Bagging for Biclustering: Application to Microarray Data

Blaise Hanczar and Mohamed Nadif

LIPADE, University Paris Descartes, 45 rue des saint-pres, 75006 Paris, France
blaise.hanczar@parisdescartes.fr

Abstract. One of the major tools of transcriptomics is the biclustering that simultaneously constructs a partition of both examples and genes. Several methods have been proposed for microarray data analysis that enables to identify groups of genes with similar expression pro?les only under a subset of examples. We propose to improve the quality of these biclustering methods by adapting the approach of bagging to biclustering problems. The principle consists in generating a set of biclusters and aggregating the results. Our method has been tested with success on artificial and real datasets.

1 Introduction

The capacity of microarray to measure simultaneously the expression of a whole genome under different experimental condition, is of great interest for biologists. Clustering techniques are one of the major tools to analyse these data. They allow the discovery of groups of genes that share a similar expression profile over all experimental conditions. We assume that genes that share similar expression profiles, have close biological functions. The clustering of gene expression over homogeneous conditions is therefore a relevant tool for functional analyses [2]. With the classic methods of clustering, like k-means, hierarchical clustering or self organizing maps, it is assume that genes in the same cluster have a similar behavior over all conditions. However, when the experimental conditions are heterogeneous, it may be more appropriate to form clusters of genes over only subset of conditions. In this case, biclustering methods are more adapted. For example, when a set of genes participates in a cellular process that is active only in a subset of conditions, or when a gene is implied in multiple pathways that may or not be co-active under a subset of conditions. Biclustering methods allow the identification of relevant groups of genes and conditions that cannot be identified by classic clustering techniques. These kinds of methods consist in simultaneous clustering on rows and columns, to reorganize the data set into homogeneous blocks. It is an old approach but over the last years it has attracted many authors, (see for instance; [12,13]).

In this paper, we try to improve the performance of biclustering algorithms by using the ensemble approach. The principle of ensemble methods is to construct a set of models, then to aggregate them into a single model, generally by using generally a voting scheme. It is well-known that these methods often perform better than a single model (see for instance; [9]). Ensemble methods first appeared in supervised learning problems. A combination of classifiers is more accurate than single classifiers [18]. A pioneer method, boosting, whose most popular algorithm adaboost, was developed

J.L. Balcázar et al. (Eds.): ECML PKDD 2010, Part I, LNAI 6321, pp. 490–505, 2010.

mainly by Shapire [22]. The principle is to assign a weight to each training example, then several classifiers are learned iteratively and between each learning step the weight of examples is adjusted depending on the classifier results. The final classifier is a weighted vote of classifiers constructed during the procedure. Boosting has been used with success on several microarray datasets [7,17]. The other type of popular ensemble methods, proposed by Breiman, is bagging [3]. The principle is to create a set a classifiers based on bootstrap samples of the original data. This approach, and especially the random forest, is also efficient for microarray based classification [8]. In the last years, several works have shown that ensemble methods can also be used in unsupervised learning. The principle of boosting is exploited by Frossyniotis et al. [11] in order to provide a consistent partitioning of the data. The boost-clustering approach creates, at each iteration, a new training set using weighted random sampling from original data, and a simple clustering algorithm is applied to provide new clusters. Dudoit and Fridlyand [10] used bagging to improve the accuracy of clustering in reducing the variability of PAM (Partitioning Around Medoids) results [15]. Their method has been applied to leukemia and melanoma datasets and permitted to differentiate the different subtypes of tissues. Strehl [23] have proposed an approach to combine multiple partitioning obtained from different sources into a single one. They introduced three heuristics to solve this problem: 1) a hypergraph partitioning algorithm that approximates the maximum mutual information objective function with a constrained minimum cut, 2) a cluster-based similarity partitioning algorithm that establishes a distance between elements based on the individual clustering, 3) a meta-clustering algorithm where groups of clusters (meta-clusters) are identified and consolidated.

Since ensemble methods allow the improvement of the performance of supervised classification and clustering, it is reasonable to think that they can also be used to tackle the biclustering problem. In this paper, we propose in this paper a bagging approach for the biclustering of microarray data. Although the problem of ensemble biclustering shares some traits with classic biclustering, there are two major issues which are specific to ensemble methods. The first one is the generation of a collection of biclusters. This raises the question of how to generate different biclusters and what is the the source of diversity ? We have chosen the bagging approach to generate biclusters from bootstrapped datasets. The second issue is about the aggregation function. How to combine the different biclusters and resolve the label correspondence problem? Based on the works of Sterhl and Ghosh [23], we propose a solution that consists in creating *meta-clusters* of biclusters. Finally, for each gene and example, we compute the probabilities of their belonging to each meta-cluster. We test our method both on artificial and real data. The results on artificial data show that ensemble methods allow to improve the accuracy of biclustering. The results obtained on real data show that ensemble biclusters give a lower residue than classic biclusters. Moreover the ensemble biclusters are also biologically more relevant with respect to the prior knowledge of data.

2 State of the Art

Consider the data matrix $X = \{E, G\}$ where $E = E_1, \ldots, E_N$ is a set of N examples represented by M-dimensional vectors and $G = G_1, \ldots, G_M$ is a set of M genes. A bicluster B is a submatrix of X defined by a subset of examples and a subset of genes;

$$B = \{(E^B, G^B); E^B \subseteq E, G^B \subseteq G\}.$$

A biclustering operator Φ_K is a function that delivers one or several biclusters $B_0 = \emptyset, B_1, \ldots, B_K$ given (E_i, G_j).

