bersini@ulb.ac.jp

Hitoshi Iba[1] iba@iba.k.u-tokyo.ac.jp

[1] IBA LAB, Post Box: 704, Graduate School of Frontier Sciences, The University of Tokyo, Kashiwanoha 5-1-5, Kashiwa-shi, Chiba 277-8561, Japan

[2] IRIDIA, Universite Libre de Bruxelles, Avenue Franklin Roosevelt 50, B-1050 Brussels, Belgium

Abstract

Clustering of the samples is a standard procedure for the analysis of gene expression data, for instance to discover cancer subtypes. However, more than one biologically meaningful clustering can exist, depending on the genes chosen. We propose here to group the genes in function of the clustering of the samples they fit. This allows to determine directly the different clusterings of the samples present in the data. As a clustering is a structure, genes belonging to the same group are functions of the same structure. Hence, the determination of groups of genes which support the same clustering could also be viewed as the detection of non-linearly linked genes. MetaClustering was applied successfully to simulated data. It also recovered the known clustering of real cancer data, which was impossible using the complete set of genes. Finally, it clustered together cell-cycle genes, showing its ability to group genes related in a non-linear way.

Keywords: clustering, gene expression data, bi-clustering

1 Introduction

Large amount of gene expression data are now publicly available. This led to the need to design algorithms this deal with this wealth of data. One of the most useful statistical analysis techniques for gene expression data is clustering. Clustering of the samples groups them together on the basis of their expression profile. It is used to reveal differences between groups of samples, like for instance between previously unidentified sub-types of cancer.

However, there can be more than one meaningful way to cluster the samples. For instance, in a study about tumors, samples could be clustered according to their pathological status or their inflammation level. If the complete data set, with all genes, is clustered, the result could be any of those clusterings, or a chimerical clustering where some samples are grouped based on one concept and others based on some other concept. Getz *et al.* [9] have shown that by clustering on only a subset of genes, different clusterings of the samples appear, each heaving its own biological interpretation.

The problem is to get the appropriate clustering for the application at hand. A first approach is to select a set of genes on which a well-defined clustering can be found [5, 6, 16]. Those techniques are based on the idea that some genes are irrelevant. But often the problem is the co-existence of different organizations of the data, and not the presence of noise. Genes that are irrelevant for one clustering can be relevant for another. The methods working by feature selection are not well adapted to such case.

Getz *et al.* [8, 9] proposed to search for tight groups of highly correlated genes, and to cluster the samples on those groups. This approach has two limitations: firstly, the clustering of the sample is done on only a handful of genes, which have essentially all the same values. Secondly, it only works for correlated genes, hence showing a linear dependence, while relationships that are more complex can exist. For instance, some genes oscillate with time during the cell cycle, and so display a similar profile. However, two oscillating genes with a 90° phase difference are not correlated.

Biclustering methods [2, 3, 4, 8, 12, 13, 15] search for groups of samples which appear on a subset of genes. As such, they can discover many different clustering of the samples. However, those methods typically obtain dozens of biclusters, which are all based on only a handful of genes. For this reason, those methods are well adapted to group the genes, proving to be very useful for gene function inference [13]. On the other hand, they are not well adapted to find the different clusterings of the samples as they find too many clusterings, based on only one gene pattern each. Furthermore, such clusterings are usually binary —a sample is in the bicluster or not— while more complex organizations are likely to exist.

We propose here to directly tackle the issue, by grouping the genes that support a similar clustering of the samples. The clusterings are used in a way similar to the prototypes in a k-means, and genes are moved to the group whose prototype they fit best. Doing this is akin to try to cluster the genes according to the clustering of the sample they fit, hence the name MetaClustering.

It is possible to view the results of the MetaClustering algorithm in a different light. A clustering could be understood as the assignation of a value to each sample. This value can be a discrete number, as in a k-means, or a complex structure, as in a hierarchical clustering. A gene fitting a clustering must be a function of those values. Since all genes fitting a clustering are function of the same values, they must depend on each other in some complex, possibly not univocal, way. This means that the determination of the different possible clusterings is a way to find groups of non-linearly related genes.

Different clustering algorithms could be used within this framework. A version based on average linkage hierarchical clustering and a version based on a neural network clustering (a k-means) are presented here.

2 Algorithm

As the MetaClustering clusters both the samples and the genes, it is necessary to precise the vocabulary to clarify the text. In the following *clustering* is defined as being a partition of the samples and *grouping* as being a partition of the genes. MetaClustering finds a grouping (on the genes), each group (of genes) defining a clustering (on the samples).

