Journal of Bioinformatics and Computational Biology

A PARALLEL SCHEME FOR COMPARING TRANSCRIPTION FACTOR BINDING SITES MATRICES

SOLENNE CARAT
REMI HOULGATTE
Institut du thorax, INSERM U 915, Université de Nantes, France
Solenne.Carat@univ-nantes.fr
Remi.Houlgatte@univ-nantes.fr

JEREMIE BOURDON
LINA CNRS UMR 6241, Université de Nantes, France
INRIA Rennes-Bretagne-Atlantique, France
Jeremie.Bourdon@univ-nantes.fr

Gene regulation implies many mechanisms. Their identification is a crucial task to construct regulatory networks, necessary to understand the pathology in many cases. This requires the identification of transcription factors that play a role in regulation. Numerous motif discovery tools are now available. Combining efficiently their results appears useful for comparing and clustering these motifs in order to reduce redundancies and to identify corresponding transcription factor. We develop a method that produces, compares and clusters a set of motifs and identifies some close motifs in databases like JASPAR and the public version of Transfac. Unlike previous comparison methods, where each matrix column is compared independently, we have developed a global method to compare motif that helps to reduce the number of false positives. We also propose an original graph motif model that generalizes the classical position specific pattern matrices. Finally, we present an application of our method to study ChIP-chip data sets in the context of an Eukaryotic organism.

Keywords: Transcription Factor binding sites identification; PWM Clustering; ChIP-chip analysis; Regulatory Networks

1. Introduction

One of the most challenging problem in genomics is the understanding of mechanisms involved in gene regulation. Identifying all transcription factors having a role in a living system is thus crucial to construct gene regulatory networks. Transcription factors usually recognize specific transcription factor binding site (TFBS) on DNA. These TFBS are commonly represented by position specifc matrices, like Position Weight Matrices (PWM) [1] or Position Frequency Matrices (PFM) [2] that allows recognizing a slightly conserved part of the DNA sequence. Given a set of DNA sequences where a transcription factor has bind, one first has to extract conserved parts in the sequences. Such parts are putative binding sites. Many tools based on

several methods, like Expectation-Maximisation, Gibbs sampling and word counting, have developed over the last decade to identify these binding sites. Expectation maximisation (EM) is a local optimization procedure to maximize a likelihood function with hidden variables, but it is sensitive to its initialization point. One tool of the most used based on EM algorithm is MEME [3]. Gibbs sampling is a general technique to perform probabilistic inference. Unlike EM, gibbs sampling algorithm is based on global search upon a parameterized distribution. However, reestimation of parameters based on randomly generated samples is time consuming because of large number of iterations. AlignACE [4] is an example of Gibbs sampling algorithm. Finally, several word counting methods integrate supplementary informations to sequences such as phylogenetical conservation, ChIP-chip results, to detect more significant patterns. For instance, MDscan [5] or MotifRegressor [6] use ChIP-chip result to refine analysis. All these tools produce significant but different results and the use of a combination of several tools provides an exhaustive analysis of TFBS [7]. The following step is the comparison of the different PWM. The most used approach to quantify PWM similarities are based on a column by column comparison with various distance measures [8,9]. We have developed a method that compares and clusters a set of motif to build a subset of pertinent and non redundant motifs. Our method uses an improved PWM comparison approach based on a global similarity search with filtering of non-informative positions. Such a global distance ensures that several bias (see Section 2.1.4) induced by a column-by-column comparison are removed. Then, a threshold clustering of PWM is performed. A PWM consensus is build from all the threshold clusters. All these PWM consensus are further compared to motif databases (e.g. JASPAR [10] and the public release of Transfac [11]), to characterize their similarities with known motifs, and searched singularities such as palindromes, tandem repeats, simple repeats... Our complete method has been implemented with some computational tricks (parallelization, dynamic programming) that allow to deal with some large sets of patterns. Version 1 of our tool, motifsComparator, is limited to PWM clustering and further versions will include the whole process.

2. Methods

In this section, we describe precisely our method. The first part corresponds to the definition of an appropriate distance between two pattern matrices. Most of them can be adapted to compare both Position Weight Matrices and Position Frequency Matrices. Next, we discuss some clustering methods that can be used in several types of use (visualization, graph motifs, consensus,...). Finally, some useful implementation tricks are described.

2.1. *Pattern matrices comparison*

The capacity to compare pattern matrices corresponding to transcription factor binding sites is essential to avoid redundancies and to identify corresponding tran-

scription factor from known matrices available in public databases. Several methods have already been developed to compare PWM. Most of them consider PWM like a product multinomial distribution in which each column is a set of independent observations. PWM comparison is reduced to a column by column comparison [12]. Here, we discuss five main methods based on this principle. Then, we define a new distance that allows to compare pattern matrices in a more global way.