Note that a submatrix is considered as a bicluster if it presents a particular pattern. There is no definition of what these patterns are. The choice of considering a submatrix as a bicluster, is subjective and depends on the context. However there are some basic patterns that can be used to identify a bicluster. They are called constant, additive and multiplicative models. In constant models, all values in a bicluster are equal. In additive and multiplicative models, there is an additive and multiplicative factor between rows and columns respectively. Biclusters can also be identified by a mixture of these three models. We also consider different bicluster structures. Biclusters can overlap on the rows and/or columns, or present a tree or checkerboard structure. This diversity in the nature of biclusters accounts for the fact that no biclustering algorithm can identify all types of biclusters.

Several biclustering algorithms have been developed and applied to microarray analysis. Cheng and Church [6] were the first to propose an algorithm for this task. They consider that biclusters follow an additive model and use a greedy iterative search to minimizing the mean square residue. This algorithm identifies biclusters one by one. They applied their method to *yeast cell cycle data* and identified several biologically relevant biclusters. Lazzeroni and Owen [16] have proposed the popular plaid model. They assume that biclusters are organized in layers and follow a given statistical model incorporating additive two way ANOVA models. The search approach is iterative: Once $K-1$ layers (biclusters) have been identified, the K^{th} bicluster that minimizes a merit function depending on all layers is selected. They also applied their method to yeast data and found that genes in same biclusters share biological functions. Kluger et al. [14] used a spectral approach for biclustering assuming that the data matrix contains a checkerboard structure after normalization. This structure is identified by a singular value decomposition. They applied their method to *Lymphoma and Leukemia* datasets which contained different subtypes of cancer. On both datasets, conditions of the same subtype have been grouped together into the same biclusters. Tanay et al. [24] have developed *SAMBA*, an approach based on the graph theory coupled with statistical modeling of the data. SAMBA, applied to a lymphoma dataset, produces biclusters representing new concrete biological associations. Cheng et al. [5] have proposed the $pCluster$ method that has the advantage it can identify both additive and multiplicative biclusters in presence of overlap. They validated their method on yeast cell-cycle dataset using Gene Ontology annotations. Prelic et al. [21] made a comparative study of different biclustering methods for gene expression. They used a very simple divide and conquer the *Bimax* algorithm as a reference to investigate the usefulness of different biclustering algorithms. They concluded that Bimax produces results similar to those of more complex methods. Abdullah et al. [1] proposed a graph-drawing-based biclustering technique based on the crossing minimization paradigm. These algorithms are the most popular ones used in bioinformatics but this list is not exhaustive. Madeira and Oliveira [19] have published a good survey of biclustering methods for biological data analysis and enumerated more than 15 used in this context. Note also a more recent review of biclustering methods in data mining [4].

3 Bagged Biclustering

The principle of bagged biclustering consists in applying a biclustering method on multiple bootstrapped datasets and aggregate the results. Our method can be divided into 3 steps. The first one is the construction of a collection of biclusters. These biclusters are generated by the multiple application of a classic biclustering algorithm on bootstrapped data. Then, a distance matrix between the obtained biclusters is computed based on their similarity. This distance is used to make a hierarchical clustering of the biclusters. From the resulting dendrogram, K meta-clusters of biclusters, are extracted. In the last step, we compute for each element (examples and genes) the probabilities of its belonging to each meta-cluster. If the probability is higher than a given threshold, the element is assigned to the meta-cluster. At the end of the procedure, the K meta-clusters contain a set of examples and genes that represent our final biclusters. Note that the number of biclusters K is a parameter to be fixed before the computation. In the next section, we describe in detail the three steps of our process.

3.1 Bicluster Collection Generation

The aim of this part is to generate a high number of different biclusters. To do so, we generate bootstrap samples of the original data. A bootstrap sample is a random drawing with replacement of the same size as the original data. It contains on average 68% of the original elements. In the case of biclustering we should sample the data in both dimensions, genes and examples. If we do that, the bootstrap sample will contain 63% of the original examples and 63% of the original genes, which means that the bootstrapped data will contain only 46% of the original matrix. To keep the same proportion as in the classic bootstrap, we decide to perform the bootstrap sample on only one dimension. Since microarray data contain much many genes than examples, the bootstrap sample will be done on genes. From the original data $X = \{E, G\}$, R bootstrapped datasets are generated $\{X^b = \{E, G^b\}, b = 1, ..., R\}$ where G^b is a bootstrap sample of G assumed an iid sample.

On each of the R bootstrapped datasets, a biclustering algorithm,with the same parameters, is applied to produce K biclusters. We obtain a collection of KR biclusters noted B^b that are used to identify meta-clusters.

3.2 Metacluster Identification

The objective is to identify K meta-clusters merging the similar biclusters. The idea is that if two biclusters, generated from different bootstrapped data, are similar, it is likely that they represent the same bicluster. All bootstrapped biclusters representing the same bicluster should be grouped into a meta-cluster. The notion of similarity between two biclusters depends on the number of elements (genes and examples) they have in common. We use the Jaccard index to evaluate this similarity:

$$Sim(B_k, B_\ell) = \frac{|B_k \cap B_\ell|}{|B_k \cup B_\ell|} = \frac{|B_k \cap B_\ell|_E + |B_k \cap B_\ell|_G}{|B_k \cup B_\ell|_E + |B_k \cup B_\ell|_G}$$

where $|.|_C$ corresponds to the cardinality computed on the set C. From this similarity, we can define the dissimilarity between B_k and B_ℓ by $d(B_k, B_\ell) = 1 - Sim(B_k, B_\ell)$. This distance belongs to $[0, 1]$, 0 indicating that two biclusters are identical and 1 that they have no element in common. From the distance matrix, a hierarchical clustering of the bicluster is constructed using the average linkage. From the obtained dendrogram we can identify K meta-clusters in cutting the dendrogram. Each meta-cluster M_g; $g = 1, \ldots, K$, is a set of $\{B_1^b, ..., B_{K_g}^b\}$ where K_g is the cardinality of M_g. Note that we do not consider trivial meta-clusters containing a few biclusters. Before deducing these meta-clusters in the third step, we compute the probability of (E_i, G_j) belonging to the meta-clusters $M_1, \ldots, M_K$. It is denoted $p_g(E_i, G_j)$ and can be estimated by the proportion of biclusters of M_g containing (E_i, G_j). Note that the parameter K is the same than the one used in bicluster collection generation step.