2.1 Hierarchical Clustering Version

This first version of the algorithm uses the average linkage hierarchical clustering to cluster the samples. The grouping of the genes is done in a manner similar to the k-means algorithm:

1. Start with some random group membership for all genes.
2. Calculate a clustering of the samples on each group.
3. Calculate the fit of each gene to each clustering.
4. Move each gene to the group whose clustering it fits best.
5. Repeat steps 2–4 until convergence occurs.

The number of groups is a user-defined input of the algorithm; it sets the trade-off between variance and bias, as in the k-means algorithm. If the number of groups is too small, then a group of genes might support more than one clustering. If the number of groups is too large, there is a risk of overfitting the data, obtaining small groups of genes which are determined mostly by the noise.

A measure of the fit of a gene expression pattern to a hierarchical clustering on the samples, i.e. a measure of the fit between a feature and a clustering, must be defined. The usual choice proposed in the literature is the cophenetic coefficient [11]. However, it is meant to compare various clusterings on the same data, not to estimate the relevance of a gene to a given clustering. We chose to define a new fitness measure which is similar to the average linkage hierarchical clustering algorithm. In that algorithm, at each step the two closest nodes are merged to form a new node. The distance $d(L,R)$ between two nodes L and R is defined as the mean of the distances between the leaves of each of those two nodes:

$$d\left(L,R\right)=\sum_{i}^{N_g}\frac{1}{s\left(L\right)s\left(R\right)}\sum_{k\in S(L)}\sum_{l\in S(R)}\left(m_{ik}-m_{il}\right)^2 \tag{1}$$

where N_g is the number of genes, $S(L)$ is the set of samples at the leaves of the node L, $s(L)$ is the cardinal of $S(L)$ and m_{ik} is the value of the gene i in the sample k. So, at each junction a certain criterion is minimized in a greedy fashion. This leads to the definition of the fitness $F(i,c)$ between a gene i and a clustering c:

$$F\left(i,c\right)=-\sum_{j=1}^{N_n}\frac{1}{s\left(L\left(j\right)\right)s\left(R\left(j\right)\right)}\sum_{k\in S(L(j))}\sum_{l\in S(R(j))}\left(m_{ik}-m_{il}\right)^2 \tag{2}$$

where N_n is the number of nodes in the clustering and $L(n)$ and $R(n)$ are the left and right children of node n. This fitness is a weighted sum of every sample differences. The weight is higher for samples closer in the tree, so the function indeed quantifies the fitness of a gene to a clustering. The quality of the solution Q_0 is the sum of the fitness of each gene:

$$Q_0=\sum_{i}^{Ng}F\left(i,C\left(G\left(i\right)\right)\right) \tag{3}$$

where $C(G(i))$ is the clustering calculated on the group $G(i)$ to which the gene i belongs. The algorithm should maximize Q_0 with respect to $G(i)$. An issue with the quality function (3) is that the fitness between a gene and the clustering calculated on its group can be influenced more by the gene itself than by the rest of the group. This effect is more pronounced in small groups. In order to really assess if a gene fits its group, a modified quality function is used:

$$Q=\sum_{i}F\left(i,C\left(G^*\left(i\right)\right)\right) \tag{4}$$

where $G^*(i)$ is the set of genes of the group $G(i)$, with the gene i excluded. With this modification, the quality function is computationally heavier but more meaningful. This also means that, for the calculation of the quality function, there are as many clusterings in each group as there are genes in the group. In practice, a few genes are excluded at the same time to speed up the calculations.

The direct maximization of (4) does not lead to satisfying results, firstly because it is very heavy to calculate and secondly because of the presence of numerous local maxima. Thus, the quality function is not directly maximized. A stochastic version is used instead. In that version, the following is done for each gene:

1. The clustering for the group to which the gene belongs is re-calculated after the removal of the gene.

2. The fitnesses between the gene and the clusterings of each group are calculated.

3. The gene is moved to the group whose clustering it fits best.

This version neglects the effect of the switching of a gene from one group to another on the quality of the other genes. This speeds the calculations and permits to avoid many local maxima.

Since the algorithm is not maximizing a global criterion, convergence is not guaranteed. However, because the algorithm is deterministic, Markovian and the search space is finite, it has to converge either to a fixed solution or to a cycle. If the number of genes or the number of groups of genes is small, then the cycles might be short enough to be detected. Otherwise, the algorithm can either be stopped after a determined number of iterations, or the quality (4) of the solutions can be monitored, and the algorithm stopped when no improvement is noted for a sufficient number of iterations. In our implementation, the algorithm was stopped after convergence, after detection of a cycle, or after a hundred iterations, whichever occurred first.