In the remainder of the section, we will use the following notations:

- $\mathcal{A}$ denotes an alphabet (typically, $\mathcal{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$) for DNA sequences, endowed with a probabilistic background model (i.e., the prior probability of letter $\mathtt{A}$ is $p_{\mathtt{A}}$);
- $P = (P_{i,\sigma})_{i\in\{1,\ldots,n\},\sigma\in\mathcal{A}}$ and $Q = (Q_{i,\sigma})_{i\in\{1,\ldots,n\},\sigma\in\mathcal{A}}$ denote two n-length pattern scoring matrices;
- $\overline{P}_i = \sum_{\sigma\in\mathcal{A}} p_\sigma P_{i,\sigma}$ is the expectation of the i-th column of P.
- $\widehat{P}_i = \sum_{\sigma\in\mathcal{A}} P_{i,\sigma}$ is the sum of all terms in the i-th column of P.
- $p(x, d) = \text{Prob}\{X_d > x\}$, where X_d is a d-order χ^2 random variable is the p-value of score x in a χ^2 statistics.

We are interested in defining some scoring measure between two matrices P and Q.

Notice that in the case of Position Frequency matrices and a uniform probabilistic background model (that is assumed in several studies), $\overline{P}_i = 1$.

2.1.1. *Kullback-Leibler score*

This distance, defined by Kullback in [13], is often used for matrice comparisons [14,15]. The similarity between two motifs is defined by

$$S_{KL}(P,Q) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{\sigma\in\mathcal{A}} p_\sigma\left[\frac{P_{i,\sigma}}{\overline{P_i}}\log\left(\frac{P_{i,\sigma}\overline{Q_i}}{Q_{i,\sigma}\overline{P_i}}\right) + \frac{Q_{i,\sigma}}{\overline{Q_i}}\log\left(\frac{Q_{i,\sigma}\overline{P_i}}{P_{i,\sigma}\overline{Q_i}}\right)\right].$$

2.1.2. *Pearson correlation coefficient score*

The Pearson correlation coefficient was introduced in motif comparison by Pietrokovski [16]. The formula to compute similarity between two PWM is as follows:

$$S_{PCC}(P,Q) = \sum_{i=1}^{n}\frac{\sum_{\sigma\in\mathcal{A}}(P_{i,\sigma}-\overline{P}_i)(Q_{i,\sigma}-\overline{Q}_i)}{\sqrt{\sum_{\sigma\in\mathcal{A}}(P_{i,\sigma}-\overline{P}_i)^2\ \sum_{\sigma\in\mathcal{A}}(Q_{i,\sigma}-\overline{Q}_i)^2}}$$

Notice that the higher the score is, the most similar the matrices are. One of the major disadvantage of this scoring diagram, is that dissimilar columns do not have the same penalization weight in the final score.

2.1.3. *Average log-likelihood ratio*

This metric, introduced by Wang and Stormo [17], is a weight sum of two log-likelihood ratio. The measure takes the background into account.

$$S_{ALLR}(P,Q) = \sum_{i=1}^{n} \frac{\sum_{\sigma\in\mathcal{A}} P_{i,\sigma} \log\left(\frac{P_{i,\sigma}}{p_\sigma \widehat{P_i}}\right) + Q_{i,\sigma}\log\left(\frac{Q_{i,\sigma}}{p_\sigma \widehat{Q_i}}\right)}{\widehat{P_i}+\widehat{Q_i}}.$$

2.1.4. *Pearson χ^2 column-by-column score*

In [9], Schones et al. noticed that Pearson χ^2 test can be used in the context of motif comparison. This score consists in comparing the distributions of two aligned columns of the matrices. These columns follow a multinomial distribution an can be compared by using a χ^2 homogeneity test. We first define the column-by-column score between $P_i = (P_{i,\sigma})_{\sigma\in\mathcal{A}}$, the i-th column of P and $Q_i = (Q_{i,\sigma})_{\sigma\in\mathcal{A}}$, the i-th column of Q.

$$C(P_i,Q_i) = (\widehat{P_i}+\widehat{Q_i})\left[\sum_{\sigma\in\mathcal{A}}\left(\frac{P_{i,\sigma}^2}{\widehat{P_i}}+\frac{Q_{i,\sigma}^2}{\widehat{Q_i}}\right)\frac{1}{P_{i,\sigma}+Q_{i,\sigma}}-1\right].$$

Under the null hypothesis that the two column follows the same multinomial law, $C(P_i,Q_i)$ is comparable to a χ^2 distribution of order 3. Then, if one assumes that all the columns are independent, one can define a distance by taking the geometric mean of all p-values.

$$S_{\chi^2}(P,Q) = \left(\prod_{i=1}^{n} p(C(P_i,Q_i),3)\right)^{1/n}.$$

Notice that when the marginal frequencies are small, Fisher-Irwin test, that involves multiple hypergeometric distribution, is more suited.

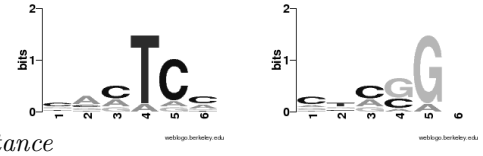
Notice also that such a method implies several bias.

Bias 1. The same absolute error impacts very differently for small frequencies than for high frequencies. The following table illustrates this fact by providing the p-value obtained when comparing two pairs of columns that look similar pair-wise.

