CE^3: Customizable and Easily Extensible Ensemble Tool for Motif Discovery

Karina Panucia Tillán[1], Mauro Leoncini[2,3], and Manuela Montangero[2,3]

[1] Dip. di Scienze e Metodi dell'Ingegneria
[2] Dip. di Scienze Fisiche, Informatiche e Matematiche
Univ. di Modena e Reggio Emilia, Italy
83672@studenti.unimore.it, {leoncini, manuela.montangero}@unimore.it
[3] Istituto di Informatica e Telematica, CNR - Pisa, Italy

Abstract. Ensemble methods (or simply *ensembles*) for motif discovery represent a relatively new approach to improve the accuracy of stand-alone motif finders. In particular, the accuracy of an ensemble is determined by the included finders and the strategy (*learning rule*) used to combine the results returned by the latter, making these choices crucial for the ensemble success. In this research we propose a general architecture for ensembles, called CE^3, which is meant to be extensible and customizable for what concerns *external tools* inclusion and *learning rule*. Using CE^3 the user will be able to "simulate" existing ensembles and possibly incorporate newly proposed tools (and learning functions) with the aim at improving the ensemble's prediction accuracy. Preliminary experiments performed with a prototype implementation of CE^3 led to interesting insights and a critical analysis of the potentials and limitations of currently available ensembles.

Keywords: ensemble methods, motif discovery, software architecture

1 Introduction

The discovery of Transcription Factor Binding Sites (TFBSs), i.e., functional DNA sequences involved in gene expression, is an important and challenging problem in molecular biology. As the experimental protocols available for TFBS discovery are lengthy and costly, the problem has been tackled also from a computational perspective. Mathematical models of TFBSs have been proposed [20, 7], often termed *motifs*, and many algorithms designed and implemented in the last thirty years (see, e.g., [4, 21, 18] and [6] for further references).

Despite such impressive efforts, the prediction accuracy remains low. A relatively recent assessment of thirteen popular algorithms performed by Tompa et al. [23] has made it clear that no single method performs well (i.e., gives accurate results) on different datasets, and that it is by no means easy to characterize the inputs for which a method may give good performances.

In relatively recent times, a new approach has been pursued with the aim of overcoming the limitations of existing motif discovery algorithms (here also

termed *finders*). This is based on the idea that accurately combining the results returned by different finders can lead to better TFBSs predictions than using each finder alone. The tools following this paradigm are known as *ensemble methods* (or simply *ensembles*) [5, 9, 10, 26], or also meta-predictors [27].

A popular reasoning that supports the design of ensembles is a more or less sophisticated *voting argument*. The idea is that the likelihood of a DNA stretch being a functional site grows with the number of different motif discovery algorithms that report that stretch among their findings. The actual procedures adopted to "combine" the finders' results, often referred to as the *learning rules*, may vary a lot across different ensembles. Together with the choice of the actual finders used (typically third-party, external software tools), the learning rule is the feature that mostly affects the performance of an ensemble.

All the above cited studies that propose ensemble methods also report the results of a number of experiments performed of benchmark data. Indeed, the results seem to support the idea that, even putting together low performance finders, the overall accuracy of an ensemble can be cast to an acceptable level, well above those of the single finders.

Our thesis, supported by a number of experiments we have performed, is that the conclusion reported in the Tompa et al. paper can be extended to ensembles as well, namely that on different datasets the observed performances of an ensemble can vary a lot and that no clear indication has emerged yet so as to characterize the inputs that are "good" for a particular ensemble. At the very least, the observed performance of an ensemble cannot but strictly depend on the finders it is based on, with different "blends" likely leading to very different results. Also, since a positive correlation must exist among the accuracies of the finders used by an ensemble and that of the ensemble itself, a clever design should allow for the latter to include newly developed accurate tools.

In order to both "prove" our thesis and understand the actual strength of ensembles for de-novo motif discovery, we propose here a general ensemble architecture, called CE^3, which is *customizable* and *extensible* with respect the two key features mentioned above. The ultimate design goal is to make CE^3 able to simulate any available ensemble, also giving end users the possibility to create their own tool quite easily through the choice of specific finders (and/or the addition of new ones) and learning rules. Actually, the inclusion of a new finder in CE^3, though not completely automatic, is already quite an easy task in present state of development which do not require any programming skill. Currently, system configuration is done via terminal interaction, i.e., through a question answering procedure guided by the system itself; the mature CE^3 will be instead customizable through a Web interface. CE^3 is written in Python and is available from the authors upon request.