In order to reduce the variance of the calculation of the fitness, a few slightly perturbed clusterings are created by using cross-validation: $(K-1)$ Kth of the genes of the group are selected K times for the calculation of the fitness. The results are then averaged. This leads to a better and more stable estimation of the fitness of a gene to a group of genes.

2.2 Neural Networks Version

The two-level clustering presented can easily be written in neural network form. This is done by taking a neural network clustering algorithm —a k-means in this case— and by adding a first layer which dispatch the genes to one clustering or another.

In a k-means, an input vector $\mathbf{y}$ is compared to the weight neurons $\mathbf{w_k}$, which correspond to the clusters of the k-means. The outputs o_k of the k-means are:

$$o_k = \sum_i y_i w_{ik} \tag{5}$$

Clustering is obtained by considering that the outputs inhibit each other, so that the largest of the o_k is set to 1 and the others to 0. To turn this network into a MetaClustering, a set of k-means clustering are used, and a first layer is used to dispatch the data to the different k-means (Figure 1). The input vector $\mathbf{m}$, which is the gene expression profile of a sample, is on the left. The input values are dispatched to the different k-means by a first layer, the Z_i neurons, leading to the inputs $\mathbf{y_k}$ of the k-means clustering k:

$$y_{ik} = m_i z_{ik} \tag{6}$$

The inputs $\mathbf{y_k}$ are then used for the different clusterings performed by the neurons, leading to the output o_{jk}

$$o_{jk} = \sum_i y_{ik} w_{ijk} \tag{7}$$

As those remains k-means, for each j the highest o_{jk} is set to 1 and the remaining to 0.

The quality function for this network is simply the regular k-means quality function, summed on the different k-means. It can be written as

$$Q = \sum_{ikl} \left(y_{ik} - w_{ib(l,k)k} \right)^2 = \sum_{ikl} \left(m_{il} z_{ik} - w_{ib(l,k)k} \right)^2 \tag{8}$$

where $b(l,k)$ is the neuron which fires in the k-means k when the observation l is presented. In order to avoid trivial solutions, some normalization constraints must be added:

$$z_{ik} \in [0,1] \tag{9}$$

$$\sum_k z_{ik} = 1 \tag{10}$$

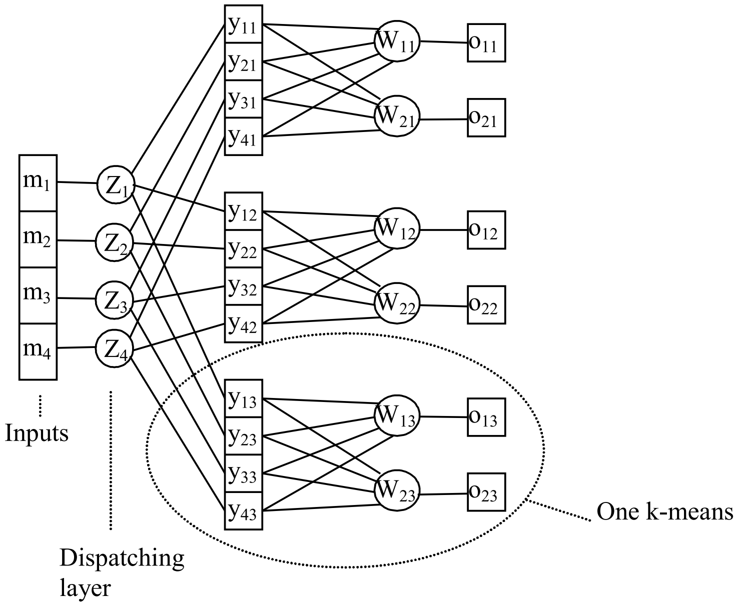

Figure 1: Neural network version of the MetaClustering algorithm, with four genes separated in three groups. The clusterings of the samples are two-group k-means. Circles represent neurons and boxes represent input/output values. m_i are the inputs, Z_i are the neurons which dispatch the values to the different k-means, y_{ij} are the input values for each k-means, W_i are the neurons performing k-means sample clustering and o_{ij} are the outputs of the network.

A quick calculation leads to the stochastic gradient:

$$\frac{\partial Q}{\partial w_{ib(,k)k}} = 2\left(w_{ib(l,k)k} - z_{ik} m_{il}\right) \tag{11}$$

$$\frac{\partial Q}{\partial z_{ik}} = 2\left(z_{ik} m_{il} - w_{ib(l,k)k}\right) x_{il} - \frac{2}{K} \sum_{x} \left(z_{ix} m_{il} - w_{ib(l,x)x}\right) m_{il} \tag{12}$$

where K is the number of k-means. This gradient is used for learning the weights.