Pairs	A	C	G	T	p-value
1	1%	33%	33%	33%	0.395
	4%	32%	32%	32%	
2	22%	26%	26%	26%	0.03
	25%	25%	25%	25%	

Bias 2. Combining the p-values of different individual column tests may imply some strange artifacts. Indeed, the p-value of a comparison between two equal columns is 0 (notice that in a normal computer precision, this is also true for very close but not necessary equal columns). As a consequence, as soon as an alignement between two motifs contains two equal columns, the score of a complete alignment is also zero, whatever the other columns are. Figure 1 illustrates these two arguments.

Fig. 1. Illustration of bias 2 : column 3 is the same in both sequences. When comparing these two patterns, one obtains a p-value of 0



2.1.5. *Euclidean distance*

The Euclidean distance score between two motifs is given by

$$S_{EUCL}(P,Q) = \sum_{i=1}^{n} \sum_{\sigma \in \mathcal{A}} (P_{i,\sigma} - Q_{i,\sigma})^2.$$

Comparing the Euclidean distances between two columns can appear to be simple and inefficient by a lack of normalization and the difficulty of providing a real significance of this score. Nevertheless, in [8], it is proved that an approximate significance can be computed by using permutations of the existing data. With this new information, the Euclidean distance score performs well in some real cases.

2.1.6. *Global comparison*

Here, we aim at defining a new score formula that decrease the scale effects of small frequencies (bias 1) and allows to compare scores with different length. First, we noticed that small values for $P_{i,\sigma}$ can drastically affect the final score, notably because all the comparisons were done between ratios of values. We decide to suppress these non representative values by taking solely the values over a given threshold ε (typically frequencies over 5%). By reducing the number of values by column, column by column comparison scores are no more appropriate. We thus design a χ^2 test suited to this case. First, let us consider the following contingency table defined by

$$T = \{(P_{i,\sigma}, Q_{i,\sigma}), i \in \{1, \ldots, n\}, \sigma \in \mathcal{A} \text{ and } P_{i,\sigma} \geq \varepsilon \text{ or } Q_{i,\sigma} \geq \varepsilon\}.$$

Finally, one has to compare two distributions that may be assumed to be of multinomial type when the threshold is small.

First, one computes a χ^2 statistics related to T,

$$K(T) = \frac{\widehat{E(T)}}{\widehat{O(T)}^2} \sum_{(E,O)\in T} \frac{O^2}{E} - 1, \tag{1}$$

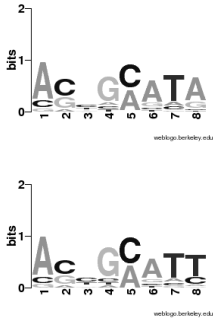
where $\widehat{E(T)} = \sum_{(E,O)\in T} E$ and $\widehat{O(T)} = \sum_{(E,O)\in T} O$.

$$S_{GLOBAL}(P,Q) = p(K(T), |T| - 1),$$

where $|T|$ is the cardinality of set T. Notice at this point that this score is not symmetric. One matrix plays the role of E(xpected values) while the other plays the role of O(bserved values). This score can be transformed into a symmetric one by summing the statistics $K(T)$ for both choices or by taking the minimal p-value between the two possibilities.

Figure 2 presents a comparison of different scoring methods. First notice that, except for Euclidean score, all scores provides coherent results. Nevertheless, there is a lack of distribution knowledge for column-by-column tests. Clearly, since they are sums or products of non normalized variables, their variance, and thus their distribution, depend on the number of columns of the motif alignment. Their interpretation for comparing large sets of heterogenous motif is difficult making the use of column-by-column tests inappropriate in this context. Furthermore, Euclidean distances always computes a better distance for shorter sub-motifs. It is thus difficult to compare motifs with different sizes. This can be partly solved by considering a p-value computation based on permutations of columns. This spends a lot of computation time. Finally, in this example, the longest coherent submotifs are those that possess the best global score (size 7).

Fig. 2. A comparison of different distance scores for submotifs composed by the 6-th, 7-th and 8-th first positions of two motifs. The motifs coincides on the 7-th first positions.



Method	sub-motif size		
	6	7	8
Kullback-Leibler	0.027	0.023	0.236
PCC	6	7	6.4
ALLR	2.5	3	1.7
Euclidean	61	66	26547
χ^2	0.067	0.062	0.088
Global	3.10^{-31}	3.10^{-38}	2.10^{-12}

2.2. *Pattern clustering*

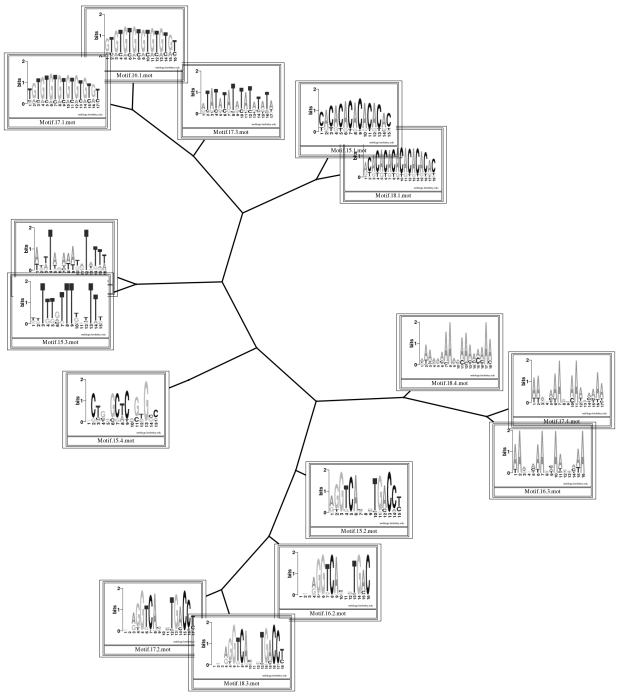
Many clustering methods have been developed. General principle of clustering is to maximize distance inter-cluster and minimize distance intra-cluster. In the context of pattern clustering, most used methods are unsupervised. Two principal methods can be described: hierarchical clustering and partition clustering. Hierarchical clustering is a deterministic and agglomerative method. The output of such method is a dendogram, and the critical point of this method is the choice of a good threshold to partition it. Partitions clustering, like k-means, are stochastic methods, so they are sensitive to initial conditions and can converge to local minimum. However, they