3.3 Bicluster Computation

The last step consists in computing the final biclusters of the original data. Each meta-cluster is assigned to a bicluster. Then each element can be assigned to biclusters depending on computed probabilities $p_g(E_i, G_j)$. We have to distinguish two cases. In the first one we consider that there is no overlapping between the biclusters, i.e. (E_i, G_j) belongs to at most one bicluster. If there are no probabilities superior to a given threshold t then (E_i, G_j) is assigned to no bicluster. If there is at least one probability superior to t then (E_i, G_j) is assigned to the bicluster $\hat{B}$ belonging to the meta-cluster M_g and maximizing the probability $p_g(E_i, G_j)$. Then $\Phi_K(E_i, G_j)$ is equal to

$$\begin{cases} B_0 = \emptyset & \text{if } p_g(E_i, G_j) < t \forall g \\ \hat{B} = \arg\max_{M_g} p_g(E_i, G_j) & \text{otherwise} \end{cases}$$

In the second case, overlapping between two biclusters is possible, i.e. an element can belong to several biclusters. If there are no probabilities superior to the threshold t then the element is assigned to no bicluster. Otherwise, all biclusters whose corresponding probabilities are higher than the threshold t are assigned to the element.

$$\Phi_K(E_i, G_j) = \begin{cases} B_0 = \emptyset & \text{if } p_g(E_i, G_j) < t \forall g \\ B_k \in M_g & \text{such as } p_g(E_i, G_j) \geq t \end{cases}$$

Depending on the overlapping choice, we will use the first or the second case. Note that we can mix these two cases if we want different overlapping choices for examples and genes. For instance, a current choice in microarray analysis is to allow the overlapping on the examples and not on the genes because of the disproportion between the number of genes and examples. The function of the second case is used on examples and the function of the first case on genes. The threshold t has a high influence on the bicluster computation. The choice of its value will be discussed later.

4 Experiments on Artificial Data

In our simulation study, we evaluate the performance of bagged biclustering and compare it to single biclustering. We performed our experiments on artificial and real datasets

with five biclustering algorithms: Bimax [21], Cheng & Church [6], plaid model [16], spectral biclustering [14] and Xmotifs [20]. This part concerns the results on artificial data.

4.1 Generation of Artificial Datasets

The first part of our experiments is based on artificial data. These data cannot be considered as a reliable representation of real microarray data. Nevertheless they can be used to measure the performance and behavior of the new methods in conventional cases. The artificial data is a matrix with random values in which we included several biclusters. The dataset contains M genes, N examples and K biclusters. In our simulations $M = 200$, $N = 100$ and $K = 2, 4, 6$. The size of each bicluster is randomly chosen between 10 examples by 20 genes to 20 examples by 40 genes. The partition on the genes is defined by the classification $M \times K$ matrix $\mathbf{z}$ defined by $z_{ik} = 1$ if the gene i belongs to the bicluster B_k and 0 otherwise. In the same way, we consider the classification $N \times K$ matrix $\mathbf{w} = (w_{jk})$ representing the partition of the examples. The partitions of biclusters are defined randomly. Note that from $\mathbf{z}$ and $\mathbf{w}$, we can define different cases of bicluster overlapping. The simplest one corresponds to no overlapping structure. The second case consists in considering the presence of the overlapping of genes. In the same way, we define the overlapping of examples. The last and most complex case is total overlapping where genes and examples can belong to several biclusters. An overlapping between biclusters can occur only in this last case. Note also that a gene or example can belong to no bicluster. The total overlapping structure is the most general case, so we consider only this situation in our experiments.

There is no general definition of what a bicluster is. The different algorithms used search different models of biclusters. In our experiments we use five different biclustering algorithms. These algorithms do not identify biclusters of the same nature. For each of them, we define a bicluster with the same model as the one used in the original paper where the algorithm was published.

- **Plaid model.** We consider the more general model. The values of X_{ij} belonging to a bicluster B_k (layer) depend on 4 parameters: a constant μ_0 describing the background layer, μ_k the average of B_k, α_{ik} and β_{jk} allowing to identify respectively a subset of genes and examples having identical responses. The values of X_{ij} are then represented as:$X_{ij} = \mu_0 + \sum_{k=1}^{K} z_{ik} w_{jk} (\mu_k + \alpha_{ik} + \beta_{jk})$. In our experiments, all values belonging to biclusters are generated according to a uniform distribution $U[0, 5]$. The values outside a bicluster, are generated according a uniform distribution $U[-10, 10]$.
- **CC model.** For the Cheng and Church model, The values of X_{ij} belonging to a bicluster B_k depend on 3 parameters: μ_k the average of B_k, μ_{ik} and μ_{jk} are respectively the means of E_i and G_j belonging to the bicluster B_k. The values of X_{ij} are then represented as:$X_{ij} = \sum_{k=1}^{K} z_{ik} w_{jk} (\mu_{ik} + \mu_{jk} - \mu_k)$. The values are generated in the same manner that for the Plaid model.
- **Xmotifs.** we consider binary data and biclusters that depend on two parameters $\alpha > 0$ and $\beta < 1$ defined by the three following rules. A bicluster must contain a number of examples superior to αN. All values in a bicluster are equal to (0 or

1). If a gene does not belong to a bicluster, there is a maximum βN values that are equal.

