# École Normale Supérieure

Master 2 IMaLiS - INTERDISCIPLINARY MASTER IN LIFE SCIENCES

**_matrix-clustering:_ a novel tool to cluster and align Transcription Factor binding motifs.**

by

CASTRO-MONDRAGÓN Jaime Abraham

A Report Submitted in partial fulfillment of the requirements for the degree:

***Master in Life Sciences***

Supervisor: Jacques VAN HELDEN

Lab. Technological Advances for Genomics and Clinics (TAGC)

**Paris, France** **JUNE 2014**

# Table of Contents

# ABSTRACT

Transcription factors binding motifs (TFBM) are classically represented either as consensus strings (IUPAC, regular expressions), or as position-specific scoring matrices (PSSM). Thousands of curated TFBM are available in specialized databases (JASPAR, RegulonDB, TRANSFAC, etc), built from collections of transcription factor binding sites (TFBS) obtained from various experimental methods (e.g. EMSA, DNAse footprinting, SELEX). TFBM can also be discovered *ab initio* from genome-scale data sets: promoters of co-expressed genes, ChIP-seq peaks, phylogenetic footprints, etc.

Motif collections sometimes contain groups of similar motifs, for different reasons: curation of alternative motifs for a same TF; homologous proteins sharing a particular DNA binding domain, motifs discovered with analytic workflows combining several algorithms (e.g. RSAT *peak-motifs*, or MEME-chip). In order to address the increasing need for efficient tools enabling to discover groups of similarities among motif collections, we developed *matrix-clustering*, which presents significant advantages over existing solutions.

1) Segmentation of the input set of TFBM into separated clusters, displayed as a motif forest rather than a single motif tree (alternative software tools force all motifs to be aligned).

2) Multiple alignment of all motifs belonging to a same cluster.

3) User-friendly display of motif trees with aligned logos and consensuses.

4) At each level of the hierarchical tree, computation of a merged motif (matrix and consensus) summarizing all the descendant motifs.

5) Support for a large series of alternative metrics (correlation, Euclidian distance, SSD, Sandelin-Wasserman, logo dot product, and length-normalized version of these scores).

6) Possibility to select a custom combination between these scores to compute an integrative threshold.

The potentialities of the tool are illustrated by study cases: clustering of matrices extracted from ChIP-seq peaks using several motif discovery algorithms. Extraction of a motif-to-motif network and clustering of all motifs from the JASPAR taxon-wise collections. The significance of the clustering results is further assessed by analysing collections of randomized matrices (column-permuted). In this negative control, most motifs are correctly assigned to a singleton, except for low complexity motifs (e.g. AAAAAA).

We analyzed the effect of hierarchical clustering parameters (hierarchical agglomeration rule, similarity metrics) on the number of clusters and on the relationships between motifs, and identified suitable parameters to obtain relevant results.

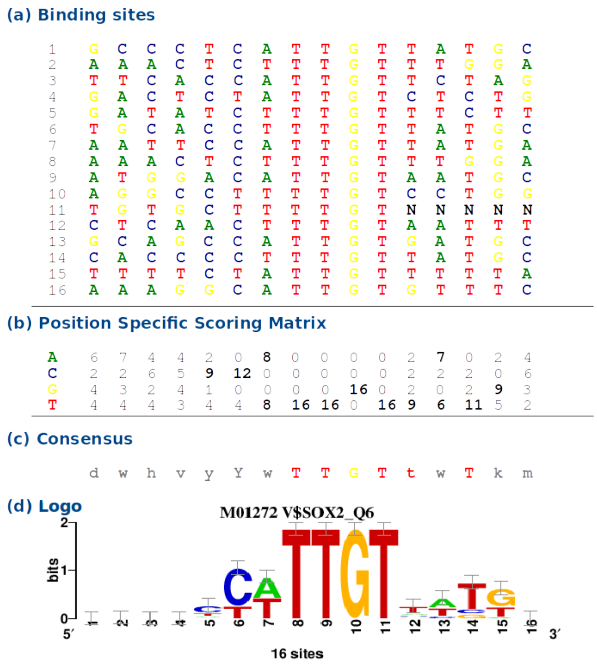
Availability: *matrix-clustering* is available on the Regulatory Sequence Analysis Tools (RSAT) Web site (RSAT; http://www.rsat.eu/). It can also be downloaded with the stand-alone RSAT distribution to be run from the  Unix shell.

# INTRODUCTION

Gene expression is a process strongly regulated in the cell at different levels (transcription, translation) by distinct molecules (proteins, RNAs). At transcriptional level, gene expression can be driven by a set of proteins known as Transcription Factors (**TFs**) which act either as activators or repressors of selected target genes by binding DNA in a sequence-specific manner[1,2]. TFs have a DNA-binding domain in which a few amino acid residues interact via weak bonds with specific nucleotides[3,4]. The DNA sequences where a TF binds are denoted as TF binding sites (**TFBSs**).

The TFBSs vary in width between 5-30 nucleotides long[5]. Although the TFBSs of a particular TF are similar to each other, they are not identical: usually they have a few well-conserved positions, whereas some other positions show residue variations between sites.

The study of TFs and their TFBSs has allowed the creation of transcriptional regulatory networks and the discovery of which particular combination of TFs give rise to complex and diverse biological processes as morphogenesis, cell differentiation, development, etc[6,2].


**(a) Binding sites**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | G | C | C | C | T | C | A | T | T | G | T | T | A | T | G | C |
| 2 | A | A | A | C | T | C | T | T | T | G | T | T | T | G | G | A |
| 3 | T | T | C | A | C | C | A | T | T | G | T | T | C | T | A | G |
| 4 | G | A | C | T | C | T | A | T | T | G | T | C | T | C | T | G |
| 5 | G | A | T | A | T | C | T | T | T | G | T | T | T | C | T | T |
| 6 | T | G | C | A | C | C | T | T | T | G | T | T | A | T | G | C |
| 7 | A | A | T | T | C | C | A | T | T | G | T | T | A | T | G | A |
| 8 | A | A | A | C | T | C | T | T | T | G | T | T | T | G | G | A |
| 9 | A | T | G | G | A | C | A | T | T | G | T | A | A | T | G | C |
| 10 | A | G | G | C | C | T | T | T | T | G | T | C | C | T | G | G |
| 11 | T | G | T | G | C | T | T | T | T | G | T | N | N | N | N | N |
| 12 | C | T | C | A | A | C | T | T | T | G | T | A | A | T | T | T |
| 13 | G | C | A | G | C | C | A | T | T | G | T | G | A | T | G | C |
| 14 | C | A | C | C | C | C | T | T | T | G | T | T | A | T | G | C |
| 15 | T | T | T | T | C | T | A | T | T | G | T | T | T | T | T | A |
| 16 | A | A | A | G | G | C | A | T | T | G | T | G | T | T | T | C |

**(b) Position Specific Scoring Matrix**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 6 | 7 | 4 | 4 | 2 | 0 | **8** | 0 | 0 | 0 | 0 | 2 | **7** | 0 | 2 | 4 |
| C | 2 | 2 | 6 | 5 | **9** | **12** | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 0 | 6 |
| G | 4 | 3 | 2 | 4 | 1 | 0 | 0 | 0 | 0 | **16** | 0 | 2 | 0 | 2 | **9** | 3 |
| T | 4 | 4 | 4 | 3 | 4 | 4 | **8** | **16** | **16** | 0 | **16** | **9** | **6** | **11** | 5 | 2 |

**(c) Consensus**

d w h v y Y w T T G T t w T k m

**(d) Logo**

**Figure 1. Representations of TFBMs built from a collection of TFBSs.** This example is illustrated with the Sox2 motif from Transfac (M01272). (a) Alignment of the 16 annotated TFBSs. (b) PSSM representation: each cell of the matrix indicates the number of ocurrences of each nucleotide (row) on each column of the aligned sites. (c) IUPAC representation of the degenerated consensus. (d) Logo representation.

## Transcription factor binding motifs (TFBM)

The fact that one TF could bind to a large set of similar (but slightly different) sequences makes the searching of putative TFBSs a complex task[7]. It is specially important to find a model that: (1) encompasses and represents all the already known TFBSs for a TF, (2) is informative and useful to search for new TFBSs. To build the model it is required to collect and align a sufficient number of TFBSs, in order to extract significant information about the conserved and variable residues (Figure 1a). Such models summarizing the conserved and variable residues among a collection of reference sites are named “transcription factor binding motif” (**TFBM**). The most common representation modes for TFBM are based either on character strings (strict consensus, IUPAC code, regular expressions) or on position-specific scoring matrices (**PSSMs**). Figure 1 shows different ways to represent TFBMs using as example the Sox2 TF.

### *String-based representations of TFBM*

Consensus sequences can be represented either as regular expressions, or using the IUPAC alphabet[8] to denote the combination of nucleotides at each position of the alignment (Figure 1b). Both methods enable to represent positions with variable residues. However, they do not take into account the nucleotide frequency at each position of the alignment. For example, in figure 1, the letter Y at the 6th position of the consensus means “C or T”, this is not informative about the respective frequencies of these two residues in this column of the aligned sites.