are easy to implement and usable on very large datasets. Furthermore, choosing the parameters, like the number of centroids for k-means, remains difficult. In our work, we use a slightly different method for clustering the patterns. This latter method is particularly well adapted for comparing a set of patterns against a large dataset of patterns. Indeed, with this method, it is not necessary to know the distances between two patterns of the database. This saves a lot of computation time.

2.2.1. *Hierarchical clustering*

At the beginning of hierarchical clustering, there are as many cluster those motifs to compare. All possible pairs of cluster are compared, with a dissimilarity measure, and the most similar clusters are gathered. At the end of clustering, there is only one big cluster, containing all the motifs. A dendogram allows following gathering during clustering. The main interest of hierarchical clustering is that it allows seeing each step of clustering from dendrograms, and it produces a nice graphical output. However, partitionning the resulting dendogram is a hard task, requiring heuristics. Here, we apply an improved version of neighbour joining [18]. Figure 3 depicts an example of dendogram.

Fig. 3. An example of hierarchical clustering result

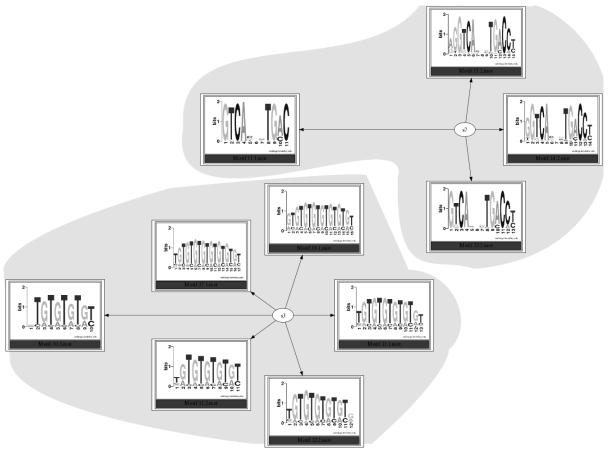


2.2.2. *Partition clustering*

Partition clustering stands for a set of methods to obtain a user defined and fixed number of clusters from the dataset. In the context of PWM clustering, Schones *et al.* [9] have used a k-medoids method. This latter method aims at finding the best configuration (assignment to each matrices to one of k chosen centers such that the squared error of the distances is minimal).

Choosing a good number of clusters is crucial, especially in the context of PWM clustering where obviously, several matrices are close and must be in the same cluster but a majority of other matrices are isolated and define their own cluster. In order to choose an appropriate number of clusters, a silhouette plotting is often used. Figure 4 shows an example of clusters obtained by a partition around medoids method.

Fig. 4. An example of partition clustering result



2.2.3. *Threshold clustering*

It is surely one of the simplest clustering method. Nevertheless, we find this method very appropriate for comparing datasets of motifs to databases. Here, one constructs a graph whose vertices correspond to all pattern matrices and there exists an edge between two vertices if and only if the distance between the associated matrices is below a given threshold. The clusters then correspond to the strongly connected components of the graph. Of course, this simple method will work with any kind of distances between matrices. In addition, this method ensures, by definition, that the closest matrices appear in the same cluster. Nevertheless, determining an appropriate threshold is crucial and it can be a difficult task depending on the scoring distance.

In this context, a good distance must ensure that there exists an important gap between matrices that can be considered as close and the other. It must ensure that matrices will be compared in a homogeneous scale which eliminates every distance (whose variances depend on the length of the aligned matrices) but the global score. In order to compute an appropriate threshold, one can consider as a characterisic parameter the number of edges in the final graph. It is null if the threshold is too restrictive and half of the square of the number of matrices if it is too permissive.

We have used this method in conjunction with the global score. We chose a threshold that guaranted to use 90% of the edges with a significant distance. This allows defining some clusters similar to Figure 5.

In addition, unlike hierarchical clustering and partition clustering, threshold clustering can deal with an incomplete distance matrix for constructing clusters. This can drastically reduce the comparison time when comparing a pattern set to a huge datatbase. Indeed, distances between pairs of motifs from the database have not to be computed in this case. It is sufficient to know the distance between every pairs of motifs in the set and between motifs from the set and every motifs from the database.

Fig. 5. An example of threshold cluster: Motifs are marked as black (motifs from the dataset) and white (motifs from the database).