The rest of this paper is organized as follows. In Section 2 we give a description of general ensembles and a sketch of CE^3 architecture; in Section 3 we present and analyze the results obtained from the experiments performed with various configurations of CE^3; finally, in Section 4 we offer some concluding remarks and discuss future work scheduled on CE^3.

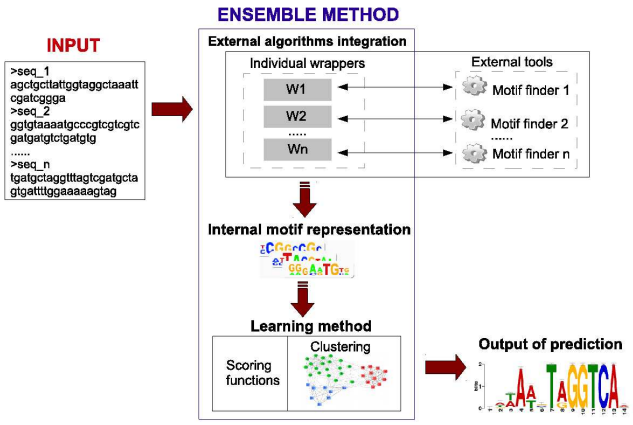

Fig. 1. General architecture of an ensemble.

2 Ensemble Architecture

Ensembles for *Motif Discovery Problem* (*MDP*) orchestrate the execution of many *de-novo* finders, each returning a set of motifs potentially describing biologically active sites, that are further analyzed to improve the accuracy of predictions. The general structure of such systems is depicted in Figure 1; there are four main components, with the first and the third being the crucial ones:

- *External algorithms integration module*: ensembles integrate possibly many different (third-party) *de-novo* finders.
- *Internal motif representation*: motifs returned by the finders are represented uniformly using appropriate data structures and handling software.
- *Learning rule* (or *function*): one or more techniques are used to discover the most promising motifs among all those predicted by single finders.
- *Output module*: prediction is returned to the user in one of the commonly adopted "external" motif representations (e.g., weight matrices and text logos), possibly with the explicit site lists.

Careful combinations of different finders may provide substantial improvements in motif predictions, as reported in the cited literature. However, ensembles implemented so far are characterized by a fixed, and to some extent arbitrary set of finders (say, the "best" ones available at design time). Some ensembles, such as Motif Voter [26], give the user the flexibility to choose a particular set of finders, but still from an immutable superset. Of course, extensions are always possible, but this requires knowledge of the ensemble internals and programming skills. In fact, one has to write at least the code to interact with the new method, *i.e.*, to wrap its execution, catch and parse the returned results.

The choice of the learning rules influences the output quality as well, affecting the prediction of relevant motifs. Currently available ensembles are characterized

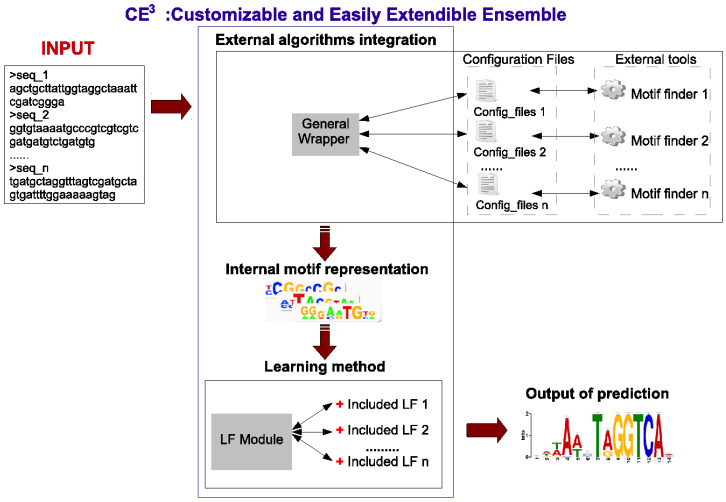

Fig. 2. CE^3 architecture: the integration of external algorithms depends on one general wrapper and many configuration files. Distinct learning functions (LF) may be included and executed in the ensemble by means of the Learning Function Module.

by a specific rules devised by the respective designers and hence changes are difficult to implement. Moreover, even greater efforts are requested if one wishes to add completely new learning rules to the ensemble.

2.1 Architecture of CE^3

CE^3's key features of easy customization and functionality extension depend on a conceptually simple modification of the generic ensemble's structure (Figure 2). For lack of space, in this paper we will concentrate only on external algorithms inclusion and learning rules, omitting all the technical details. A more comprehensive description can be found in [17].