### P*osition specific scoring matrices (PSSMs)*

Position-specific scoring matrices[9] indicate the number of occurrences of each residue (rows) in each column of the aligned sites. This model captures the nucleotide variability and conservation of a collection of TFBSs (Figure 1c). The PSSMs allow observe that many positions of the alignment have higher frequency associated to a specific nucleotide. A convenient way to provide a visual and intuitive representation of the PSSMs is the *sequence logo,* which indicates the information content within each column of the matrix (Figure 1d). Currently the PSSMs are the most extensively used computational method to search TFBSs in a sequence[7] because they take into account the differences of nucleotide composition between the TFBSs and the analyzed sequences, the search is supported by different statistical approaches to validate the putative TFBSs.

### *Collections of reference position-specific scoring matrices*

Several specialized databases provide PSSMs built from collections of TFBSs, for example RegulonDB[10], JASPAR[11], TRANSFAC[12] , etc. The process to build PSSMs is generic and automatized as part of analysis of biological sequences and in the study of TFs. Software packages such as RSAT[13] or MEME suite[14], allow to build PSSMs from input sequences. This task is relatively easy when we already know the TFBSs (e.g. collection of binding sites characterized by gel shift or DNAse protection experiments), but becomes more complicated when the precise TFBSs are not known, and we only dispose of a set of relatively large sequences where a TF possibly binds (e.g. promoters of co-expressed genes). To address this situation, one uses a bioinformatic approach known as *de novo* motif discovery, which attempts to detect significant motifs[15,16,17] (e.g. over-represented, or positionally biased) in a set of sequences, and build PSSMs from them. This has been a fundamental problem in computational biology since many years,  and a variety of motif discovery algorithms have been designed[18], for example, searching overrepresented oligonucleotides for monomeric TFs, spaced oligonucleotides for dimeric TFs, positional distribution of sites inside the ChIP-seq peaks, overrepresented words in  windows of variable or fixed size, etc.

For the cases of high-throughput experiments (genomic Selex, ChIP-seq, microarrays) or studies of conservation of cis-regulatory elements across species[19] *de novo* motif-discovery tools have to analyze large sets varying from a few hundreds to severals tens of thousand sequences. More than one algorithm can be is used[20], complementing themselves for their limitations: some of them find the motifs that others do not, but sometimes the same motifs are found by more than one algorithm and hence they could be almost similar and hence redundant with small variations in size and nucleotide frequencies at some positions. Once a set motifs has been discovered, the user is confronted to the next question: do the different motifs found correspond to known TFBMs ?

*Motifs comparison metrics*

Actually this questions is one of the challenges on the field, many efforts have been done to develop statistical methods and to find adequate metrics to compare the motifs, although there are plenty of these metrics each one uses different statistical approaches, each one with its own limitations. For these reason it must be mentioned that there is no a standard statistical method neither a standard metric to measure the similarity between PSSMs, and this issue has been discussed in several publications[21,21,23,24]. Currently there are at least 3 software tools which measure the similarity between motifs, *compare-matrices* available in RSAT, *TomTom*[21,24] in MEME suite and STAMP[23].

The free software package RSAT[13] integrates a collection of tools for detection and analysis of cis-regulatory elements in genomic sequencea. RSAT includes the program *compare-matrices* which measures the similarity among a set of motifs against a plenty of motifs databases. Unlike others motif comparison tools, it enables to compute several metrics in the same analysis and then selects the best matches using rankings statistics on the combined scores. A drawback is that the current version does not compute p-values on the different scores.

Matching discovered motifs against reference databases is not the only challenge to compare motifs. Another application is to regroup the redundant motifs discovered from the same sequences. Both issues are faced, among others, by the tools[20,25,26,27,28] to analyze ChIP-seq data. Some of these tools use many motif discovery variants to search exceptional motifs in the peaks, after found the motifs the next step is motif comparison, but given the redundancy in the found motifs, the results could be difficult to interpret. As part of motif analysis, it should be useful to group similar motifs.

## Objectives

Knowing either the value of similarity and the offset among all the pairs a of a set of motifs could be useful information that can be integrated to group the motifs in clusters and align them, this approach could have many applications for example: (1) it could help to simplifying the interpretation of results, (2) to help to find compound-motifs, (3) to highlight the common positions between a set of motifs[23].

In order to address the increasing need for efficient tools enabling to discover groups of similarities among motif collections, in this project it had been created the tool *matrix-clustering* which is a tool that combines motif clustering and motif alignment.

# MATERIAL AND METHODS

## Software tools

Motif comparison is done using the tool *compare-matrices.* The tool *convert-matrix* is used to add empty columns on the flanks of PSSM in order to align them, to generate the logos alignment, to change the orientation of the motifs, and to permute the columns of the matrices for the negative control. The tool *merge-matrices* is used to create the merge-level matrices and consensuses at each branch of the trees. These tools used in this work are available at Regulatory Sequence Analysis Tools[13] (RSAT).

The logo trees is done with D3 which is a JavaScript library for manipulating documents based on data (http://d3js.org/).

Motifs studied in study cases 1 and 2 were analyzed with STAMP[23], a tool to cluster, compare and align motifs.

## Motif datasets

For the study case 1, I used a set of 21 motifs discovered from the peaks set of Oct4 ChIP-seq from Chen et al[6] with the tool *peak-motifs*[20,29].
For the study case 2, I used the non-redundant sets of insect and vertebrates core motifs from the JASPAR[11] database.

## Implementation

*matrix-clustering* was implemented in PERL,R and the JavaScript library D3.

# RESULTS

## *Development of the software tool matrix-clustering*

In this work I present a novel bioinformatic tool called *matrix-clustering* to face with one of the current challenges in the analysis of cis-regulatory sequences: the clustering of motifs. This tool is now functional and available on the Regulatory Sequence Analysis Tools[13] (RSAT) Web site (http://www.rsat.eu/). It can also be downloaded with the stand-alone RSAT distribution to be used on the Unix shell, alllowing to include it in automated pipelines.

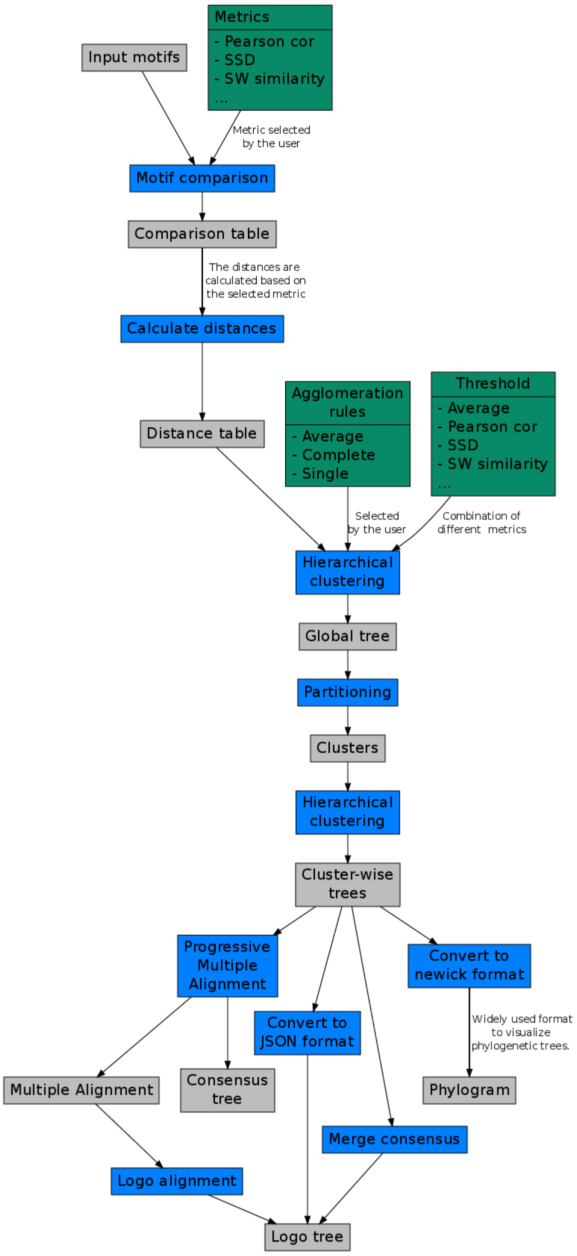


**Figure 2. *matrix-clustering* pipeline** The figure shows the pipeline from the input motifs and parameters selected by the user to the final output and the interconnections between the programs and files. Grey boxes represent input/output files. Blue boxes represent software tools used in this algorithm. Green boxes represent the user selection parameters.

This tool takes as input a set of motifs (PSSMs), measures the similarity between each motif pairs runs hierarchical clustering to group the similar motifs. The clusters are defined based on one or more metrics selected by the user. Once the clusters are defined, they are displayed in separated trees, which are used as guide trees to produce a progressive alignment of the motifs. The result is displayed in different modes: logo phylogram, logo cladogram, and consensus tree. Figure 2 shows the flowchart of the algorithm, which is explained below.

### *Input files*

*matrix-clustering* takes as input a file with a set of motifs (several formats enabling to store multiple PSSM in a file are supported: MEME, transfac, tab-delimited, etc).