- **Spectral biclustering.** It is a special case since the data follow a checkboard structure. Each value belongs to a bicluster. The examples are partitioned into two clusters and genes into $K/2$ clusters. Each bicluster is defined by its means value μ_k. In our simulations, all values of a bicluster are chosen equal to their mean value.
- **Bimax**. The model used for Bimax is the simplest. The dataset is considered binary, all values in a bicluster are equal to 1 and all values out of any bicluster are 0. Then, biclusters are sub-matrices containing only 1s. Each X_{ij} value is then defined by $X_{ij} = z_{ik} \times w_{jk}$.

Once the biclusters are defined and included in the dataset, we add a Gaussian noise $N(0, \sigma^2)$. The variance permits to control the difficulty of the biclustering task.

4.2 Study Design

We evaluate the performance of each biclustering method by comparing the biclusters obtained by different algorithms and true biclusters. We compute the error of biclustering by estimating the total number of misclassified values. Since this estimation should not depend on the labellings of the biclusters, we enumerate all possible relabellings and retain the label that gives the smallest error noted e. The study design used in our experiment is the following:

1. Generate an artificial dataset X with K biclusters
2. Apply a biclustering algorithm on X to identify K biclusters
3. Compare the true biclusters to the biclusters obtained by the five algorithms and compute the biclustering error noted e^{single}.
4. Apply the associated bagged biclustering of the five algorithms on X to identify K biclusters
5. Compare the true biclusters to the obtained biclusters and compute the biclustering error noted e^{bagged}.
6. Iterate steps 1-5 200 times and compute the means of e^{single} and e^{bagged}.

4.3 Results on Artificial Data

The choice of the decision threshold t is crucial. A too small value of t may imply that an element, belonging to a bicluster, will be assigned to no bicluster. The obtained biclusters will be small and several genes will miss to be assigned. But we will be highly confident that all identified genes and examples are really assigned to biclusters. On the other hand, a too high value of t leads to assign an element to biclusters not containing this element. The obtained biclusters will be large and tend to contain all elements actually belonging to the biclusters. The risk is that they will also contain false positives, i.e. elements wrongly assigned. The dependence of t on these different situations implies its difficult choice. Here, in our experiments the assessing of t is performed empirically, giving a good tradeoff between false and true positives.

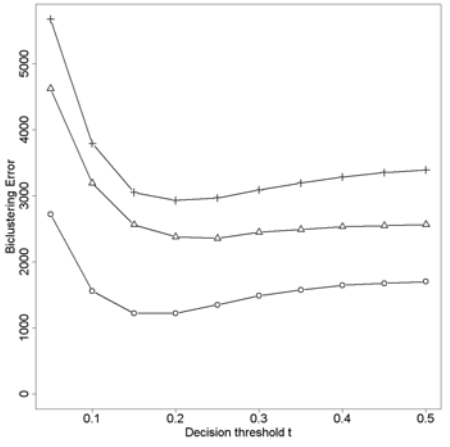

Fig. 1. $\circ : K = 2, \triangle : K = 4, + : K = 6$. Biclustering error of the Bimax algorithm in function of the value of the decision threshold t. The dot line represents results for problems with 2 biclusters, the triangle line with 4 biclusters and the cross line with 6 biclusters.

Figure 1 shows the biclustering error in function of the decision threshold t with the Bimax algorithm. The three lines, dot, triangle and cross, represent respectively the results on dataset containing 2, 4 and 6 biclusters. The behavior of the three curves is the same: they are strongly decreasing from $t = 0$ to $t \approx 0.2$ then the biclustering error increases slowly. For all curves, the minimum error is around $t = 0.2$. We use this threshold in our experiments. We employ the same procedure to assess the threshold for the other algorithms.

We performed different experiments giving the same results for different values of $K = 2, 4, 6$. Figure 2 and 3 show the biclustering error with the five algorithms: Bimax, Cheng & Church, plaid model, spectral biclustering and Xmotifs. For each algorithm, three graphics represent the results on datasets containing 2,4 and 6 biclusters. In each graphic, the error biclustering is computed in function of the quantity of noise introduced in the dataset. The dot curve represents the error of the single biclustering algorithm, the triangle curve the bagged biclustering. We see that in all cases the error is naturally increasing with the noise. For Bimax, we see that at $\sigma = 0.4$ the error of single biclustering "jumps" from 0 to 1700. The error of the bagged biclustering increases slowly to join the error of the single biclustering at $\sigma = 1.8$. We can interpret this as follows: for $\sigma \leq 0.4$ the biclustering problem is easy, the performance of biclustering is maximal. For $\sigma \geq 1.8$ the datasets are so noisy that all biclusterings are meaningless, we have checked that the error is the same as a random biclustering. These graphics show that bagged Bimax gives a better biclustering than single Bimax. We find the same type of results in the major part of our experiments. The Cheng & Church algorithm and Bimax have the same behavior. Error of single CC increases faster than bagged CC. In plaid model results, we see that the error of the bagged plaid model is lower than that of the single plaid model for $\sigma < 1$. For higher level of noise, $\sigma \geq 1$, the two algorithms give the same performance as "random" biclustering. The results of

Spectral biclustering are singular since the error does not jump for a given value of σ but is regularly increasing with σ. The error curves of single and bagged spectral biclustering are almost parallel. The error of bagged spectral biclustering is always below the error of the single spectral biclustering. For Xmotifs results, the error curve of bagged Xmotifs always begins below the curve of single Xmotifs, then joins it for higher levels of noise. The results, including various biclustering algorithms and different number of biclusters (not reported here), show that bagged biclustering gives better results than single biclustering.