After convergence of the network, the grouping of the genes can be found by using the **Z** neurons. The different clusterings obtained on each sample are the output of the network, o_{jk}.

3 Results

3.1 Simulated Data

An artificial data set including non-linearly linked genes has been created in order to show the power of MetaClustering compared to more usual methods. The data set consists of 20 genes and 50 samples. The 20 genes are organized in 4 groups of 5 genes each. In the first three groups, genes are linked together in a similar fashion. The last five genes are simply random noise.

For each of the first three groups, a random permutation **s** of the numbers 1 to 50 is drawn. This permutation gives a value from 1 to 50 to each sample. The genes $\mathcal{F}1$–$\mathcal{F}5$ are related to **s** as follow: $\mathcal{F}1 = \mathbf{s}$; $\mathcal{F}2 = (\mathbf{s} - 25)^2$; $\mathcal{F}3 = \sin(9\mathbf{s}/50)$; $\mathcal{F}4 = \sin(12\mathbf{s}/50)$; $\mathcal{F}5 = -\mathbf{s}$. The permutation is different for each of the first three groups. Each gene is centered and normalized, then Gaussian noise with a standard deviation of 0.3 is added and the resulting genes are centered and normalized again.

The random genes are drawn from a Gaussian distribution of unity standard deviation. This data set is constructed so that although the genes are related inside each group, correlation between some of them remains small. In particular, the second and third genes of each group ($\mathcal{F}2$, $\mathcal{F}3$) are hardly correlated to the other genes ($\mathcal{F}1$, $\mathcal{F}4$ and $\mathcal{F}5$) of their group. This data set is designed so that usual clustering methods are unable to group the genes meaningfully.

Table 1: Percentage of the simulated data sets in which hierarchical clustering using mutual information was able to find the expected structure.

Number of discrete levels	3	4	5	6
Single linkage	1%	64%	31%	20%
Average linkage	8%	70%	43%	38%
Complete linkage	2%	2%	10%	8%

A classical way to group together non-linearly related genes is to use a non-linear metric, mutual information being the most common choice. As mutual information can only be calculated on discrete data, each gene was discretized to 3, 4, 5 or 6 levels. Mutual information was then calculated between all genes, and the genes were clustered using one of the three classical hierarchical clustering algorithms: single linkage, average linkage or complete linkage. We then checked if the expected structure was uncovered. A clustering was considered as correct if it was possible to find 3 nodes such that the leaves of each node contained all the genes of one of the non-random group and no genes of any of the other non-random group. 100 simulated data sets were drawn. Results are summarized Table 1. Hierarchical clustering using mutual information was often able to find the right structure, but this was not always the case. Even in the best-case scenario (discretization in 4 levels and average linkage), only 70% of the data sets were correctly clustered. So, mutual information is not able to robustly recover the known structure.

Data sets were then MetaClustered. The algorithm was run twenty times for each data set, with different random initialization. The number of discrepancies between the known grouping and the obtained grouping were recorded, as well as the quality of the results as measured by (4). It is not expected for each run to give the right solution, as the algorithm is sensitive to its initialization. However, the solution with the highest quality, in the sense of (4), should be the correct one.

The hierarchical clustering version converged in 61% of the runs to the correct solution (see Table 2). The run with the best quality (4) was consistently the correct solution, showing the effectiveness of the algorithm. Results with the neural network version were slightly worse, as the correct solution was sometimes missed. This happened in 1 to 2% of the runs with a k-means layer using 4 to 6 groups.

Table 2: MetaClustering of simulated data. First line is the percentage of MetaClustering runs giving the expected result for each simulated data set. Second line is the percentage of data sets for which the highest quality MetaClustering gave the expected result.

	Hierarchical	Neural network version, k-means with				
		2 groups	3 groups	4 groups	5 groups	6 groups
MC with expected result	$61 \pm 4\%$	0%	$18 \pm 16\%$	$31 \pm 12\%$	$28 \pm 10\%$	$21 \pm 9\%$
Data sets with correct best MC	100%	0%	82%	98%	99%	99%

Those results are still much better than the ones obtained with the mutual information, but worse than the ones obtained with the hierarchical version of the algorithm. The last five genes, which are purely random, were split between the groups since MetaClustering always keeps all features. Their presence did not compromise the ability of the algorithm to correctly group the other genes.

In conclusion, MetaClustering proved to be able to robustly group genes which are non-linearly related, contrary to more classical approaches. The hierarchical clustering version is more effective. It leads more consistently to good solutions and is computationally more efficient. The hierarchical version is systematically used in the next applications.