Motif finder extensibility. There is a unique wrapper that specializes to the various external finders thanks to the finder descriptions available in XML files. To qualify for insertion in CE^3, a tool must run as a command line utility under Unix/Linux operating systems. This is the strongest constraint, but fortunately one of minimum negative impact in light of the intended use of CE^3 (which is not commercial); indeed, many currently available motif finders do satisfy such requirement. At present, CE^3 includes ten motif finders, namely: Aglam [12], AlignAce [11], BioProspector [13], MotifSampler [21], MEME [4], RSAT [22], MDscan [14], Weeder [18], Space [25] and Improbizer [3].

Learning rules customization. A module in CE^3 handles the available learning rules of the ensemble and the insertion of new ones. Each rule (i.e., its code) is stored in a separate folder that contains a standard Python interface (customized from a common template) and an XML configuration file used by the module to recognize and include the function in the ensemble.

Here however we observe that the addition of an existing learning rule (i.e., one adopted in some other ensemble) is not as easy as the integration of a whole new finder. In fact, it is by no means a trivial task to "isolate" the learning function module from the rest of the software, even when the source code is available. This essentially depends on the code quality and its structure. Perhaps the best (but time consuming) solution amounts to re-implementing the most promising existing learning rules reported in the published papers and/or the software documentation.

The inclusion of existing learning rules makes CE^3 able to simulate existing ensembles not only as originally designed, but also on an extended set of finders not previously included in the ensemble. We exploit this feature in our preliminary experiments, for what concerns the ensemble MotifVoter, the first ensemble tool for which we were able to retrieve properly working code. Currently, CE^3 includes a re-implementation of MotifVoter learning function [26], and a variant that seems to give even better results.

Moreover, CE^3 includes a learning rule based on motif clustering. More precisely, CE^3 selects the motif(s) to be returned according to the following three steps procedure in which, at each step, a few options are available to the user:

1. Compute pairwise motif similarities, either using normalized correlation of PWMs [22], or "degree" of sites overlapping using to the following formula:
$$I_{\mathcal{S}}(M_1, M_2) = \frac{|N_1(\mathcal{S}) \cap N_2(\mathcal{S})|}{\min\{|N_1(\mathcal{S})|, |N_2(\mathcal{S})|\}} \quad (1)$$
where M_1 and M_2 are the motifs being compared and $N_1(\mathcal{S})$ and $N_2(\mathcal{S})$ denote the sets of nucleotides in the input sequence set $\mathcal{S}$ matching M_1 and M_2, respectively. According to (1), M_1 and M_2 are regarded as very similar when the sites of one include those of the others. Note that $0 \leq I_{\mathcal{S}}(M_1, M_2) \leq 1$ and that $I_{\mathcal{S}}(M_1, M_2) = 1$ if $M_1 \subseteq M_2$ or $M_2 \subseteq M_1$; *i.e.*, M_1 and M_2 are highly similar when the sites of one include those of the others.
2. Compute motif clusters using either a simple single-linkage algorithm or single-linkage followed by the detection of dense cluster cores.
3. Discard the clusters that do not include motifs determined by at least two different finders, and rank the remaining clusters according to one of a number of available criteria.

The ensemble output is a set of putative motifs, given as a collection of binding sites or of PWMs. Optionally, CE^3 also provides output statistics.

To conclude this section, we observe that CE^3 is characterized by a great number of "parameters" that can be set by the user to tune the program behavior (e.g., see the just outlined procedure for combining the finders' results). This is not as advisable a circumstance for a bioinformatic application software. In particular, a biologist sees the need for parameter setting as a weakness of the tools. We do concur with this viewpoint. Indeed, many options available in CE^3 will be hidden to the end user. They primarily serve the goal of understanding the potentials of ensembles and possibly devise favorable (automatic) parameter settings as a function of the dataset at hand.

3 Discussion

We performed a number of experiments by running CE^3 under many different configurations, by varying both the set (number and composition) of the underlying finders and the various learning rule options. We run CE^3 on four different and widely adopted benchmark datasets, described in details in [23, 19, 1, 16]. Finally, we compared the results obtained using all the main statistics commonly used at nucleotide level (see again [23]). Figure 3 and, respectively, Figure 4 report a selection of results concerning the Tompa [23] and, respectively, Singh [16] datasets: we run the single finding tools using CE^3 default parameters and without performing any pre- or post-processing (contrary to what was allowed in the Tompa et al.'s assessment, which explains why some results are not consistent with those reported in [23]). We compare these results with those achieved by CE^3, MotifVoter and SCOPE [5], under different configurations (when possible) defined by varying the set of finders and the learning function, as stated in the figures captions.