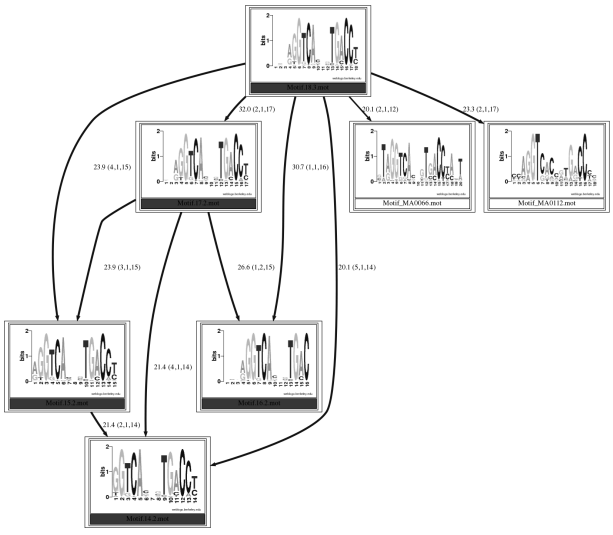
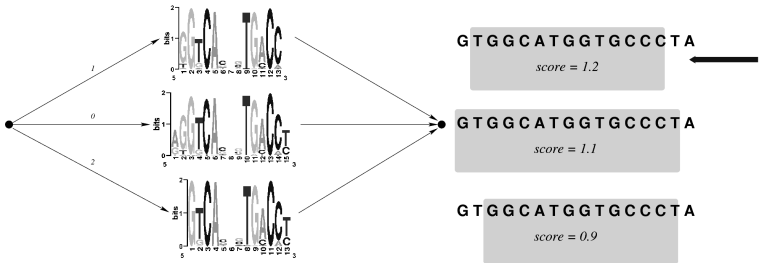


Fig. 6. An example of graph motif and an occurrence of such motif



2.2.4. *Graph motifs*

After the clustering phase, it appears natural to deal with a new style of pattern consisting of a composition of several alternative but close pattern matrices. This allows to increase the significativity of motifs. In this work, we define the so-called motif graphs and show how any classical operations can be defined for such new motifs. Notice that our graph motif representation is quite close to Hidden Markov Models representations developed in grammatical inference (see [19] for a short review of HMM classical models as well as a presentation of HMMER3). There, one focusses on chains of highly conserved patterns described by PWM with the opposite objective of our to increase the specificity of motifs regarding a set of sequences.

Definition 1. A graph motif is defined by a set $\mathcal{M}$ of pairs (P, o), where P is a pattern matrix and o is an integer offset.

Here, the offset described a kind of gap between the current position and the beginning position of the matrix. This can be summarized by Figure 6.

Occurrences of a graph motif Let us first recall that computing an appropriate score for pattern matrices is essential to determine if a position specific pattern matrix P of length n occurs in position k of a given sequence $\mathbb{S}$. One assigns a contribution $s_{i,\sigma}$ to each term $P_{i,\sigma}$ of the matrix P (in the case of PWM matrices, $s_{i,\sigma} = -\log_2 \frac{P_{i,\sigma}}{P_i p_\sigma}$). The global score $s(P, \mathbb{S}, k)$ (*i.e.,* the score of a n-length matrix P at position k in $\mathbb{S}$) is then defined as the sum

$$s(P, \mathbb{S}, k) = \sum_{i=1}^{n} s_{i,\mathbb{S}[k+i-1]},$$

where $\mathbb{S}[j]$ denotes the j-th symbol of sequence $\mathbb{S}$.

Such a definition translates in the context of graph motifs by taking the maximum of all possible scores.

Definition 2. The score $s(\mathcal{M}, \mathbb{S}, k)$ of a graph motif $\mathcal{M}$ at position k in $\mathbb{S}$ is defined by

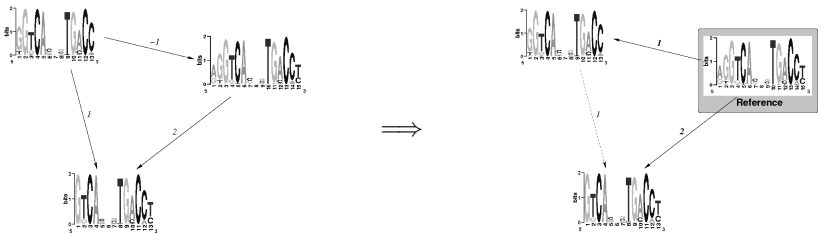
$$s(\mathcal{M}, \mathbb{S}, k) = \max_{(P,o)\in\mathcal{M}} \frac{1}{\ell(P)} s(P, \mathbb{S}, k + o),$$

where $\ell(P)$ stands for the length of motif P.

From graph clustering to graph motifs . In the previous section, one constructs some clusters that group several matrices together. The edges of this cluster are endowed with some relative offset between two matrices. It is easy to construct a graph motif from such a structure by choosing a particular vertex of the cluster as a reference and by computing the offsets of all the remaining matrices relatively to this vertex.

Figure 7 shows an example of transformation from a clustering of pattern to a graph motif. First, one choose a reference pattern. Then one computes the offset between all the other patterns to the reference.