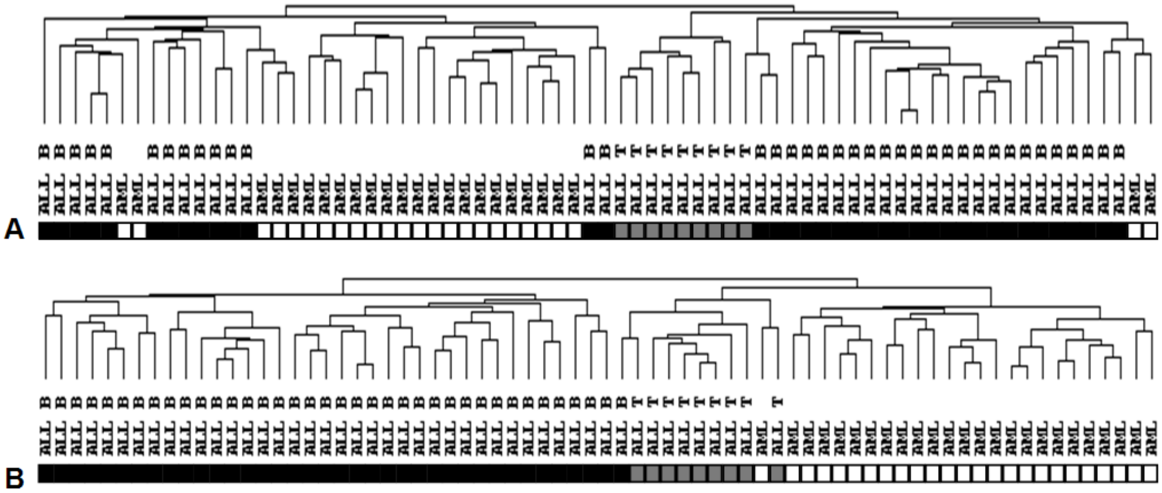

Figure 2: Clusterings obtained on the ALL/AML data. A: With all the genes. B: With the genes from a group determined by MetaClustering in three groups. AML: white; ALL-B:black; ALL-T: gray.

3.2 Real Data —Leukemia

Golub *et al.* [10] have studied with Affymetrix oligochips two types of leukemia: acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). This last type of leukemia can further be separated into T-lineage ALL and B-lineage ALL. They showed that the distinction between ALL and AML could be inferred directly from the data by clustering. We discovered that this was indeed true, but depended on the normalization and filtering scheme used and on the initialization of the clustering algorithm.

We clustered the normalized data using an average linkage hierarchical clustering algorithm (Figure 2A). Leaves were ordered [1], and the images displayed using TreeView [7]. Some parts of the clustering were close to the ALL/AML separation, but other parts seemed unrelated. Since hierarchical clustering is based on local similarities, some samples were merged because they were close on one set of genes, others because of another set of genes.

In the rest of the analysis, k-means clustering are used intensively on the groups obtained by MetaClustering. The quality of a k-means can be estimated using its objective function (called the k-means error here): the sum of the square distances between the samples and their cluster center. The k-means should also fit the expected labels, so δ is defined as the number of differences between a clustering and the expected labels (ALL/AML or ALL-T/ALL-B/AML, depending on the context).

Samples were clustered in two groups using k-means with random initialization. This was done 1000 times, leading to 324 different solutions. As shown Figure 3A, clusters similar to the ALL/AML separation ($\delta \leq 5$) were obtained in only 4% of the runs. As shown in Figure 3B, the k-means error did not point to the clusters closest to the ALL/AML separation. So the ALL/AML separation might be obtained through k-means, but if it was not known beforehand, it could be missed.