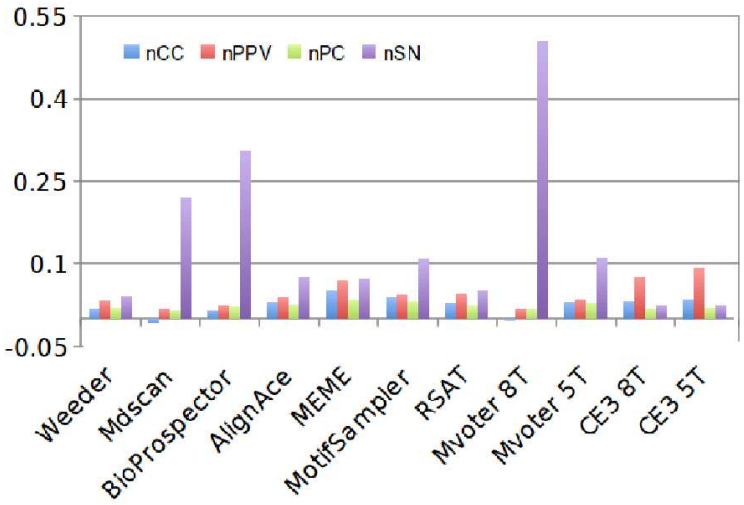

Fig. 3. Statistics for Tompa dataset [23]. "Mvoter"' means that CE^3 has been run using the re-implementation of the MotifVoter learning function, "8T" with all the eight finders whose statistics are reported in the histogram, "5T" with a selection of five (Meme, Alignace, MotifSampler, Weeder, RSAT). CE^3 has been run with the learning function based on clustering.

While we leave to the full paper a systematic presentation of the comprehensive amount of data obtained, here we offer a synthetic compendium of the crude facts that emerged from the experiments, followed by some reflections on what they suggest for future research directions in this field.

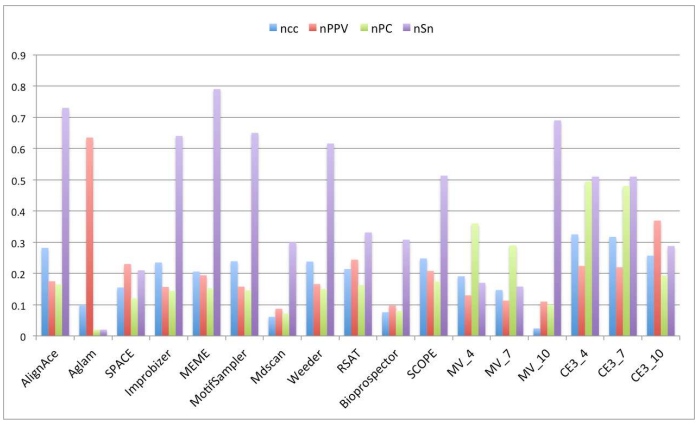

Fig. 4. Statistics for Singh dataset [16]. We compared results given by the 10 single finders and the following ensembles: SCOPE [5]; MV_4 (resp. MV_7) , the original MotifVoter run with the selection of 4 (resp. 7) finders giving the best results (namely MEME, MDscan, Bioprospector and Weeder (resp. SPACE, AligAce, Improbizer, Weeder, MEME, MDscan and MotifSampler)); MV_10, our MotifVoter re-implementation, using all 10 finders; CE^3 with the best selections of 4 (resp. 7) finders (namely Aglam, MEME, AlignAce and MotifSampler (resp. Aglam, Bioprospector, MEME, MDscan, AligAce, MotifSampler and RSAT)) and all 10 finders, using the variant of the MotifVoter learning function.

- The main (negative) finding is that no single configuration of CE^3 returned "decent" results across all the tested datasets.
- Apparently, no group of properties of the dataset seems able to predict the quality of the results produced by a given configuration, not even when considering properties that can be hardly known in advance to a de-novo motif discovery software (say, average site length or degree of conservation).
- Increasing the number of tools is almost never a winning strategy. Devising three or four good component finders seems the right way to go.
- The most dense motif clusters only rarely help locate the functional sites.
- Some tools very often appeared in the best performing set. Not surprisingly (see [23]), Weeder is the most frequent one.
- Having more than one tool with similar optimization criteria (e.g., Gibbs sampling) leads to very bad results more often than not. Instead, the voting criterion seems more suitable to a blend of finders representative of all the few good algorithmic strategies developed so far.