### *Motif comparison*

All the input motifs are compared each other using the program *compare-matrices*, which computes the similarities using many metrics (correlation, Euclidian distance, SSD, Sandelin-Wasserman, logo dot product, and length-normalized version of these scores). All the pairwise comparisons are exported in a tab-delimited file, which can be accessed from the *matrix-clustering* result page.

### *Distance calculation*

Before the clustering step it is necessary select one of the supported metrics, which will be used to build the motif-to-motif distance table. However some of the metrics supported by *compare-matrices* measure a similarity (e.g. correlation, normalized correlation) whereas others measure a distance (Euclidian, sum of squared deviations, Sandelin-Wasserman). Since hierarchical clustering assumes a distance table as input, the values of the selected metric are transformed into distance values and the resulting table with all the resulting distances among each pair of the motifs is used for the hierarchical clustering step. This distance table is also exported in the *matrix-clustering* results.

### *Hierarchical clustering*

After having calculated the distance table between all the motifs, the hierarchical clustering approach is applied to produce a *global tree* encompassing all input motifs. By default *matrix-clustering* uses the average linkage method, but the user can select others (complete or single linkage).

The global tree is then partitioned by applying one or more user-selected thresholds to each agglomeration step: if the nodes fail to satisfy any of the thresholds, a new cluster is created. It must be noticed that the tree topology and number of clusters can change according to two factors: 1) the merge method selected and 2) the metric selected to create the clusters (see results below). At the end of this process the clusters are defined. This step is implemented in R.

*Progressive alignment*

After having partitioned the global tree into clusters, the subtree corresponding to each cluster is used as a guide to align the matrices at each agglomeration level of the tree. First, the motifs (matrices) are orientated (forward or reverse) and then they are shifted by adding empty columns (gaps) at the beginning or the end of the matrix. Note that this algorithm does not add internal gaps, in contrast with STAMP, which uses dynamical programming and allows to produce global or local alignments between matrices. The number of flanking gaps added is recalculated on each step of the alignment. The criteria to align the motifs are the same rules applied in the agglomeration steps of hierarchical clustering  (average, complete, single linkage). The result of this process is one global multiple alignment for each cluster, unlike other algorithms where all the input motifs are forced to be aligned into a single alignment. At the end of the analysis a dendrogram tree in PNG and PDF format is exported, highlighting with different colors each cluster and their motifs with their aligned consensuses. At the end of this step, *matrix-clustering* exports a tab-delimited file showing the alignment of each cluster. This step is implemented in R.

*Branch-wise matrices, logos and consensuses*

During the agglomeration step of each cluster, once the motifs have been aligned and extended to occupy the same width, *matrix-clustering* calculates a *branch-wise matrix* by summing or averaging the frequencies of the aligned motifs at each level of the hierarchical tree. The computation of branch-wise matrices is done with the program *merge-matrix* implemented in Perl. These branch-wise matrices are then used to generate *branch-wise* consensus and a *branch-wise logo* (using *convert-matrix*), which highlight the common residues of the aligned motifs. As more branches are merged the resulting consensus will be more similar to the consensuses of the descendant motifs. On the logo cladogram, internal nodes are labeled with consensuses, and clicking on a branch-wise consensus opens a link to the corresponding branch-wise logo.

*Tree export*

The tree obtained with the hierarchical clustering approach can be converted and exported to the newick format which is a widely used textual format to represent phylogenetic trees.

To show the logo alignments, for each cluster the trees obtained with the hierarchical clustering approach are converted and exported to the JavaScript Object Notation (JSON) format.

*Phylogram*

The program can take the tree in newick format to create a phylogram (i.e. a tree where the branch lengths are scaled to reflect distances) using JavaScript.

*Consensus alignment*

Once the progressive alignment is done, *matrix-clustering* represents this alignment in two versions, either as a consensus phylogram, or as a logo cladogram. For the consensus tree both PNG and PDF files are exported. The gaps in this alignment are represented with '-' and clusters are highlighted on the tree with different colors and numbers. In addition, each leaf of this tree displays the motif ID and the strand. This step is implemented with R.

*Logos alignment*

For each cluster, a logo cladogram is displayed and beside each branch is displayed its own logo aligned in both orientations with its own consensuses and IDs of the leaves. In this alignment the gaps are represented by empty columns (filled with zeros) at the beginning and/or at the end of the original logos of the motifs. This step is implemented with the JavaScript library D3 and HTML5.

## Evaluation of the matrix-clustering results and selection of relevant parameters

### ***Case study 1: grouping redundant matrices resulting from multiple motif discovery tools***

As a first study case, I evaluated the capability of matrix-clustering to identify groups of similar motifs within a set of matrices resulting from the program *peak-motifs*[20,29] which runs four complementary motif discovery algorithms to detect exceptional motifs within the peaks. Each algorithm complements the limitation of the others, sometimes similar motifs are found by more than one algorithm with small differences in width or nucleotide frequencies at some positions, resulting in a redundant motif set. The aim of this example is to demonstrate how a set of motifs can be grouped and aligned to facilitate its interpretation.

Oct4 (Pou5f1) is an essential TF in cell fate decision, ES cells and early embryonic development, it binds the canonical sequence 5'-ATGCAAAT-3'. In ES cells, Oct4 often interacts with another TF, Sox2, which binds to an adjacent Sox motif 5'-CATTGTA-3. Together, both TFs co-regulate specific genes[6,30]. During the analysis of Oct4 or Sox2 binding peaks, the so-called *SOCT* motif is usually found, which is a composite motif encompassing both Oct and Sox motifs.

To illustrate this situation, I collected the 21 motifs found using *peak-motifs* from ChIP-seq peaks obtained by immunoprecipitating Oct4 in ES cells, and analyzed them using *matrix-clustering*. Given these 21 discovered motifs, is it possible to group all of these motifs supposedly corresponding to the immunoprecipitated factor? If not, do some motif clusters correspond to alternative motifs potentially bound by other TFs or composite motifs?

The 21 motifs were analyzed with *matrix-clustering* using the average-linkage method as agglomeration rule and threshold values on correlation (cor >= 0.6) and normalized correlation (Ncor >= 0.4). With these parameters the program identified six clusters (Figure 3a). The result is displayed as a motif forest rather than a motif tree: each cluster is displayed as a separate hierarchical tree, rather than one tree grouping all the motifs as done by STAMP (Figure 4). On the rightmost side are displayed the aligned logos in both orientations highlighting local and global similarities between motifs of the same cluster. Each internal branch is labeled in blue with the consensus encompassing the descendent motifs, summarizing the informative positions of the motif variants returned by different algorithms. In this example, in the names of the leaves are represented the names of the motif discovery algorithm which produced the motif (*oligo-analysis, dyad-analysis, local-words, postion-analysis*). It must be noticed that similar motifs can be discovered by more than one of these algorithms.