Fig. 7. Transformation from a cluster of pattern to a graph motif



Consensus for a graph motif. For several applications, it is important to deal with a single but representative pattern matrix. It is possible to construct such consensus matrices from a graph motif (that is supposed to represent the same motif with some minor differences). The consensus matrix consists in a alignment of a sub-window of each occurrences of a graphe motif. An appropriate choice for defining a sub-window is to take the one defined by the longest pattern matrix of the graph motif. Is this case, it is more convenient to compute all the offset values with reference to this central motif as depicted in Figure 8.

3. Implementation tricks

In order to have an exhaustive comparison of all the pattern matrices, one has to compute a distance between any pairs of the pattern set that is a maximum over all possible shifts between the two matrices and any sub windows of the aligned matrices. Our principle is summarized in Figure 9.

Fig. 8. An example of consensus built from a graph motif

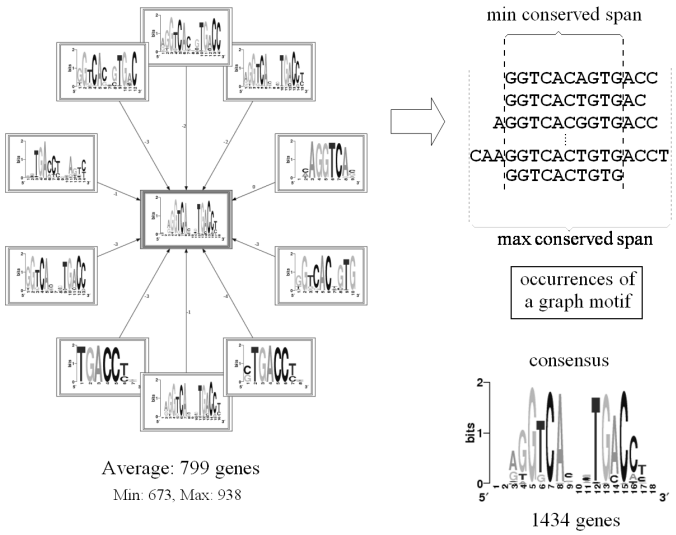
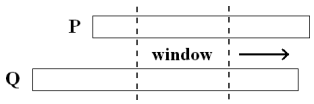


Fig. 9. Principle of the comparison scheme between two matrices. All possible shifts and all possible sub-windows are compared.



Computing the distance between two matrices depends linearly on the length of the matrices. This results in a naive complexity of order n^2k^3, where n is the number of matrices and k is the length of the matrices.

We have used two implementation tricks that reduce drastically the execution time when comparing large sets of matrices. First one uses a parallelization technic to reduce the complexity by a factor depending on the capacity of the server that executes our application. Next, one draws an iterative technics that reduces the number of comparisons by a factor k by using a recursive definition of our global distance. Finally, threshold clustering methods do not require to know the complete distance matrix. For instance, it is not necessary to compare the pattern matrices of the databases together. The final complexity of our clustering method is $\frac{n(n+N)k^2}{C}$, where N is the number of motifs in the database, n is the number of motifs in the pattern set, k is the maximal length of a pattern in the set, and C is the number

of cores of the server.

3.1. *Parallelization*

First notice that the computation of each distance between two matrices is independent. Such a task can be performed in parallel by some independent processes in a multi-core framework. Our implementation assigns the computation of all the distances of a matrix at one particular process. The distances are then written in the shared memory when the process finishes. This avoids the classical bottlenecks of parallelized application that only one process can access the shared memory at a time. Thus having too many accesses to the shared memory increase the computation time.

By using this trick, we have reduced the computation time of all the distances by a factor seven on an eight core computer.

3.2. *Iterative computation of the global score*

Notice now that for a fixed alignment of the two matrices, one computes the maximum global score for several different sizes of windows. We now show that these scores are dependent and they can be computed iteratively. This is mainly involved by the fact that $\widehat{E(T)}$ and $\widehat{O(T)}$ are sums as defined in (1). Indeed, let us consider the sum $R(T) = \sum_{(E,O)\in T} \frac{O^2}{E}$. It is obvious that $R(T \cup (\alpha, \beta))$ can be calculated in constant time when $R(T)$ is known. It is also obvious that $K(T)$ can be computed in constant time when $R(T)$ is known. As a consequence, a dynamical programming algorithm that reports iteratively $K(T)$ can easily be designed. It then allows to compute the distance between two aligned matrices for increasing sizes of windows in linear time.

4. Results and discussion

4.1. *Software availability*

The complete method has been developped as a web application. The core of the implementation is done in PHP 5 language making the application platform independent. A first online version is available on our servers[a]. This version implements a hierarchical clustering that allows to obtain an informative representation of the cluster set. A threshold clustering methods is also implemented. It permits to compare a set of motifs to JASPAR and TRANSFAC free public releases. All comparisons are based on our global distance measure for the moment. Nevertheless, we plan in the close future to add some other distance score such as Kullback-Lebler and other used distances.