Figure 5 gives at least a partial explanation of the difficulties that can be encountered in the design of an ensemble. The plot refers to a particular input sequence of a particular dataset, but is definitely representative of the whole state of affairs. We ran CE^3 with eights finders (the same as in Figure 3) and, for each nucleotide position along the sequence, we reported the number of tools that included that position in their predictions. Noticeably, each of the 1500

positions is "voted" by at least one tool. How clever the learning rule must be to find the needle of the functional sites (highlighted by the tall rectangles) in the haystack of the predicted ones should appear evident to all.

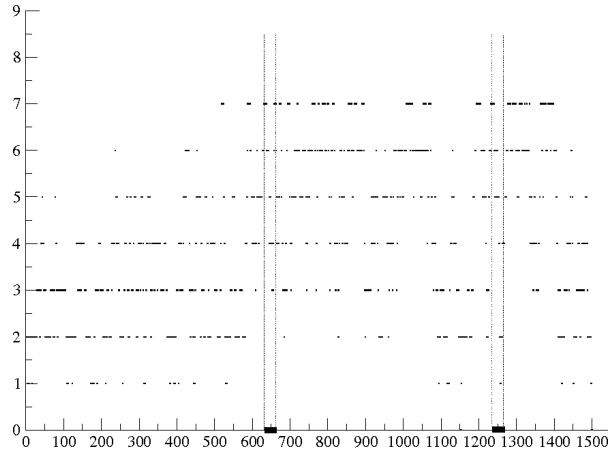

Fig. 5. Nucleotide level tool prediction on one of the datasets included in the benchmark in [23] (dm01) in one of the experiments depicted in Figure 3: the number of predicting tools is plotted against nucleotide position along one of the input sequences. Vertical dotted lines highlight the actual motif positions.

The phenomenon outlined in Figure 5 suggests that using a large number of finders likely decreases the ratio between the number of true positives and the total number of predicted nucleotides (i.e., the Positive Predicted Value, or PPV [23], of the set of finders as a whole). In other words, using a terminology from signal processing, this amounts to decreasing the signal-noise ratio given in input to the ensemble, which makes the task of the latter much more difficult. This observation is well supported by our experimental results.

On the other hand, in our experiments we often observed that "peaks" in the signals of the finder predictions (as in Figure 5) correspond also to the experimentally validated sites. In our opinion this clearly indicated that the idea behind ensemble construction is worth being pursued. The key design point is thus that of keeping the PPV of the set of finders at a reasonably high value (and Figures 3 and 4 suggest that, in this respect, CE^3 is working in the right direction). To achieve this goal, a fundamental ingredient is the number and quality of the component finders. In this respect, a good ensemble must be open to the inclusion of newly developed accurate finders, which is a major CE^3 design feature.

4 Further Work

Ongoing work on CE^3 is following three different directions. At present time, system configuration is done via terminal interaction, *i.e.*, through a question answering procedure guided by the system itself; we are working to make CE^3 available to "real" end-users as a Web service. In parallel, we are retrieving code (apparently not available on-line) to enhance CE^3 and to make comparison with other ensemble tools (*e.g.*, MProfiler [2]). In particular, we are interested in recently proposed finders (*e.g.*, Gimsan [15]) and in the implementation of new PWMs similarity measures (*e.g.*, FISim [8]). With respect to ensemble comparison, we will have to evaluate if it pertinent to compare CE^3 with existing ensemble expressly designed for ChIP-seq experiments (*e.g.*, GimmeMotifs [24]). Even if these may sometimes work also without ChIP-Seq input data, comparison might not be fair, as those ensemble were designed to take advantage from extra information that we will not provide in our experiments. Last but not least, probably the harder task, we are still investigating the best configurations (composing finders and learning rules) that will possibly guarantee accurate results across different datasets.