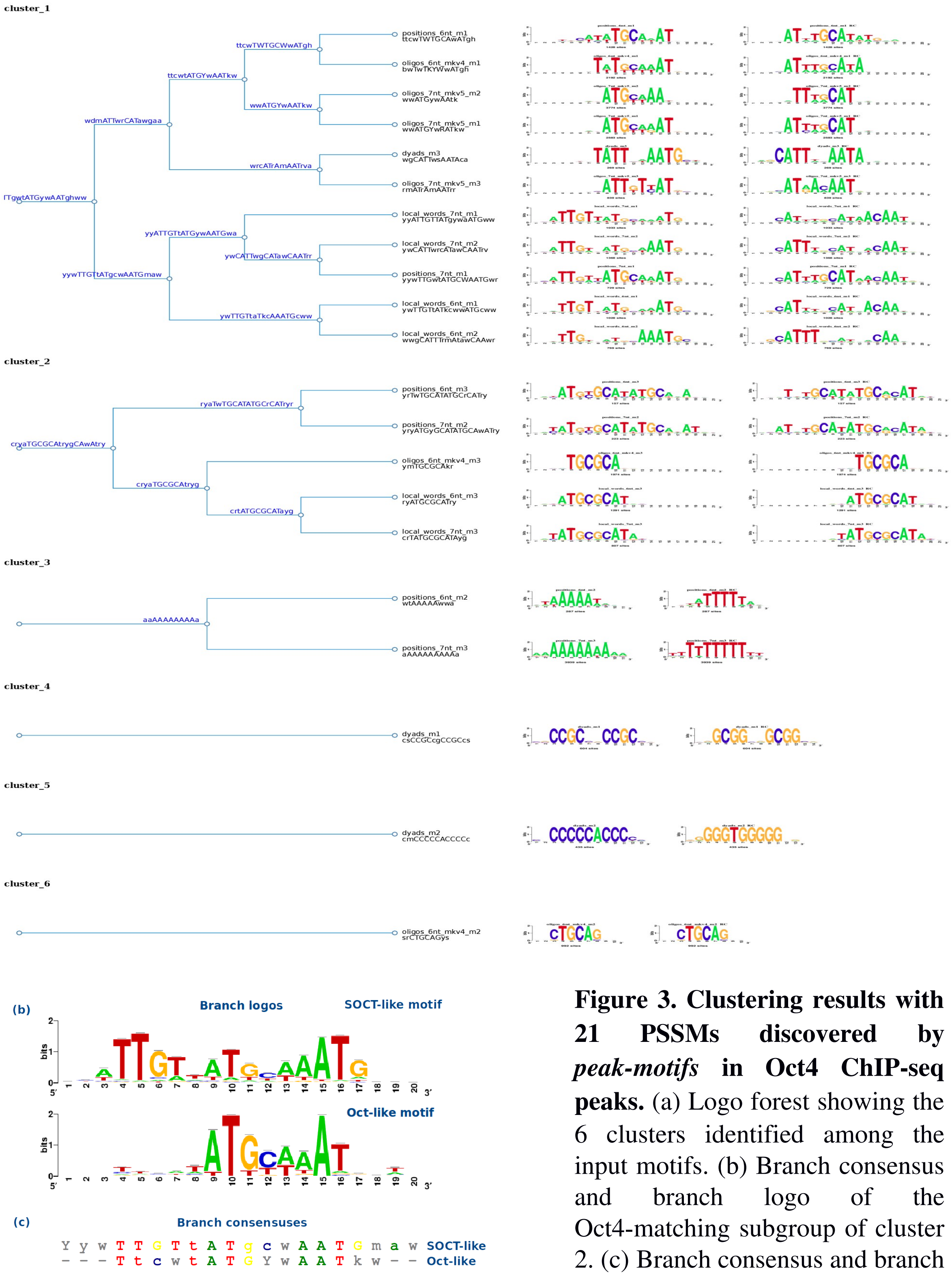
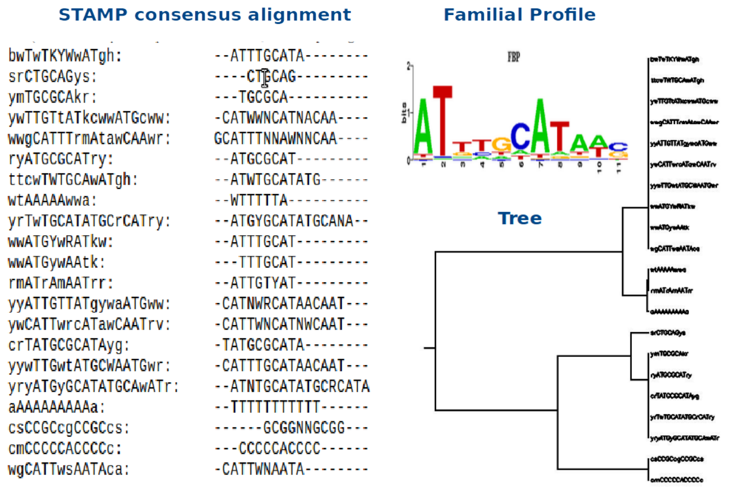


**Figure 3. Clustering results with 21 PSSMs discovered by *peak-motifs* in Oct4 ChIP-seq peaks.** (a) Logo forest showing the 6 clusters identified among the input motifs. (b) Branch consensus and branch logo of the Oct4-matching subgroup of cluster 2. (c) Branch consensus and branch logo of the *SOCT*-matching subgroup.

The largest cluster (cluster 1) regroups 11 motifs. The upper subgroup contains 6 motifs matching the Oct4 matrix from JASPAR. These Oct-like motifs are summarized by a cluster consensus (wdmATTwrCATawgaa) visible on the parental branch of the tree. By clicking on this consensus, the user can access to the cluster logo (Figure 3b-c). The lower subgroup includes 5 wider motifs matching the composite Sox/Oct (*SOCT*) motif.

The figures 3.b and 3.c show the aligned branch motifs and consensus of the Oct4 and SOCT motifs. This figure shows how the cluster consensus highlights the relevant positions of the grouped motifs. Also it is noticeable that the octamer canonical sequence of OCT4 (5'-ATGCAAAT-3') are the positions with higher information content in the Oct4 cluster logo. In regard to the SOCT motif, the canonical sequence of Sox (5'-CATTGTA-3') is clearly represented in the logo, except for the 'C' at the first position of the canonical sequence. Some of the other motifs matches with TFs regulating development, for example MEF2C in the cluster 3, Sp1 in cluster 5.



```
bwTwTKYWwATgh:           --ATTTGCATA--------
srCTGCAGys:              ----CTGCAG---------
ymTGCGCAkr:              ---TGCGCA----------
ywTTGTtATkcwwATGcww:     -CATWWNCATNACAA----
wwgCATTTrmAtawCAAwr:     GCATTTNNAWNNCAA----
ryATGCGCATry:            --ATGCGCAT---------
ttcwTWTGCAwATgh:         --ATWTGCATATG------
wtAAAAAwwa:              --WTTTTTA----------
yrTwTGCATATGCrCATry:     --ATGYGCATATGCANA--
wwATGYwRATkw:            --ATTTGCAT---------
wwATGywAAtk:             ---TTTGCAT---------
rmATrAmAATrr:            --ATTGTYAT---------
yyATTGTTATgywaATGww:     -CATNWRCATAACAAT---
ywCATTwrcATawCAATrv:     -CATTWNCATNWCAAT---
crTATGCGCATAyg:          -TATGCGCATA--------
yywTTGwtATGCWAATGwr:     -CATTTGCATAACAAT---
yryATGyGCATATGCAwATr:    --ATNTGCATATGCRCATA
aAAAAAAAAAa:             --TTTTTTTTTTT------
csCCGCcgCCGCcs:          ------GCGGNNGCGG---
cmCCCCCACCCCc:           ---CCCCCACCCC------
wgCATTwsAATAca:          -CATTWNAATA--------
```

**Figure 4. STAMP result with the 21 PSSMs discovered by *peak-motifs* in Oct4 ChIP-seq peaks.**

Although the comparison and alignment of TFBMs can be done independently by many tools, besides *matrix-clustering,* STAMP is the only tool that simultaneously produces the hierarchical clustering and alignment of TFBMs. However *matrix-clustering* presents significant advantages over existing solutions. (1) Display of motif trees either with aligned logos and string-based consensuses. (2) Segmentation of the input set of TFBM into separated clusters, rather than a motif tree. This approach is particularly an advantage because the motifs are not forced to be aligned in one global alignment. (3) Generation, at each level of the trees, of a merged motif (matrix and consensus) summarizing all descendent motifs rather than one general profile, which could be affected when distant motifs are aligned.

To show the differences among these tools I analyzed the same 21 motifs with STAMP, results are shown in Figure 4. STAMP grouped the same input motifs in 4 clusters, with the same subdivisions as *matrix-clustering*, except for the motif CTGCAG, which appears as a singleton in the matrix-clustering results. This occurs by the combination of metrics values used as threshold to create the clusters which allows *matrix-clustering*.

This study case shows that *matrix-clustering* identifies clusters of similar and redundant motifs depending on the threshold used, it display a friendly interplay to visualize the alignments which allow to recognize compund motifs as SOCT, it merge the matrices to produce a cluster logo and consensus which highlight the relevant positions of the children nodes on each branch of each tree.

### *Case study 2: negative control with randomized motifs*

In order to test the relevance of the clusters returned by *matrix-clustering* I submitted a set of randomized matrices, expecting that the program would not regroup them into clusters. I randomized the same 21 PSSMs from *peak-motifs* result in the Oct4 ChIP-seq peaks by permuting their columns. This approach presents the advantage of maintaining the number of sites, the residue frequencies and the information content of each matrix, but the biological context (the order of the relevant positions) is usually lost after this randomization. The same threshold values were used (Ncor >= 0.4,  cor >= 0.6) as in the study case 1.

Figure 5 shows that the 21 input motifs were split in 20 clusters. Only two motifs were grouped into one cluster (cluster 1). It must be noticed that both motifs are A-rich motifs which are not so informative and the permutations do not alter significantly the motifs. The 19 remaining motifs were not grouped and are displayed as single-leaf trees. The cluster consensus is only displayed where the motifs were clustered.

The number of cluster in the negative control can variate each time the set of matrices is permuted. To have an estimate of how many clusters can be found each time of randomization, I took the same 21 matrices and generated 100 sets of permutations. The threshold used is the same than the other examples: Ncor >= 0.4, cor >= 0.6. The distribution of the number of clusters is shown in figure 6. In most of the cases,  20 or 19 clusters were found, as in the Figure 5. In most cases, some motifs are clustered because they have poor complexity (A-rich motifs) and it is not surprising to find these motifs in the same cluster because they are made of essentially a repetition of the same column.