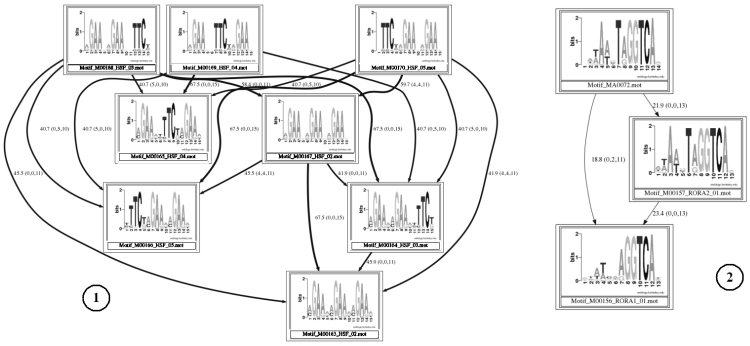
[a] `http://cardioserve.nantes.inserm.fr/madtools/motifsComparator`

4.2. *Comparing JASPAR and Transfac databases*

The main transcription factors binding site databases are JASPAR and Transfac. Our analysis is based on the latest free release of Transfac (Transfac Rel.3.2) and JASPAR CORE 2008 dataset. Transfac release 3.2 includes 261 matrices rather than JASPAR contains only 138 PWM. JASPAR is smaller than Transfac because of a manual curation that eliminates redundancies in JASPAR.

Comparison of all PWM available in these two databases (Figure 10) allows to find redundancies between the two databases, but also redundant elements in Transfac. They appear in the same cluster as for instance Figure 10-2. Here, one presents the transcription factor sub-family ROR (Retinoic Acid receptor relatead Orphan Receptor). We have found redundancies of RORA between JASPAR (MA0071) and Transfac (M00156,M00157). Besides, several transcription factor families are reconstructed, like the HSF family (figure 10-1). Indeed, we have reconstructed a graph motif with the entire HSF element available in Transfac. These results allow a biological validation of our global comparison method.

Fig. 10. A example of two clusters extracted from a comparison between JASPAR and Transfac databases



4.3. *Comparing several experiments*

One of the aims of this method is to facilitate ChIP-chip data analysis. Indeed, motifs discovered in input sequences are compared together to eliminate redundancies, with public databases to identify known transcription factor binding site and to other ChIP-chip experiments to find some conserved patterns. We have used our method to analyze a study of Carroll et al. [20] on ChIP-chip data for ESR1 transcription factor. Elimination of redundancies is essential in this study because of the number of motifs to analyze. Furthermore, using a consensus pattern allows to increasing significantly the number of sequences concerned by the motif (Figure 8).

Comparison of motif with public databases allows to identify corresponding transcription factor binding site. In the case of ChIP-chip ESR1, we have compared all the motifs discovered by MDmodule[b] to JASPAR and Transfac databases (Figure 5). Several motifs are found similar to the best input motif. They correspond to PPARgamma and T3R. PPARgamma, T3R and ESR1 are part of the nuclear receptor superfamily, and more specificly to nuclear hormone receptors family (NHR). NHR dimers bind to regulatory sequences composed of two half-sites, where half-sites have the consensus sequence AGGTCA as showed in [10]. Gathering of such motif is so expected.

4.4. *Building Motif networks*

Comparison of motif found in several ChIP-chip experiments allows to construct regulatory network, where known motifs are represented by transcription factor name and unknown by consensus motif (Figure 11). We have identified various groups of genes, represented in circle containing their number. They are co-regulated by a subset of motifs and are associated to particular functions. This is proven by mapping a significant functional annotation (obtained with GOminer tool [21]) of each gene cluster. Comparison of several ChIP-chip experiments allows to validate interaction between transcription factors and a set of deregulated genes and to refine network. Significative functional annotations make sense on each genes cluster and can assist to define putative therapeutic target.

5. Conclusion

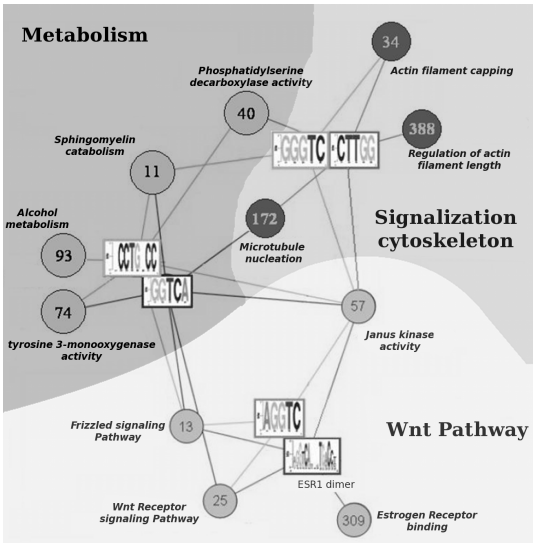
We have presented a complete method to study some sets of motifs described by pattern matrices like PWM, PSSM or PFM. Eliminating redundancies requires to compare all motifs together, but also with reverse complement of all motifs and to databases. It is thus necessary to use some tricky implementation technics in order to deal with large sets of motifs, like those arising in the study of ChIP-chip data for instance. Finally, as a complementary result, by comparing a pattern matrix to itself, to its shifts and to its reverse complement, one can determine if the motif is a palindrome, a common characteristic of TFBS, or a periodic motif that usually represents artifacts of the discovery step.