In these experiments on artificial data, we assume the knowledge of the true number of biclusters, i.e. the number of meta-clusters, but in real data problems this number is generally unknown. Here we show that a measure of clustering goodness, like the *within-group sum of squares* criterion noted commonly W, also named *within-group inertia*, can be used on the bagged bicluster dendogram to find an appropriated number of meta-clusters. In our simulations we perform the agglomerative hierarchical clustering on the biclusters and we try several cuts of the tree in order to obtain different numbers of meta-clusters. For each cut, we compute W of the obtained partition of bagged biclusters. The W criterion, depending on a partition, decreases with the number of meta-clusters, and so a scree plot with one or several elbows may used to propose a cut of the dendrogram. For instance, the scree plots of Figure 4 express W computed in function of the number of meta-clusters with the CC algorithm. From each scree plot, we can retain the first elbow corresponding respectively to 2, 4 and 6 meta-clusters (represented with a dotted line in the graphics). Note that we have also tested the influence of the number of bootstrap iterations R on our method. Our simulations show that, when R is large ($R > 100$), this parameter has no effect on the results of bagged biclustering.

5 Experiments on Real Data

We tested our approach on real microarray datasets. The evaluation is more subjective than on artificial data since we cannot know the true biclusters. We rely on statistical measures and the biological context of the datasets to evaluate the goodness of the biclusters. In the following, we have applied the plaid and CC models and evaluate the quality of a bicluster B_k by computing the Mean Square Residue (MSR) defined by

$$\frac{1}{\#B_k} \sum_{i,j} z_{ik} w_{jk} (X_{ij} - \mu_{ik} - \mu_{jk} + \mu_k)^2.$$

The symbol # denotes the cardinality. We have computed the MSR for the biclusters in single and bagged contexts. The partition of genes, extracted from biclusters, should be coherent with known genetic information contained in public databases like KEGG. We expect to find genes belonging to the same biological pathways in a bicluster. We have applied single biclustering algorithms and bagged biclustering to four microarray datasets. These datasets are available online[1] and their characteristics can be found in table 1.

Firstly, the Bimax, Xmotifs (after binarization process) and spectral algorithms, are not commented in the following. Using both single and bagged versions they did not

[1] http://algorithmics.molgen.mpg.de/Static/Supplements/CompCancer/datasets.htm

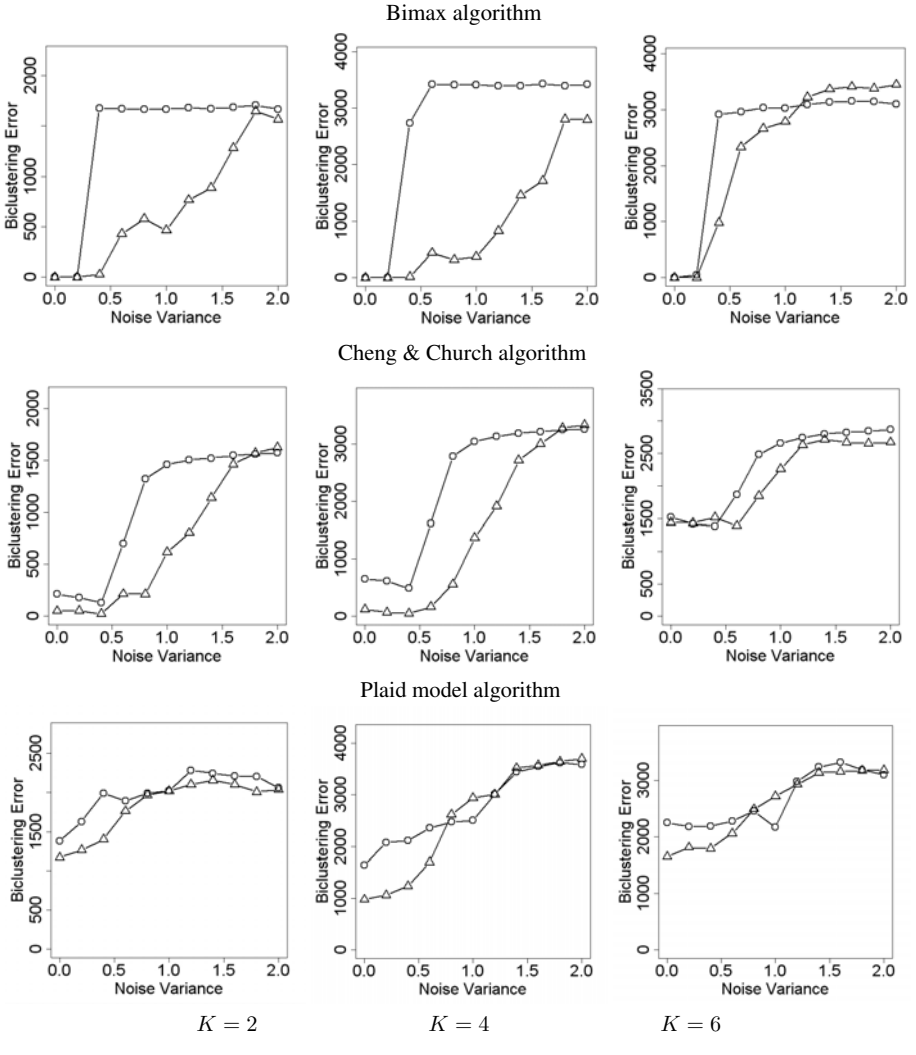

Fig. 2. Biclustering error in function of noise variance on artificial data. The three graphic lines correspond respectively to results with Bimax, CC algorithm and plaid model. The columns correspond respectively to results with 2, 4 and 6 biclusters. Dot lines and triangle lines correspond respectively to single and bagged biclustering.