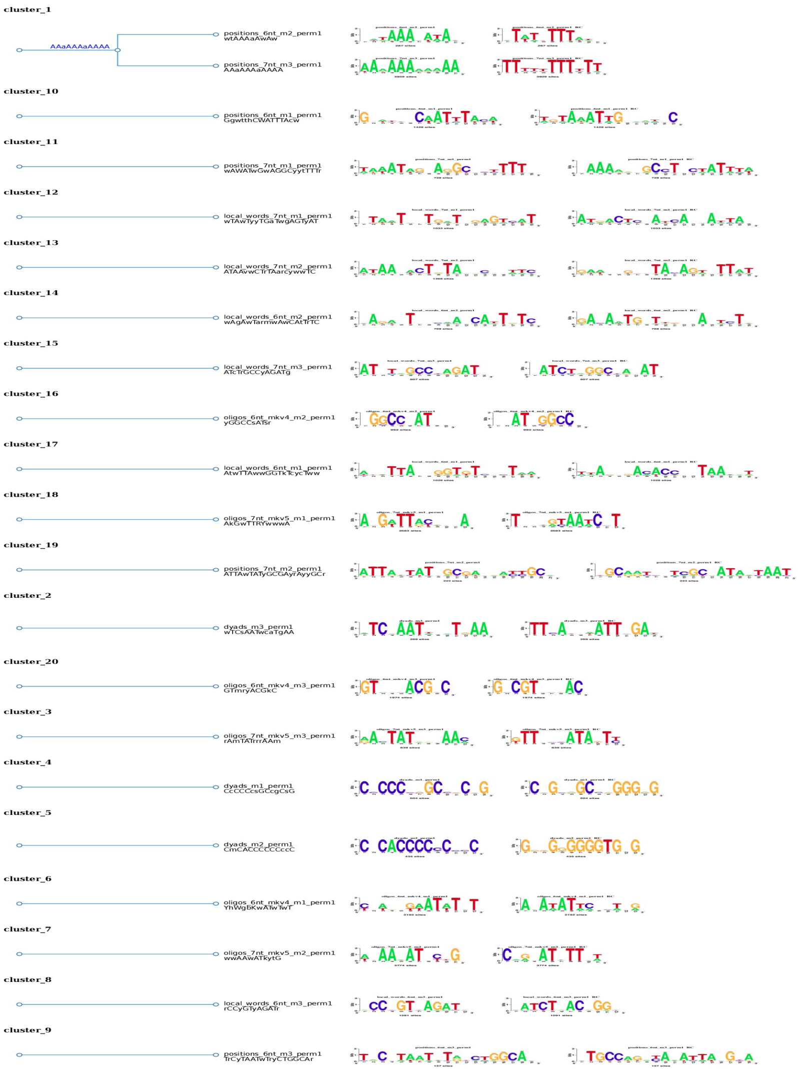


**Figure 5. Negative control: clustering of permuted Oct4 PSSMs.** Logo forest showing the clusters found among the permuted matrices.

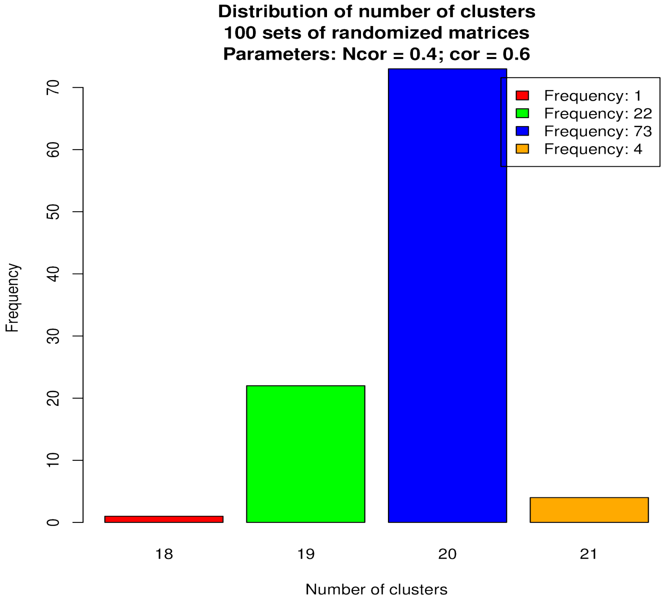
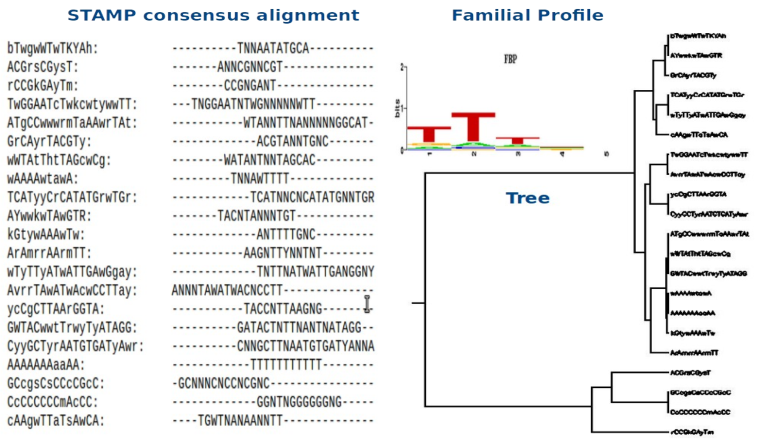


**Figure 6. Distribution of the number of clusters.** The barplot shows the distribution of the number of clusters found in 100 trials using the permuted versions of the 21 Oct4 PSSMs.

This test shows the importance of the order of the conserved residues in the  motifs as a factor which affects considerably the similarity. Indeed, these permuted matrices have the same information content as the real motif; the only difference is that the columns were sorted in a different way. This test also shows the capability of *matrix-clustering* to avoid clustering non-related motifs, and hence the motifs are not forced to be aligned. However as is discussed later, this capability depends on the user-chosen thresholds on matrix comparison metrics, and on the agglomeration rule selected for hierarchical clustering.

I also analyzed the same randomized motifs with STAMP, the results are shown in figure 7. The motifs were grouped in 7 clusters depicted in the tree, the resulting familial profile is not so informative and is reflecting the aligned positions of a few motifs. This test shows that *matrix-clustering* can effectively separate a set of unrelated motifs into clusters and it and highlights the importance of a threshold defined in a combination of metrics, which is not supported in SATMP.



```
bTwgwWTwTKYAh:          ----------TNNAATATGCA----------
ACGrsCGysT:             -------ANNCGNNCGT--------------
rCCGkGAyTm:             --------CCGNGANT---------------
TwGGAATcTwkcwtywwTT:    ---TNGGAATNTWGNNNNNWTT---------
ATgCCwwwrmTaAAwrTAt:    -----------WTANNTTNANNNNNGGCAT-
GrCAyrTACGTy:           -------------ACGTANNTGNC-------
wWTAtThtTAGcwCg:        --------WATANTNNTAGCAC---------
wAAAAwtawA:             ---------TNNAWTTTT-------------
TCATyyCrCATATGrwTGr:    ------------TCATNNCNCATATGNNTGR
AYwwkwTAwGTR:           -------TACNTANNNTGT------------
kGtywAAAwTw:            -------------ANTTTTGNC---------
ArAmrrAArmTT:           -----------AAGNTTYNNTNT--------
wTyTTyATwATTGAwGgay:    -------------TNTTNATWATTGANGGNY
AvrrTAwATwAcwCCTTay:    ANNNTAWATWACNCCTT--------------
ycCgCTTAArGGTA:         -----------TACCNTTAAGNG--------
GWTACwwtTrwyTyATAGG:    ----------GATACTNTTNANTNATAGG--
CyyGCTyrAATGTGATyAwr:   ----------CNNGCTTNAATGTGATYANNA
AAAAAAAaaAA:            ------------TTTTTTTTTTT--------
GCcgsCsCCcCGcC:         -GCNNNCNCCNCGNC----------------
CcCCCCCCmAcCC:          -------------GGNTNGGGGGGNG-----
cAAgwTTaTsAwCA:         ----TGWTNANAANNTT--------------
```

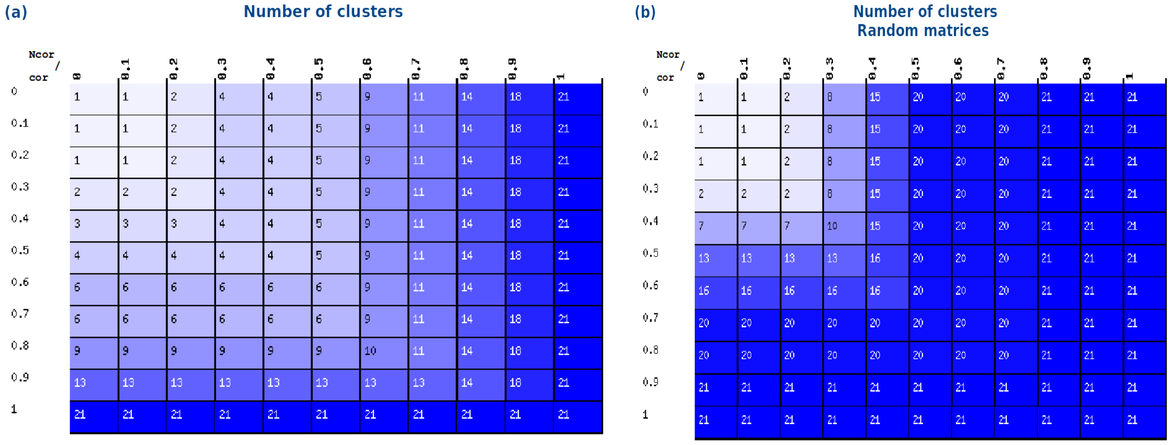
**Figure 7. STAMP result with permuted Oct4 PSSMs**.

## *Impact of threshold values*

To define the clusters, the user must specify some threshold value. However given that the tool *compare-matrices* allows to combine multiple metrics in the same analysis, it is possible to impose simultaneous thresholds on more than one metric. So, in the previous examples, I used a combination of Ncor >= 0.4 and cor >= 0.6. This approach allows users to vary threshold values to adapt the granularity of partitioning of the motif tree and of the associated motif alignments, in order to find a combination producing relevant groups.