Aknowledgments

The authors deeply appreciate anonymous referees comments that significantly improve the quality of the paper. They also would like to thank Mireille Régnier for many fruitful discussions around this work. Finally, this work is partially supported by BIL regional research program.

[b]MDmodule is the successor of MDscan tool. It is a part of MotifRegressor pattern discovery package [6]

Fig. 11. An example of motif networks for a ChIP-chip experiments on ESR1



References

1. G. D. Stormo, T. D. Schneider, L. Gold, and A. Ehrenfeucht, "Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli," *Nucleic Acids Res.*, vol. 10, pp. 2997–3011, May 1982.
2. R. Staden, "Computer methods to locate signals in nucleic acid sequences," *Nucleic Acids Res.*, vol. 12, pp. 505–519, Jan 1984.
3. T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," *Proc Int Conf Intell Syst Mol Biol*, vol. 2, pp. 28–36, 1994.
4. J. D. Hughes, P. W. Estep, S. Tavazoie, and G. M. Church, "Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae," *J. Mol. Biol.*, vol. 296, pp. 1205–1214, Mar 2000.
5. X. S. Liu, D. L. Brutlag, and J. S. Liu, "An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments," *Nat. Biotechnol.*, vol. 20, pp. 835–839, Aug 2002.
6. E. M. Conlon, X. S. Liu, J. D. Lieb, and J. S. Liu, "Integrating regulatory motif discovery and genome-wide expression analysis," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 100, pp. 3339–3344, Mar 2003.
7. K. D. MacIsaac and E. Fraenkel, "Practical strategies for discovering regulatory DNA sequence motifs," *PLoS Comput. Biol.*, vol. 2, p. e36, Apr 2006.
8. S. Gupta, J. A. Stamatoyannopoulos, T. L. Bailey, and W. S. Noble, "Quantifying similarity between motifs," *Genome Biol.*, vol. 8, p. R24, 2007.
9. D. E. Schones, P. Sumazin, and M. Q. Zhang, "Similarity of position frequency matrices for transcription factor binding sites," *Bioinformatics*, vol. 21, pp. 307–313, Feb 2005.
10. A. Sandelin, W. Alkema, P. Engström, W. W. Wasserman, and B. Lenhard, "JASPAR:

an open-access database for eukaryotic transcription factor binding profiles," *Nucleic Acids Res.*, vol. 32, pp. D91–94, Jan 2004.

11. E. Wingender, X. Chen, R. Hehl, H. Karas, I. Liebich, V. Matys, T. Meinhardt, M. Prss, I. Reuter, and F. Schacherer, "TRANSFAC: an integrated system for gene expression regulation," *Nucleic Acids Res.*, vol. 28, pp. 316–319, Jan 2000.
12. J. Liu, A. Neuwald, and C. Lawrence, "Bayesian models for multiple local sequence alignment and gibbs sampling strategies," *Journal of the American Statistical Association*, pp. 90–432, 1995.
13. S. Kullback, *Information Theory and Statistics*. Wiley, New York, 1959.
14. S. Aerts, P. Van Loo, G. Thijs, Y. Moreau, and B. De Moor, "Computational detection of cis -regulatory modules," *Bioinformatics*, vol. 19 Suppl 2, pp. 5–14, Oct 2003.
15. S. Roepcke, S. Grossmann, S. Rahmann, and M. Vingron, "T-Reg Comparator: an analysis tool for the comparison of position weight matrices," *Nucleic Acids Res.*, vol. 33, pp. W438–441, Jul 2005.
16. S. Pietrokovski, "Searching databases of conserved sequence regions by aligning protein multiple-alignments," *Nucleic Acids Res.*, vol. 24, pp. 3836–3845, Oct 1996.
17. T. Wang and G. D. Stormo, "Combining phylogenetic data with co-regulated genes to identify regulatory motifs," *Bioinformatics*, vol. 19, pp. 2369–2380, Dec 2003.
18. L. Sheneman, J. Evans, and J. A. Foster, "Clearcut: a fast implementation of relaxed neighbor joining," *Bioinformatics*, vol. 22, pp. 2823–2824, Nov 2006.
19. S. R. Eddy, "A probabilistic model of local sequence alignment that simplifies statistical significance estimation," *PLoS Comput. Biol.*, vol. 4, p. e1000069, May 2008.
20. J. S. Carroll, C. A. Meyer, J. Song, W. Li, T. R. Geistlinger, J. Eeckhoute, A. S. Brodsky, E. K. Keeton, K. C. Fertuck, G. F. Hall, Q. Wang, S. Bekiranov, V. Sementchenko, E. A. Fox, P. A. Silver, T. R. Gingeras, X. S. Liu, and M. Brown, "Genome-wide analysis of estrogen receptor binding sites," *Nat. Genet.*, vol. 38, pp. 1289–1297, Nov 2006.
21. B. R. Zeeberg, W. Feng, G. Wang, M. D. Wang, A. T. Fojo, M. Sunshine, S. Narasimhan, D. W. Kane, W. C. Reinhold, S. Lababidi, K. J. Bussey, J. Riss, J. C. Barrett, and J. N. Weinstein, "GoMiner: a resource for biological interpretation of genomic and proteomic data," *Genome Biol.*, vol. 4, p. R28, 2003.