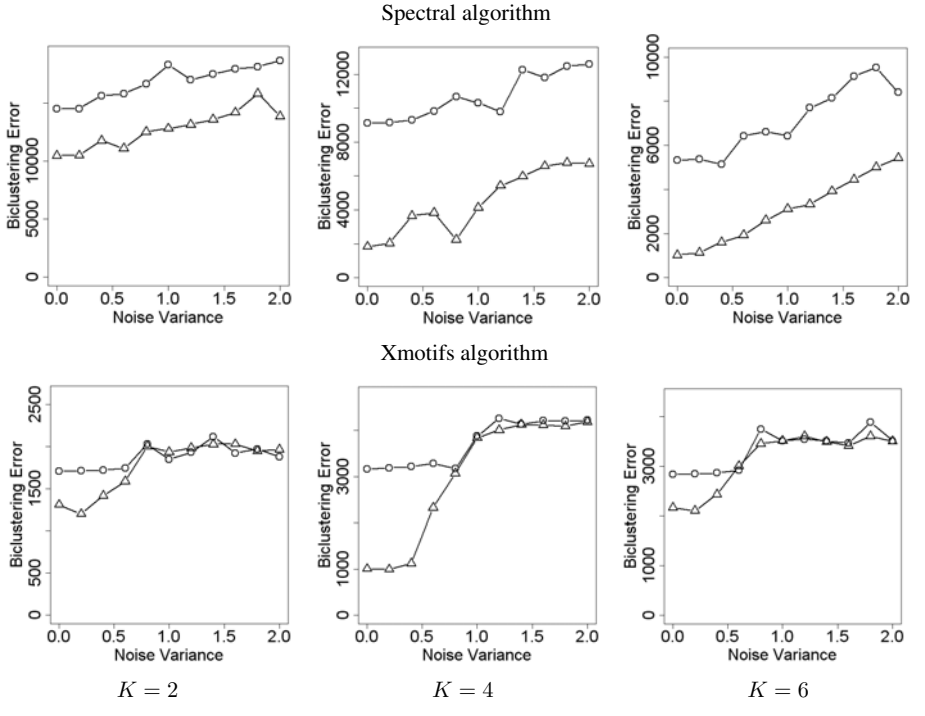

Fig. 3. Biclustering error in function of noise variance on artificial data. The two graphic lines correspond respectively to results with Spectral biclustering and Xmotifs. The columns correspond respectively to results with 2, 4 and 6 biclusters. Dot lines and triangle lines correspond respectively to single and bagged biclustering.

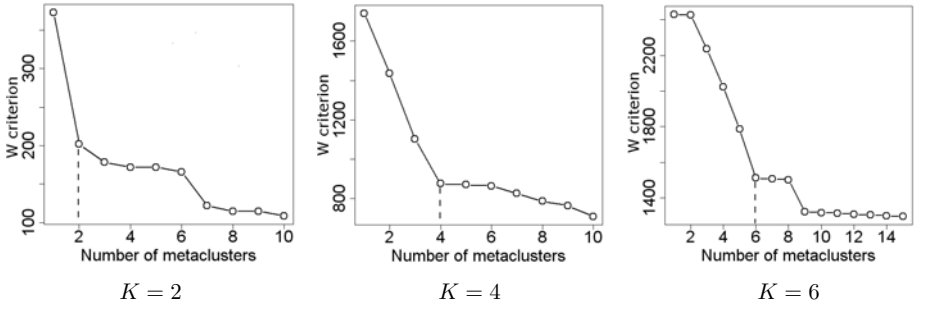

Fig. 4. The within-group sum of squares in function of the number of meta-clusters with the CC algorithm