Figure 8a shows a heatmap indicating the number of clusters found by *matrix-clustering* on the  21 motifs from *peak-motifs* result in the Oct4 ChIP-seq peaks, when threshold values for Ncor and cor vary from 0 to 1. At very low values most motifs are grouped in one cluster whilst with higher values the number of clusters tend to increase. I also applied the same procedure using column-permuted versions of the 21 motifs (Figure 8b). As expected, the number of clusters increases considerably  by comparison with the original matrices.

These data demonstrate how a combination of stringent parameters can alter the number of clusters identified by *matrix-clustering*.



(a) Number of clusters

| Ncor / cor | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 4 | 4 | 5 | 9 | 11 | 14 | 18 | 21 |
| 0.1 | 1 | 1 | 2 | 4 | 4 | 5 | 9 | 11 | 14 | 18 | 21 |
| 0.2 | 1 | 1 | 2 | 4 | 4 | 5 | 9 | 11 | 14 | 18 | 21 |
| 0.3 | 2 | 2 | 2 | 4 | 4 | 5 | 9 | 11 | 14 | 18 | 21 |
| 0.4 | 3 | 3 | 3 | 4 | 4 | 5 | 9 | 11 | 14 | 18 | 21 |
| 0.5 | 4 | 4 | 4 | 4 | 4 | 5 | 9 | 11 | 14 | 18 | 21 |
| 0.6 | 6 | 6 | 6 | 6 | 6 | 6 | 9 | 11 | 14 | 18 | 21 |
| 0.7 | 6 | 6 | 6 | 6 | 6 | 6 | 9 | 11 | 14 | 18 | 21 |
| 0.8 | 9 | 9 | 9 | 9 | 9 | 9 | 10 | 11 | 14 | 18 | 21 |
| 0.9 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 14 | 18 | 21 |
| 1 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 |

(b) Number of clusters
Random matrices

| Ncor / cor | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 8 | 15 | 20 | 20 | 20 | 21 | 21 | 21 |
| 0.1 | 1 | 1 | 2 | 8 | 15 | 20 | 20 | 20 | 21 | 21 | 21 |
| 0.2 | 1 | 1 | 2 | 8 | 15 | 20 | 20 | 20 | 21 | 21 | 21 |
| 0.3 | 2 | 2 | 2 | 8 | 15 | 20 | 20 | 20 | 21 | 21 | 21 |
| 0.4 | 7 | 7 | 7 | 10 | 15 | 20 | 20 | 20 | 21 | 21 | 21 |
| 0.5 | 13 | 13 | 13 | 13 | 16 | 20 | 20 | 20 | 21 | 21 | 21 |
| 0.6 | 16 | 16 | 16 | 16 | 16 | 20 | 20 | 20 | 21 | 21 | 21 |
| 0.7 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 21 | 21 | 21 |
| 0.8 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 21 | 21 | 21 |
| 0.9 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 |
| 1 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 |

**Figure 8. Impact of combination of threshold values on the number of clusters.** (a) Heatmap indicating the number of clusters found in the 21 motifs from Oct4 ChIP-seq experiment.. (b) Heatmap indicating the number of clusters in one permuted version of the same 21 motifs . The values of cor and Ncor variates from 0 to 1.

## ***Impact of clustering method***

In addition of the thresholds, another parameter that strongly impacts on the number and composition of the clusters is the agglomeration rule used to build the trees. This parameter affects not only  the number of clusters, but also the structure of the trees (order of motif incorporation in the progressive alignment), and hence the motifs repartition among the clusters. Figure 9 shows the differences of tree topologies and motif regroupment  in clusters, depending on the agglomeration rule: (a) single linkage, (b) average linkage, (c) complete linkage. Clusters are highlighted on the trees by the alternance of different colors.