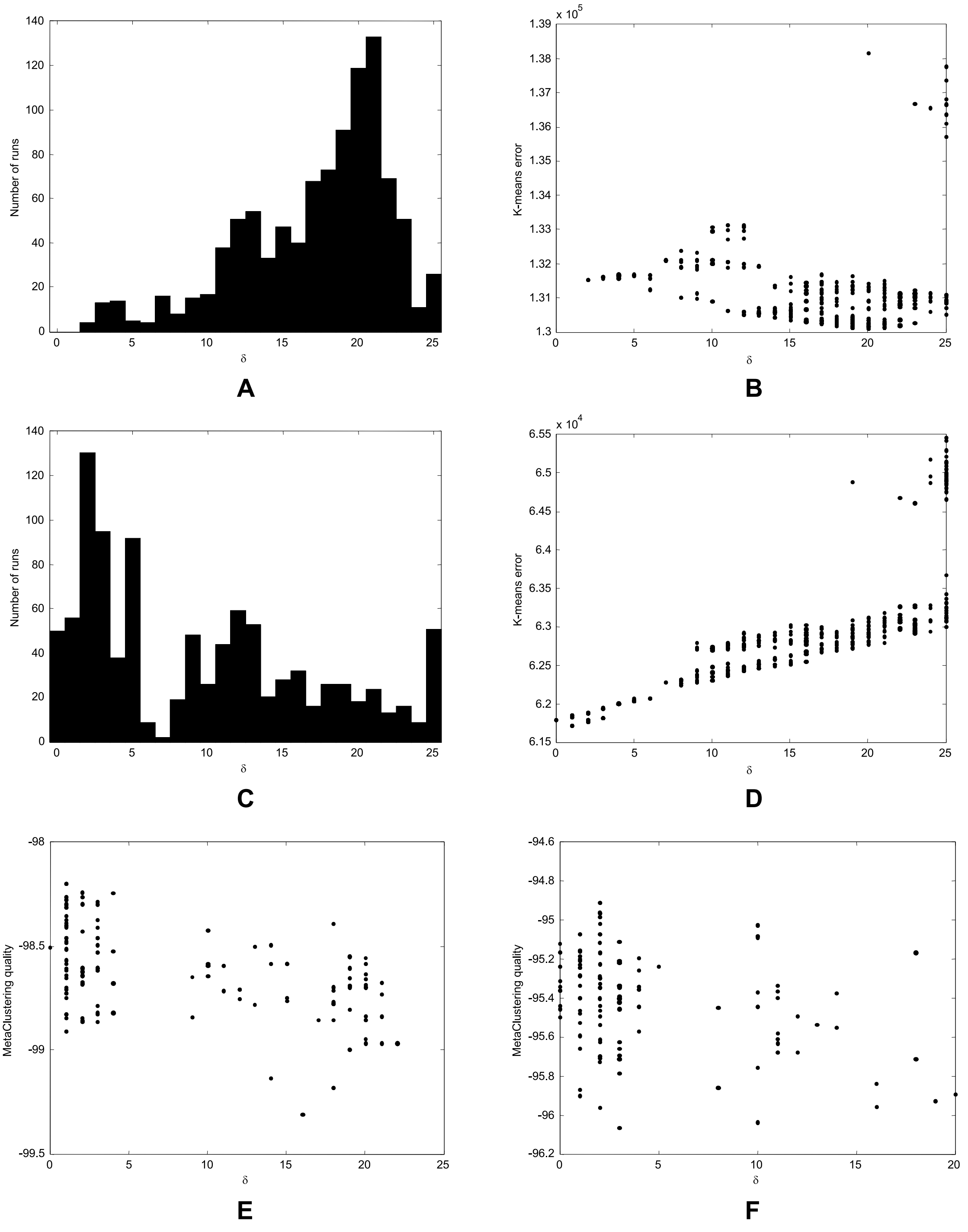

Figure 3: δ is the number of differences between the known separation and the clusters obtained by k-means. A: Histogram of δ for k-means clustering on all genes using different random initializations. B: Mean squares error of those clusterings. C and D are similar results obtained on one of the groups after MetaClustering in two groups E and F δ as a function of the MetaClustering quality. MetaClustering was done with two (E) or three (F) groups.

MetaClustering allows the determination of the different clusterings of the samples. As the ALL/AML separation is expected to be one of those clusterings, it should be prominent on one of the group of genes determined by the algorithm. Indeed, this was often the case after MetaClustering in three groups (e.g. Figure 3C). Furthermore, the k-means error did usually point to the ALL/AML separation (e.g. Figure 3D). In the following analysis, the k-means algorithm was always run 100 times, and the solution with the lowest error was kept.

Since different initializations of the MetaClustering can lead to different results, the algorithm was run 200 times with random initialization, with two and three groups. Among the runs with two groups, 61% had a δ below 6. With three groups, this fraction raised to 77%. It is possible to decide which of the MetaClustering runs should be chosen by using the quality function (4). The solution with the highest quality (4) had indeed a low δ (Figure 3E and 3F), showing that it is possible to use (4) to decide which MetaClustering run to keep. With the highest quality MetaClustering, δ was 1 with two groups and 2 with three groups. This means that, as with k-means, it may be necessary to perform a hundred or so runs with random initialization to pick the best one. Our results compare favorably with the technique of Xing and Karp [16], which used a feature selection algorithm and obtained a δ of 3. Getz *et al.* [9] have determined a cluster of 60 correlated genes on which they claim that a clustering close to the ALL/AML separation appears. Using a k-means on that group of genes, the result was disappointing as δ was 6. This shows the importance of using more than one pattern of genes for the clustering.