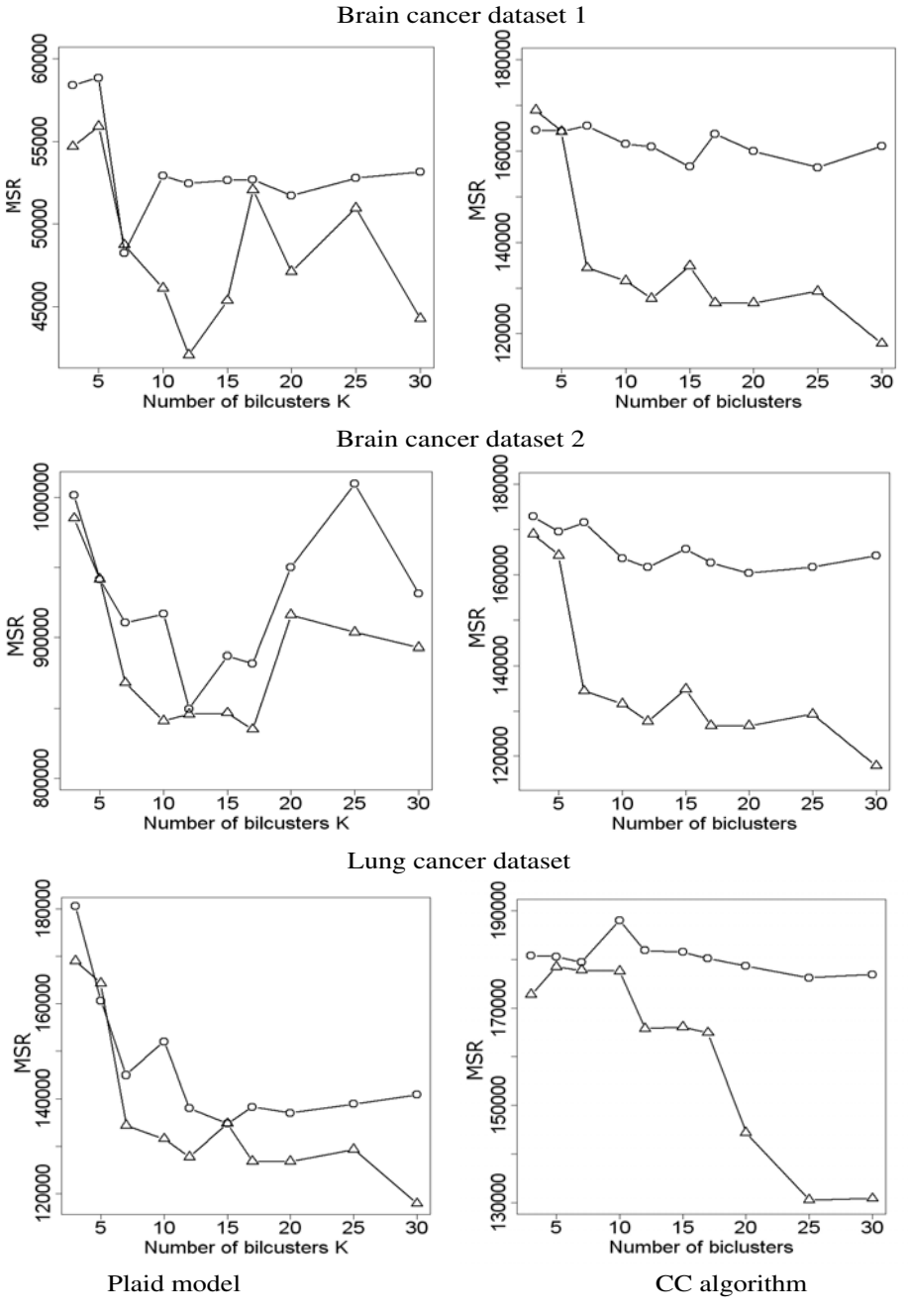

Fig. 5. MSR in function of the number of biclusters K. Dot line correspond to MSR of single biclustering and triangle line of bagged biclustering.

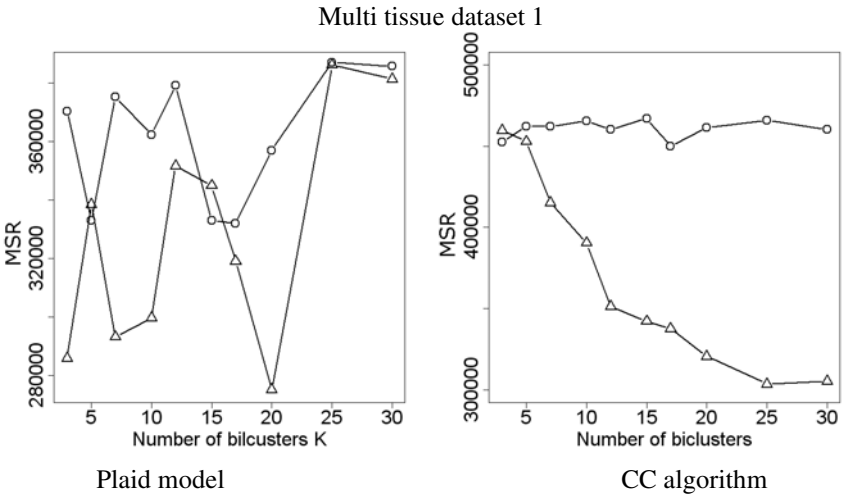

Fig. 6. MSR in function of the number of biclusters K. Dot line correspond to MSR of single biclustering and triangle line of bagged biclustering.

find any biclusters from the 4 datasets. The type of biclusters sought by these methods, are not suitable for our datasets.

Applying single and bagged biclustering with the plaid model and the CC models on the four microarray datasets, we compute the MSR of results produced by each method according different values of K. The number of bootstrap iterations for bagged biclustering is $R = 100$. The figures 5 and 6 show the MSR of the single (dot lines) and bagged (triangle lines) biclustering in function of the number of biclusters K. First column represents results with plaid model and second column with CC algorithm. The results with the plaid model exhibit a lot of variability, but the MSR of bagged biclustering is lower than single biclustering. In the four datasets there is a peak phenomenon, MSR is firstly decreasing then increasing with the number of biclusters. The minimum is around $K = 12$. For lung cancer dataset, MSR is strongly decreasing until $K = 10$ then is stable. In multi tissue dataset, the results are very variable, it is hard to see the general behavior of MSR in function of K. The important point is that MSR of bagged biclustering is below MSR of single biclustering. The only cases where the MSR of the two approaches are equal is where MSR are at its maximum. That means for an optimal number of biclusters, bagged biclustering produces better results than single biclustering. The results with CC algorithm are clearer. All five graphics present the same behavior and we note that the performance increases with the number of biclusters. The MSR of single biclustering remains stable whereas the MSR of bagged biclustering decreases strongly and becomes stable for high number of biclusters. In all datasets, bagged biclustering is much better than single biclustering for $K > 7$.