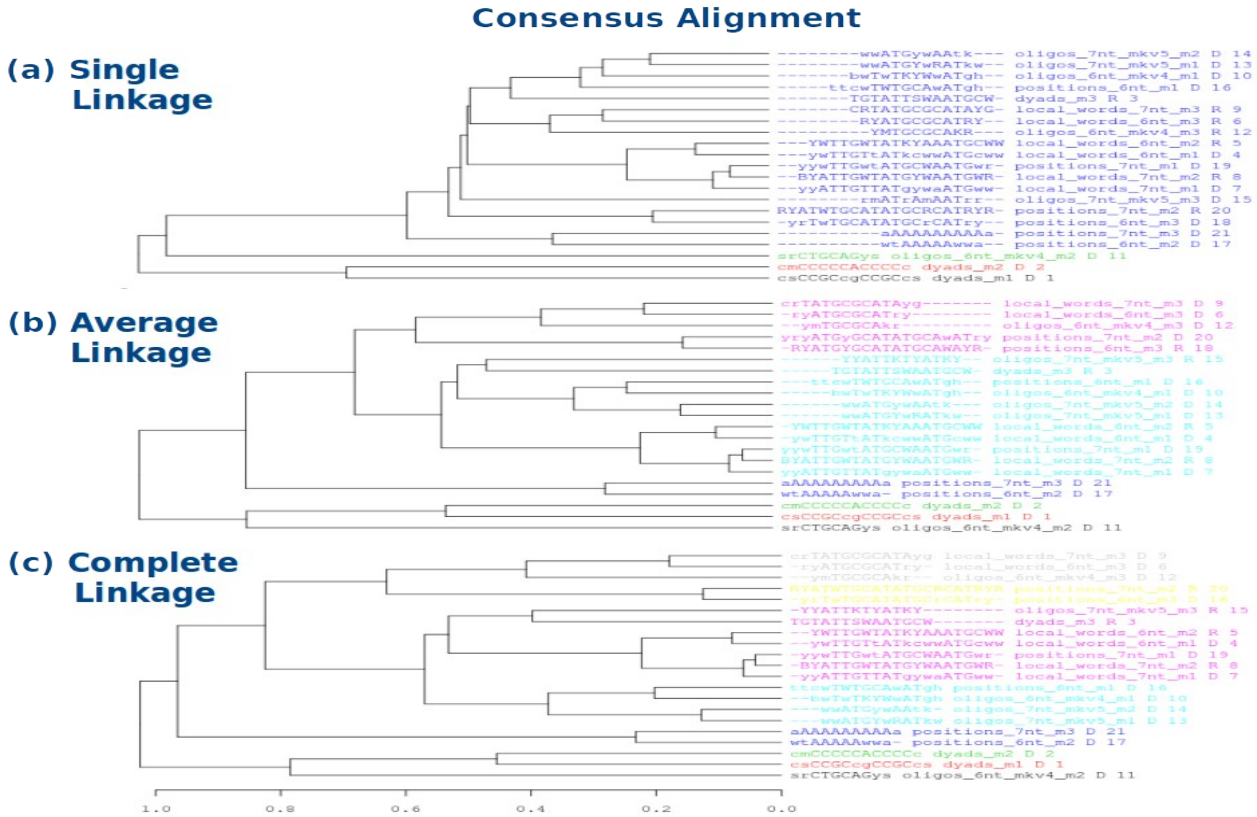

Consensus Alignment
(a) Single Linkage
---------wwATGywAAtk--- oligos_7nt_mkv5_m2 D 14
---------wwATGYwRATkw-- oligos_7nt_mkv5_m1 D 13
--------bwTwTKYWwATgh-- oligos_6nt_mkv4_m1 D 10
-----ttcwTWTGCAwATgh-- positions_6nt_m1 D 16
--------TGTATTSWAATGCW- dyads_m3 R 3
--------CRTATGCGCATAYG- local_words_7nt_m3 R 9
---------RYATGCGCATRY-- local_words_6nt_m3 R 6
----------YMTGCGCAKR--- oligos_6nt_mkv4_m3 R 12
---YWTTGWTATKYAAATGCWW local_words_6nt_m2 R 5
---ywTTGTtATkcwwATGcww local_words_6nt_m1 D 4
--yywTTGwtATGCWAATGwr- positions_7nt_m1 D 19
--BYATTGWTATGYWAATGWR- local_words_7nt_m2 R 8
--yyATTGTTATgywaATGww- local_words_7nt_m1 D 7
---------rmATrAmAATrr-- oligos_7nt_mkv5_m3 D 15
RYATWTGCATATGCRCATRYR- positions_7nt_m2 R 20
-yrTwTGCATATGCrCATry-- positions_6nt_m3 D 18
-----------aAAAAAAAAAa- positions_7nt_m3 D 21
-----------wtAAAAAwwa-- positions_6nt_m2 D 17
srCTGCAGys oligos_6nt_mkv4_m2 D 11
cmCCCCCACCCCc dyads_m2 D 2
csCCGCcgCCGCcs dyads_m1 D 1
(b) Average Linkage
crTATGCGCATAyg-------- local_words_7nt_m3 D 9
-ryATGCGCATry--------- local_words_6nt_m3 D 6
--ymTGCGCAkr---------- oligos_6nt_mkv4_m3 D 12
yryATGyGCATATGCAwATry positions_7nt_m2 D 20
-RYATGYGCATATGCAWAYR- positions_6nt_m3 R 18
------YYATTKTYATKY-- oligos_7nt_mkv5_m3 R 15
-----TGTATTSWAATGCW- dyads_m3 R 3
---ttcwTWTGCAwATgh-- positions_6nt_m1 D 16
-----bwTwTKYWwATgh-- oligos_6nt_mkv4_m1 D 10
------wwATGywAAtk--- oligos_7nt_mkv5_m2 D 14
------wwATGYwRATkw-- oligos_7nt_mkv5_m1 D 13
-YWTTGWTATKYAAATGCWW local_words_6nt_m2 R 5
-ywTTGTtATkcwwATGcww local_words_6nt_m1 D 4
yywTTGwtATGCWAATGwr- positions_7nt_m1 D 19
BYATTGWTATGYWAATGWR- local_words_7nt_m2 R 8
yyATTGTTATgywaATGww- local_words_7nt_m1 D 7
aAAAAAAAAAa positions_7nt_m3 D 21
wtAAAAAwwa- positions_6nt_m2 D 17
cmCCCCCACCCCc dyads_m2 D 2
csCCGCcgCCGCcs dyads_m1 D 1
srCTGCAGys oligos_6nt_mkv4_m2 D 11
(c) Complete Linkage
crTATGCGCATAyg local_words_7nt_m3 D 9
-ryATGCGCATry- local_words_6nt_m3 D 6
--ymTGCGCAkr-- oligos_6nt_mkv4_m3 D 12
RYATWTGCATATGCRCATRYR positions_7nt_m2 R 20
-yrTwTGCATATGCrCATry- positions_6nt_m3 D 18
-YYATTKTYATKY--------- oligos_7nt_mkv5_m3 R 15
TGTATTSWAATGCW-------- dyads_m3 R 3
--YWTTGWTATKYAAATGCWW local_words_6nt_m2 R 5
--ywTTGTtATkcwwATGcww local_words_6nt_m1 D 4
-yywTTGwtATGCWAATGwr- positions_7nt_m1 D 19
-BYATTGWTATGYWAATGWR- local_words_7nt_m2 R 8
-yyATTGTTATgywaATGww- local_words_7nt_m1 D 7
ttcwTWTGCAwATgh positions_6nt_m1 D 16
--bwTwTKYWwATgh oligos_6nt_mkv4_m1 D 10
---wwATGywAAtk- oligos_7nt_mkv5_m2 D 14
---wwATGYwRATkw oligos_7nt_mkv5_m1 D 13
aAAAAAAAAAa positions_7nt_m3 D 21
wtAAAAAwwa- positions_6nt_m2 D 17
cmCCCCCACCCCc dyads_m2 D 2
csCCGCcgCCGCcs dyads_m1 D 1
srCTGCAGys oligos_6nt_mkv4_m2 D 11
1.0
0.8
0.6
0.4
0.2
0.0

**Figure 9. Impact of agglomeration rule on the number of clusters.** Consensus trees and alignments of the 21 motifs discovered by *peak-motifs* in Oct4 ChIP-seq peaks. Each tree results from a different agglomeration rule for the hierarchical clustering. (a) Single linkage. (b) Complete linkage. (c) Average linkage. Separate clusters are created whenever one of the user-specified thresholds is not satisfied, cor >= 0.6 and Ncor >= 0.4.

### <u>*Case study 3: identification of motif families in the JASPAR database*</u>

Beyond the identification of redundant motifs in motif-discovery results, *matrix-clustering* can be used also to group motifs bound by TF assigned to the same protein family, because they share a common DNA binding domain. This study case shows how *matrix-clustering* can group motifs of the same TF families and how the logo alignment simplifies the interpretation of motifs networks. I analyzed separately two sections of the non-redundant “core” Jaspar[11] database, corresponding to insect and vertebrate TFs, respectively. For both cases, all the motifs of the collection where compared with each other, using RSAT *compare-matrices.* The resulting table of motif-to-motif similarities were converted to a motif-to-motif network (Figures 10a and 10c), where each motif is represented as a vertex, and the edges denote pairs of similar motif, with color and thickness reflecting the similarity scores. This representation allows a visualization of the groups, however this visualization does not provide a partition in the data into proper clusters.

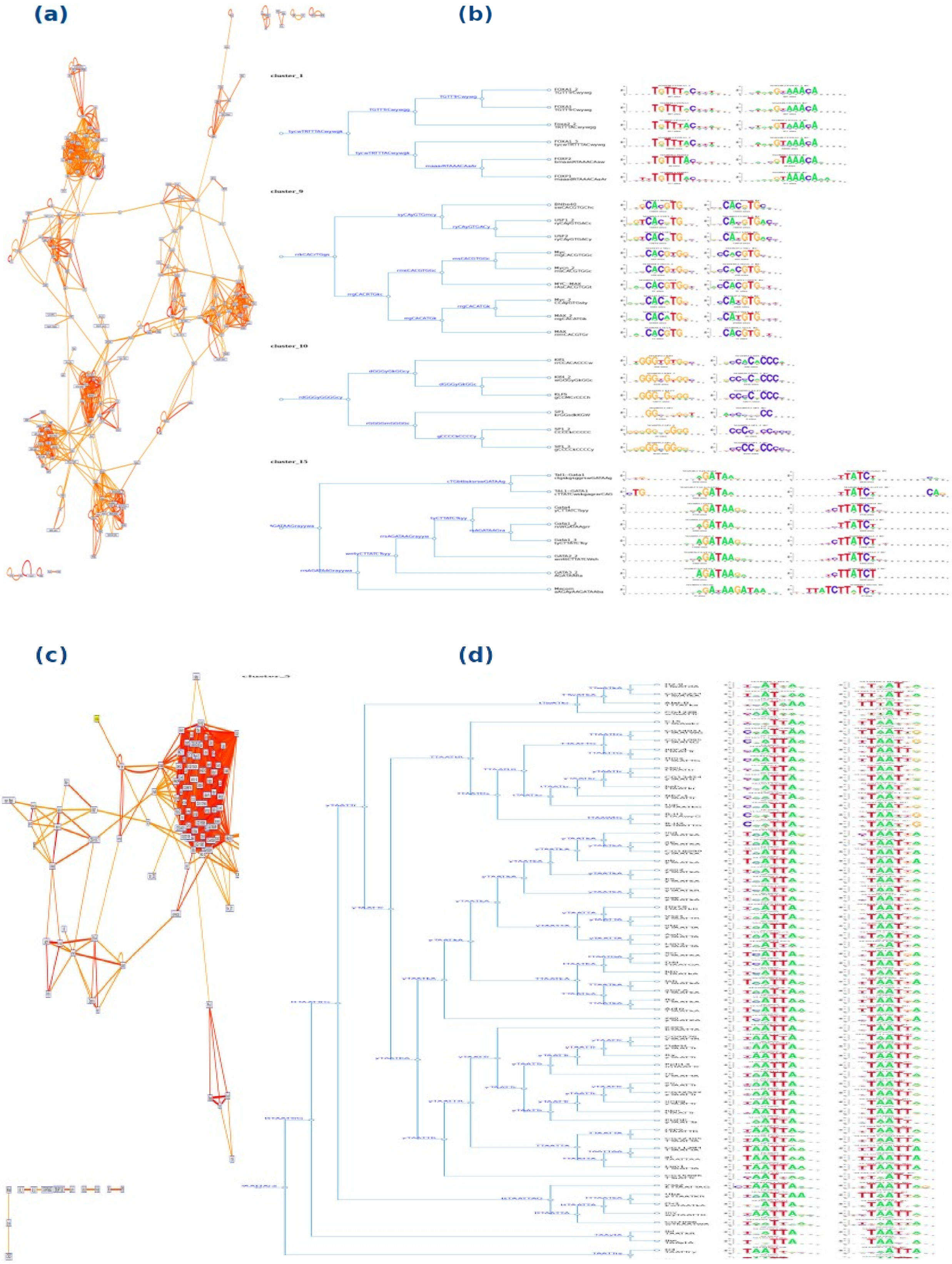


**Figure 10. Similarity networks extracted from the taxon-specific collections of matrices from the Jaspar database.** (a) Vertebrate TFBMs. (b) Insect TFBMs. (c) Fragment of the logo tree showing the typical Hox-motifs responsible for the highly connected subgroup.