In the original publication— [10], a test was created to determine the most informative genes for the ALL/AML separation. 50 genes were selected using the same test. The group on which the ALL/AML separation appears indeed concentrated many of those genes, on average 74% in two-group MetaClustering and 70% in three-group MetaClustering. The MetaClustering run with the best quality (4) concentrated a larger number of those genes than average, that is 86% with two groups and 72% with three groups, showing again that (4) is an effective criterion to judge the quality of the MetaClusterings.

The separation of the samples in AML, T-ALL and B-ALL should also be found by clustering of the samples using one of the groups of genes obtained by MetaClustering. This was assessed by k-means clustering in 3 clusters. Again, clustering the whole data set led to a large δ, 14. However, after MetaClustering, results were closer to the expected separation. For the MetaClustering run with the highest quality, δ was 10 using two groups of genes and 3 using three groups. In this case, three groups seemed necessary to recover the known structure.

The genes selected by MetaClustering were clustered using a hierarchical clustering algorithm (Figure 2B). This clustering is much closer to the known separation than the one calculated on the whole set of genes (Figure 2A). There were three main clusters: AML, T-ALL and B-ALL. This explains the presence of good quality MetaClusterings in 3 groups which have a δ with the ALL/AML separation of 10 (Figure 3F): the k-means algorithm being biased towards equivalent-sized groups, a solution with one tight B-ALL cluster and one loose T-ALL and AML cluster may compare favorably to a solution with one tight AML cluster and one not-so-tight B-ALL and T-ALL cluster.

The meaning of the clustering obtained on the other groups of genes is harder to understand. On one group, samples coming from one of the sources (CALGB) are tightly clustered together. This corresponds to one of the clusters found by Getz *et al.* This shows that other clustering are indeed present in the data, and that would the determination of the source be the important parameter it could have been found by MetaClustering. The other groups were not intelligible with available biological information.

3.3 Real Data —Yeast Cell Cycle

Spellman *et al.* [14] analyzed cell cycle in yeast using microarrays. In those experiments, yeast cells were synchronized at a certain point in their cycle. They were then released and began to cycle while keeping their synchrony. The expression levels of many genes, the cell-cycle regulated genes, showed a periodic behavior. However, other genes showed different profiles, like for instance steady increase or decrease with time. Spellman *et al.* used a method based on Fourier transform to identify cell-cycle regulated genes. Since those genes are not all correlated, it is impossible to cluster them together using classical clustering algorithms like hierarchical clustering or k-means (results not shown). However, MetaClustering groups together genes which support the same organization of the data. Since cell-cycle genes have a specific periodic organization, they should be grouped together.

Data from one of the cell-cycle experiments were taken. Genes were MetaClustered in two groups. The first group contained genes which showed a periodical behavior, that is cell-cycle genes. The second group was not as coherent, but its main feature was large variations from one time point to the next. Clustering the samples using the genes of the second group led to a surprising two clusters result, one cluster comprising the odd time points (70 mins, 90 mins...) and the other the even time points (80 mins, 100 mins...). This seems to correspond at least partially to dye-swapping.

Spellman *et al.* have determined 800 genes to be cell-cycle regulated. Among the 569 of those genes which were analyzed by MetaClustering, 458 (80%) belonged to the first group. The remaining 20% of the genes seem to show large fluctuations from one time point to the next, which is a characteristic of the second group. On the other hand, 601 genes were grouped with the cell-cycle regulated genes while they were not considered as such by Spellman *et al.* As judged visually, a large part of those genes could indeed be considered as cell-cycle regulated (see supplement Figure 1 in [17]). Since every gene must be assigned one group, even if it does not fit any group well, the presence of a certain percentage of non-cell cycle regulated genes in the cell-cycle group was expected.

4 Conclusion

A new framework allowing uncovering the overlapping structures of the samples has been presented. The algorithm works as well for discrete structures (e.g. cancer type, as in the leukemia data) than for continuous structures (e.g. cell cycle phase, as in the yeast data). The outputs of the algorithm are groups of genes which have a similar structure of the samples. This means that any clustering algorithm can then be used on those groups of genes, be it hierarchical clustering, k-means, or anything else. This flexibility makes MetaClustering a powerful tool for the discovery of the different structures present in the data.