References

1. `http://bioinformatics.psb.ugent.be/webtools/MotifSuite/benchmarktest.php`.
2. D. Altarawy, M. A. Ismail, and S. M. Ghanem. MProfiler: A profile-based method for dna motif discovery. In *Proc. PRIB 2009*, volume 5780 of *LNCS*, pages 13–23, 2009.
3. W. Ao, W. Gaudet, J. NS Kent, et al. Environmentally induced forgut remodelling by pha-4/foxa and daf-12/nhr. *Science*, 305(5691):1743–1746, 2001.
4. T. L. Bailey and C. Elkan. The value of prior knowledge in discovering motifs with meme. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 21–29, Menlo Park, CA, 1995. AAAI Press.
5. A. Chakravarty, J. M. Carlson, R. S. Khetani, and R. H. Gross. A novel ensemble learning method for de novo computational identification of dna binding sites. *BMC Bioinformatics*, 2007.
6. M. K. Das and H. K. Dai. A survey of dna motif finding algorithms. *BMC Bioinformatics*, 8, 2007.
7. P. D'haeseleer. What are dna sequence motifs? *Nature Biotechnology*, 24:423–425, 2006.
8. F. Garcia, F. J. Lopez, C. Cano, and B. Armando. FISim: A new similarity measure between transcription factor binding sites based on the fuzzy integral. *BMC Bioinformatics*, 10, 2009.
9. J. Hu, Y. D. Yang, and D. Kihara. Emd: an ensemble algorithm for discovering regulatory motifs in dna sequences. *BMC Bioinformatics*, 2006.
10. B. R. Huber and M. L. Bulyk. Meta-analysis discovery of tissue-specific dna sequence motifs from mammalian gene expression data. *BMC Bioinformatics*, 2006.
11. J. D. Hughes et al. Computational identification of cis-regulatory elements associated with groups of functionally related genes in saccharomyces cerevisiae. *J. Mol. Biol.*, (296):1205–1214, 2000.

12. N. Kim, K. Tharakaraman, L. Mario-Ramrez, , and J. Spouge. Finding sequence motifs with bayesian models incorporating positional information: an application to transcription factor binding sites. *BMC Bioinformatics*, 9, 2008.
13. X. Liu, D. L. Brutlag, and J. S. Liu. Bioprospector: discovering conserved dna motifs in upstream regulatory regions of co-expressed genes. pages 127–138, 2001.
14. 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.*, 8(20), 2002.
15. P. Ng and U. Keich. Gimsan: a gibbs motif finder with significance analysis. *Bioinformatics*, 24(19):2256–2257, 2008.
16. R. Osada, E. Zaslavsky, and M. Singh. Comparative analysis of methods for representing and searching for transcription factor binding sites. *Bioinformatics*, 20:3516–3525, 2004.
17. K. Panucia Tillán, M. Leoncini, and M. Montangero. CE^3: Customizable and easily extensible ensemble tool for motif discovery. Technical Report TR-16-2012, Istituto di Informatica e Telematica - CNR, Pisa, Italy, 2012.
18. G. Pavesi, G. Mauri, and G. Pesole. An algorithm for finding signals of unknown length in dna sequences. *Bioinformatics*, (17):207–214, 2001.
19. G. Sandve, O. Abul, V. Walseng, and F. Drablos. Improved benchmarks for computational motif discovery. *BMC Bioinformatics*, 8(1):193, 2007.
20. G. D. Stormo. Dna binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, 2000.
21. G. Thijs, K. Marchal, M. Lescot, S. Rombauts, B. D. Moor, P. Rouze, and Y. Moreau. A gibbs sampling method to detect over-represented motifs in the upstream regions of co-expressed genes. *Computational Biology*, 9(2):447–464, 2002.
22. M. Thomas-Chollier, M. Defrance, A. Medina-Rivera, O. Sand, C. Herrmann, D. Thieffry, and J. van Helden. Rsat 2011: regulatory sequence analysis tools. *Nucleic Acids Research*, 39(suppl 2):W86–W91, 2011.
23. M. Tompa et al. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, (23):137–144, 2005.
24. S. J. van Heeringen and G. J. C. Veenstra. GimmeMotifs: a de novo motif prediction pipeline for chip-sequencing experiments. *Bioinformatics*, 27(2):270–271, 2011.
25. E. Wijaya, K. Rajaraman, S. Yiu, and W. Sung. Detection of generic spaced motifs using submotif pattern mining. *Bioinformatics*, 23:1476–1485, 2007.
26. E. Wijaya, S. M. Yiu, N. T. Son, R. Kanagasabai, and W. K. Sung. Motifvoter: a novel ensemble method for fine-grained integration of generic motif finders. *Bioinformatics*, 24(20):2288–2295, 2008.
27. F. Zambelli, G. Pesole, and G. Pavesi. Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. *Brief. in Bioinf.*, 2012.