I used *matrix-clustering* in order to study the similarities of the grouped motifs looking for cluster of motifs belonging to the same TF family.  Threshold values are Ncor >= 0.4 and cor >= 0.7, average-linkage as agglomeration rule. For this example I used the vertebrate core dataset. In the motif-to-motif network of these motifs (Figure 10a) at first glance we can observe several well-separated clusters, with a high intra-cluster connectivity, which correspond to well-known TF families, such as FOX, GATA, SOX, MYC, SP, KLE, etc. The *matrix-clustering* results brings additional insight to understand how the clusters that can be seen on the motif-to-motif network were segmented and aligned. Figure 10b shows a selection of clusters found in the analysis. On cluster 1, corresponding to the FOX family, we notice the high similarity between the motifs and how the relevant positions in the cluster-consensus are almost the same position in the logos. Cluster 9 regrouops the myc, max and usf motifs, which all belong to the Helix-Loop-Helix Leucine Zipper family. Cluster 10 presents a particular case where motifs bound by two unrelated TF families: (Klf and Sp, resp.) are grouped in the same cluster. Interestingly,are separated in two subgroups, one of them is the Krüppel-like factors (kpl) and the other is with the SP factors, both groups are members of the Sp/KLF family, both KLF and SP factors are characterized by zinc finger domains. However the capability of matrix-clustering to segregate motis respectively bound by Klf and SP factors suggest that tjeir respective domains have sufficiently diverged to confer them significant differenences in the DNA-binding specificity. Cluster 15 shows the GATA motifs, the upper subgroup has a composite motif TAL::GATA. We also observe a leaf with the Mecom TF, which is known to recognize tha same 'GATA' sequence as the canonical GATA factors, whereas heaving a distinct binding domain. This example shows the capability of *matrix-clustering* to group motifs belonging to the same families, and even to distinguish relevatn subgroups of TFs based on the evolutionary divergence of their binding domains.

To further investigate how matrix-clustering can help to understand the topology of a network I also analyzed the insects core dataset from JASPAR. The insect motif-to-motif network (Figure 10c) presents a very particular topology where most of the motifs are grouped in one big cluster. A fraction of the logo alignment is shown in figure 10d. The logo alignment shows a big strongly connected cluster. The logo tree indicates that this big quasi-clique of the network corresponds to a set of highly similar motifs (TAATTA) corresponding to Hoxs. In addition to help to explain the topology of this network, for these set of motifs, it is interesting to note that many PSSMs are grouped according to their TF families.

# CONCLUSIONS AND OUTLOOK

In this work I presented *matrix-clustering* which is a novel bioinformatic tool that addresses one of the main challenges in the field of analysis of cis-regulatory sequences: the clustering and alignment of TFBMs.

To improve the analysis, specially for large collections of motifs, the MCL approach can be integrated in *matrix-clustering,* this would allow use the motif similarities resulting from *compare-matrices* to partition the network before applying the hierarchical clustering. Indeed,MCL can treat very large graphs (thousands of nodes) within a few seconds, whereas the time required for hierarchical clustering increases quadratically with the number of motifs. MCL could thus be use as a first, rapid way to partition large motif sets (e.g. full databases), and each subgraph would subsequently be treated by hierarchical clustering.

The next goal will be to integrate this program within other programs of RSAT to complement the analysis of motifs, for example within *peak-motifs* to simplify the analysis of redundant motifs found on ChIP-seq peaks or within *motif-discovery* which is a phylogenetic footprint program where it could be useful group the motifs and study if they belong to certain taxa.

This tool present advantages over other existing solutions: (1) partitioning the input motif set into distinct clusters, (2) cluster-wise multiple alignment, (3) visual representations with either consensuses or logos, (4) capability to specify thresholds on more than one metric, (5) user-friendly interface to display the clusters and logo alignments. The study cases show many application which *matrix-clustering* can be use for. For example, grouping sets of redundant motifs resulting from motif discovery, identifying composite motifs, identifying groups of motifs belonging to the same family.

## ACKNOWLEDGMENTS

I do really want to thank to all the people behind this work and this experience to live in France. Thanks to Conacyt for the bourse! To Guigon family for the advices and french food. To Jonas, Amhed, Yuvia, Blanca, Eva, Violena, Steph, Otoniel and Lucia for discover Paris with them. To Nina for be patient, cheery and ready to listen and specially for always look at the eyes. To the Erasmus family at Luminy. To my professors and classmates. Thanks to Morganne Thomas and Dennis Thieffry for host me in their lab, for the discussions and the suggestions for this project. To my family and friends in Mexico for all their support along this trip. To Gustavo Ruiz (Monrroik): “Por los N-4 trapos, marmotas y bebés deliciosos que la casa nos ha invitado. Aquí se vale to' si tu te atleves”. Finally I would like to thank Jacques van Helden to teach me not just about science, but enthusiasm and passion for everything I do.

# BIBLIOGRAPHY

1. Ishihama A. (2009) **The Nucleoid: an Overview.** EcoSal—Escherichia coli and Salmonella: Cellular and Molecular Biology.

2. Vaquerizas J, Kummerfeld S, Teichmann S, Luscombe N. (2009) **A census of human transcription factors: function, expression and evolution.** Nat Rev Genetics.

3. Prabakaran P, Siebers J, Ahmad S, Gromiha M, Singarayan M, Sarai A. (2006) **Classification of Protein-DNA Complexes Based on Structural Descriptors.** Structure.

4. Contreras-Moreira B., Sancho J., Espinosa Angarica V. (2009) **Comparison of DNA binding across protein superfamilies.** Proteins.

5. Stewart A, Hannenhalli S, Plotkin J. (2012) **Why Transcription Factor Binding Sites Are Ten Nucleotides Long.** Genetics.

6. Chen X, Xu H, Yuan P, Fang F, Huck-Hui Ng. (2008) **Integration of external signaling pathways with the core transcriptional network in embryonic stem cells.** Cell.

7. Wasserman W, Sandelin A. (2004) **Applied bioinformatics for the identification of regulatory elements.** Nat Rev Genet.

8. Cornish-Bowden. (1985) **IUPAC-IUB symbols for nucleotide nomenclature.** Nucl. Acids Res.

9. Stormo D. (2000) **DNA binding sites: representation and discovery.** Bioinformatics.

10. Salgado H, Peralta-Gil M, Gama-Castro S, Santos-Zavaleta A, Collado-Vides J et al. (2012) **RegulonDB (version 8.0): Omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more.** Nucleic Acids Research.

11. Sandelin A, Alkema W, Engstrom P, Wasserman W and Lenhard B. (2004) **JASPAR: an open-access database for eukaryotic transcription factor binding profiles.** Nucleic Acids Res.

12. Matys V, Fricke E, Geffers R, Gossling E., Haubrock M, et al. (2003) **TRANSFAC: transcriptional regulation, from patterns to profiles.** Nucleic Acids Res.

13. Thomas-Chollier M, Defrance M, Medina-Rivera A, Sand O, Herrmann C, Thieffry D, van Helden J. (2011) **RSAT 2011: regulatory sequence analysis tools.** Nucleic Acids Res.

14. Bailey T, Bodén M, Buske FA, Frith M, Grant CE, Clementi L, Ren J, Li WW, Noble WS. (2009) **MEME SUITE: tools for motif discovery and searching.** Nucleic Acids Research.

15. Bailey T, Elkan C. (1994) **Fitting a mixture model by expectation maximization to discover motifs in biopolymers.** Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology.

16. van Helden J, André B, Collado-Vides J. (1998) **Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies.** J. Mol. Biol.

17. McLeay R, Bailey T. (2010) **Motif Enrichment Analysis: a unified framework and an evaluation on ChIP data.** BMC Bioinformatics.

18. Tompa M, Li N, Bailey T, Zhu Z et al. (2005) **Assessing Computational Tools for the Discovery of Transcription Factor Binding Sites.** Nature Biotechnology.

19. Janky R, van Helden J. (2008) **Evaluation of phylogenetic footprint discovery for predicting bacterial cis-regulatory elements and revealing their evolution.** BMC Bioinformatics.

20. Thomas-Chollier M, Herrmann C, Defrance M, Sand O, Thieffry D and van Helden J. (2011) **RSAT peak-motifs: motif analysis in full-size ChIP-seq datasets.** Nucleic Acids Research.

21. Pape U, Rahmann S, Vingron M. (2007) **Natural similarity measures between position frequency matrices with an application to clustering.** Bioinformatics.

22. Gupta S, Stamatoyannopoulos JA, Bailey T, Noble W. (2007) **Quantifying similarity between motifs.** Genome Biology.

23. Mahony S, Benos P. (2007) **STAMP: a web tool for exploring DNA-binding motif similarities.** Nucleic Acids Res.

24. Tanaka E, Bailey T,Grant C, Noble W, Keich U. (2011) **Improved similarity scores for comparing motifs.** Bioinformatics.

25. Machanick P, Bailey T. (2011) **MEME-ChIP: motif analysis of large DNA datasets.** Bioinformatics

26. Kulakovskiy I, Boeva VA, Favorov AV, Makeev VJ. (2010) **Deep and wide digging for binding motifs in ChIP-Seq data.** Bioinformatics.

27. Kuttippurathu L, Hsing M, Liu Y, Schmidt B, Maskell DL, Lee K, He A, Pu WT, Kong SW. (2011) **CompleteMOTIFs: DNA motif discovery platform for transcription factor binding experiments.** Bioinformatics.

28. van Heeringen SJ, Veenstra GJ. (2011) **GimmeMotifs: a de novo motif prediction pipeline for ChIP-sequencing experiments.** Bioinformatics.

29. Thomas-Chollier M, Darbo E, Herrmann C, Defrance M, Thieffry D, van Helden J. (2012) **A complete workflow for the analysis of full-size ChIP-seq (and similar) data sets using peak-motifs.** Nature Protocols.

30. Loh Y, Wu Q, Chew L, Huck-Hui Ng et al. (2006) **The Oct4 and Nanog transcription network regulates pluripotency in mouse embryonic stem cells.** Nature.