Two implementations were presented, one using hierarchical clustering and one using neural networks. Although the hierarchical clustering version was more effective, the neural network version could be considered as promising, as a few runs were still usually enough to find the expected solution. The k-means algorithm used is the simplest neural network clustering, and more complex types of clustering like self-organizing maps or fuzzy clustering could improve the result. The structure of the simulated data was also very different from the clear-cut clusters expected by a k-means, which lowered its effectiveness compared to hierarchical clustering. An issue with the neural network version is that the number of clusters in the k-means must be specified. When this number is too small, k-means are too coarse and the network is unable to uncover the real structure, when it is too large the algorithm can have some trouble converging.

This work could also be viewed as a means to perform feature selection: genes are selected so that each group gives a tight clustering. The main idea which allows for this feature selection is that all genes are informative, but not to answer the same question. Hence, genes can be selected according to the question asked, i.e. according to the clustering obtained on the samples. It could be possible to further select the genes, by excluding those which do not really fit any group. We are investigating this possibility.

MetaClustering was able to group together cell-cycle regulated genes in an unsupervised fashion, leading to results similar of those obtained using a specialized algorithm. Since cell-cycle regulated genes are not all correlated, this is a result that could not have been obtained by clustering algorithms based on pair-wise similarity (like hierarchical clustering) nor by algorithms based on prototypes defined in the original space (like k-means). This ability of MetaClustering to group non-linearly related features, or even discrete and continuous features, might be a promising path to non-linear dimensionality reduction.

In conclusion, the algorithm presented here is able to find groups of linearly or non-linearly linked genes. Any clustering algorithm can then be used to extract the type of relationship between the samples on those groups of genes. It has proved its effectiveness on both artificial and real-world data sets.

Acknowledgments

David Venet was in receipt of a JSPS postdoctoral fellowship. Earlier works on this topic were supported by UCB Pharma and the Region Wallone.

References

[1] Bar-Joseph, Z., Gifford, D. K., and Jaakkola, T. S., Fast optimal leaf ordering for hierarchical clustering, *Bioinformatics*, 17:S22–S29, 2001.

[2] Bryan, K., Cunningham, P., and Bolshakova, N., Application of simulated annealing to the biclustering of gene expression data, *IEEE Trans. Inf. Technol. Biomed.*, 10:519–525, 2006.

[3] Carmona-Saez, P., Pascual-Marqui, R. D., Tirado, F., Carazo, J. M., and Pascual-Montano, A., Biclustering of gene expression data by non-smooth non-negative matrix factorization, *BMC Bioinformatics*, 7:78, 2006.

[4] Cheng, Y. and Church, G. M., Biclustering of expression data, *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 8:93–103, 2000.

[5] Dash, M., Liu, H., and Yao, J., Dimensionality reduction for unsupervised data, *Proc. IEEE Int. Conf. Tools AI*, 9:532–539, 1997.

[6] Devney, M. and Ram, A., Efficient feature selection in conceptual clustering, *Proc. Int. Conf. Machine Learning*, 14:92–97, 1997.

[7] Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D., Cluster analysis and display of genome-wide expression patterns, *Proc. Natl. Acad. Sci. USA,* 95:14863–14868, 1998.

[8] Getz, G., Gal, H., Kela, I., Notterman, D. A., and Domany, E., Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data, *Bioinformatics*, 19:1079–1089, 2003.

[9] Getz, G., Levine, E., and Domany, E., Coupled two-way clustering analysis of gene microarray data, *Proc. Natl. Acad. Sci. USA*, 97:12079–12084, 2000.

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

[11] Halkidi, M., Batistakis, Y., and Vazirgiannis, M., On clustering validation techniques, *J. Intell. Inf. Syst.*, 17:107–145, 2001.

[12] Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Buhlmann, P., Gruissem, W., Hennig, L., Thiele, L., and Zitzler, E., A systematic comparison and evaluation of biclustering methods for gene expression data, *Bioinformatics*, 22:1122–1129, 2006.

[13] Reiss, D. J., Baliga, N. S., and Bonneau, R., Integrated bioclustering of heterogeneous genome-wide datasets for the inference of global regulatory network, *BMC Bioinformatics*, 7:280, 2006.

[14] Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B., Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization, *Mol. Biol. Cell*, 9:3273–3297, 1998.

[15] Tanay, A., Sharan, R., and Shamir, R., Discovering statistically significant biclusters in gene expression data, *Bioinformatics*, 18:S136–S144, 2002.

[16] Xing, E. P. and Karp, R. M., CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts, *Bioinformatics*, 17:S306–S315, 2001.

[17] `http://www.jsbi.org/journal/GIW06/GIW06F017Suppl.html`