We also compared single and bagged biclustering based on the coherence of the obtained gene partition. If two genes are in the same bicluster, we can assume that these two genes are biologically related. A good tool to check if there is an identified relation between two genes, is the pathway database of the Kyoto Encyclopedia of Genes

Table 1. Characteristics of the datasets and number of over-represented pathways

			Plaid model		CC algorithm	
Datasets	#ex.	#genes	Single	Bagged	Single	Bagged
Brain cancer 1	50	1377	8	18	19	25
Brain cancer 2	42	1379	10	26	3	14
Lung cancer	203	1543	12	20	15	19
Multi tissue	190	1363	4	15	26	41

and Genomes (KEGG). These pathways represent molecular interaction and reaction networks for metabolism, various cellular processes, and human diseases. All genes in the same pathway are considered biologically related. The number of over-represented pathways is computed for each bicluster. We use the hypergeometric test to check if a path is over-represented in a bicluster. Given pathway a PW_i, let p_i be the probability that a gene belongs to the pathway PW_i and $S_i = [p_i M]$ ([.] denotes the integer part) the number of genes belonging to PW_i. The probability of obtaining k genes belonging to the pathway PW_i from a random selection of A genes follows a hypergeometric law:

$$P(k, S_i, M, A) = \frac{C_{S_i}^{k} C_{M-S_i}^{A-k}}{C_M^A}.$$

Given a bicluster $B = \{E, G\}$, let X_i be the number of genes belonging to the pathway PW_i, the probability of obtaining at least X_i genes belonging to PW_i by a random selection is defined by:

$$Pr(k \geq X_i) = \sum_{x \geq X_i} P(x, S_i, M, |G|).$$

We compute this probability for all pathways and the Holm correction is applied to address the problem of multiple comparisons. Finally all pathways with a corrected p-value inferior to 0.05 is considered over-represented. We assume that the number of over-represented pathways represents the biological information captured by the genes of the biclusters. A good biclustering should provide high number of over-represented pathways. We use this measure to compare the results of the different algorithms. The chosen number of biclusters is the one that minimizes the MSR. In the table 1, we report the number of over-expressed pathways in datasets and it appears clearly that bagged biclusters contains much more over-expressed pathways and can be consider more biologically relevant than single biclusters.

6 Conclusion

In this paper we have introduced the concept of ensemble methods for biclustering in the context of microarray data. The proposed bagged biclustering generates a collection of biclusters based on bootstrap samples of the original data. Then a distance matrix between biclusters is computed based on the Jaccard index. From these distances we

construct a dendrogram and identify meta-clusters. The probability to belong to each meta-cluster is computed for each element (E_i, G_j). Finally the final biclusters are built based on these probabilities. The performance of our method has been tested on both artificial and real microarray data. On artificial data, we have shown that ensemble method enables to strongly decrease the biclustering error compared to classic methods whatever the used biclustering algorithm, the number of biclusters and the chosen noise level. On real data, bagged biclustering provides biclusters more relevant than single biclustering according to their MSR value. In addition, we have noted that bagged biclusters capture more biologic information since they contain more overexpressed pathways. The use of our approach is a new powerful tool for microarray analysis and should allow biologists to identify new relevant patterns in gene expression data.

Aknowledgments

This research was supported by the CLasSel ANR project.

References

1. Abdullah, A., Hussain, A.: A new biclustering technique based on crossing minimization. Neurocomputing 69(16-18), 1882–1896 (2006)
2. Alizadeh, A.: Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 403, 503–511 (2000)
3. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
4. Busygin, S., Prokopyev, O., Pardalos, P.: Biclustering in data mining. Computers and Operations Research 35(9), 2964–2987 (2008)
5. Cheng, K.O., Law, N.F., Siu, W.C., Liew, A.W.: Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization. BMC Bioinformatics 9, 210 (2008)
6. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. Int. Conf. Intell. Syst. Mol. Biol., vol. 8, pp. 93–103 (2000)
7. Dettling, M., Bühlmann, P.: Boosting for tumor classification with gene expression data. Bioinformatics 19(9), 1061–1069 (2003)
8. Diaz-Uriarte, R., Alvarez de Andres, S.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7(3) (2006)
9. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
10. Dudoit, S., Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. Bioinformatics 19(9), 1090–1099 (2003)
11. Frossyniotis, D., Likas, A., Stafylopatis, A.: A clustering method based on boosting. Pattern Recognition Letters 25, 641–654 (2004)
12. Govaert, G., Nadif, M.: Clustering with block mixture models. Pattern Recognition 36, 463–473 (2003)
13. Govaert, G., Nadif, M.: Block clustering with Bernoulli mixture models: Comparison of different approaches. Computational Statistics and Data Analysis 52, 3233–3245 (2008)
14. Kluger, Y., Basri, R., Chang, J.T., Gerstein, M.: Spectral biclustering of microarray data: coclustering genes and conditions. Genome Res. 13(4), 703–716 (2003)
15. van der Laan, M., Pollard, K., Bryan, J.: A new partitioning around medoids algorithm. Journal of Statistical Computation and Simulation 73(8), 575–584 (2003)

16. Lazzeroni, L., Owen, A.: Plaid models for gene expression data. Tech. rep., Stanford University (2000)
17. Long, P., Long, P.M., Vega, V.B.: Boosting and microarray data. Machine Learning 1-2(52), 31–44 (2003)
18. Maclin, R.: An empirical evaluation of bagging and boosting. In: Proceedings of the Fourteenth National Conference on Artificial Intelligence, pp. 546–551. AAAI Press, Menlo Park (1997)
19. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1(1), 24–45 (2004)
20. Murali, T., Kasif, S.: Extracting conserved gene expression motifs from gene expression data. Pacific Symposium on Biocomputing 8, 77–88 (2003)
21. Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Buhlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9), 1122–1129 (2006)
22. Schapire, R.: The boosting approach to machine learning: An overview. In: Nonlinear Estimation and Classification. Springer, Heidelberg (2003)
23. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)
24. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics 18(Suppl. 1), 136–144 (2002)