CMStalker: a combinatorial tool for composite motif discovery

Mauro Leoncini[1,2], Manuela Montangero[1,2], Marco Pellegrini[2], and Karina Panucia Tillan[3]

[1]Dept. of Physics, Computer Science, and Mathematics, Univ. of Modena and Reggio Emilia, Italy. Email: name.surname@unimore.it
[2]IIT-CNR, Pisa, Italy. Email:marco.pellegrini@iit.cnr.it
[3]Dept. of Science and Methods for Engineering, Univ. of Modena and Reggio Emilia, Italy. Email: karina.panuciatillan@unimore.it

Abstract*—Controlling the differential expression of many thousands different genes at any given time is a fundamental task of metazoan organisms and this complex orchestration is controlled by the so-called *regulatory genome* encoding complex regulatory networks: several Transcription Factors bind to precise DNA regions, so to perform in a cooperative manner a specific regulation task for nearby genes. The *in silico* prediction of these binding sites is still an open problem, notwithstanding continuous progress and activity in the last two decades. In this paper we describe a new efficient combinatorial approach to the problem of detecting sets of cooperating binding sites in promoter sequences, given in input a database of Transcription Factor Binding Sites encoded as Position Weight Matrices. We present CMStalker, a software tool for composite motif discovery which embodies a new approach that combines a constraint satisfaction formulation with a parameter relaxation technique to explore efficiently the space of possible solutions. Extensive experiments with twelve data sets and eleven state-of-the-art tools are reported, showing an average value of the correlation coefficient of 0.54 (against a value 0.41 of the closest competitor). This improvements in output quality due to CMStalker is statistically significant.

Index terms: Algorithms, Biology and genetics

I. Introduction

Biological Motivation. *Transcription Factors* (*TF*) are proteins that bind to short specific stretches of DNA, called *TFBS - Transcription Factor Binding Sites*, usually in the proximity of genes and participate in regulating the expression of those genes [10]. The discovery of truly functional TFBSs is an important step in order to elucidate gene regulatory networks; this is witnessed by more than a hundred algorithms that have been proposed over the last two decades for the prediction "in silico" of *single* TFBS (see [49] and the many references contained therein). However, especially in eukaryotes, gene regulation involves a cohort of cooperating TFs, which typically have binding sites located in a short span within the genes' promoter, as well as enhancer and silencer, regions. The combinatorial nature of this cooperation is exploited by a number of other algorithms for the prediction of corresponding TFBS clusters, a task that is often termed as *composite motif* (CM) (or *pattern*) discovery in the literature (see [12] for one of the earliest contributions using the term "composite pattern"). Regions containing cooperating TFBS are also called *Cis-regulatory modules* (CRM for short).

Problem formalization. In this paper we address a well-studied variant of the composite motif discovery problem that can be informally stated as follows: given a set of DNA sequences, typically taken from the promoter regions of co-regulated genes, and given descriptions of DNA binding affinities (aka *simple motifs*) for allegedly cooperating TFs, predict the location and composition of sites bound by (subsets of) those TFs (*composite motifs*).

Models for Composite Motifs. Composite motifs are defined by three main features: the type of the component TF, the order and orientation in which the the TFBS appear in the gene's upstream sequence, and the relative mutual distances of the TFBS's. Some models impose a stringent rule on the order/orientation aspects (e.g. [54]) while other (see. [52]) do not impose a precise ordering. Indeed the degree to which biologically relevant CRMs are structured or unstructured is still largely unknown [28] [38], and it is conjectured that the TFBS order impact more on the fine working of the CRM, rather than being a signature of CRM presence or absence [9]. In our setting we use a model in which order is not stressed, and only TF composition and the relative TFBS distance are highlighted, though these two factors play different roles.

Proposed novel algorithm overview. We describe here *CMStalker*, a composite motif discovery tool whose computational core is a combinatorial search algorithm that explores the space state of possible solutions by progressively relaxing some (internal) parameters. Our algorithm can be seen as a very specialized form of *constraint satisfaction* engine with a specific strategy used to explore the configuration space of solutions determined by the two main parameters (quorum and window size). Our algorithm uses the combinatorial fingerprint of the composite motif in the solution-generation phase, and a distance uniformity constraints in a subsequent solution-filtering phase, but without any hard threshold. CMStalker also uses a statistical filtering criterion, based on the approximation of the p-values of recurrent groups of potential TF binding sites (see Section III), using the same approach adopted in [40] for modules. This strategy can be appreciated in contrast to, say, CPMmodule that employs instead a generic *Frequent Itemset Mining* tool, not specifically designed for solving the

A preliminary version of this paper appeared in the Proceedings of the 7th Learning and Intelligent OptimizatioN Conference (2013).

Composite Motif problem, in which the internal handling of the constraints is rather blind to their nature.

Input parameters. One of CMStalker's main design goal was to bring simplicity of operation to end-users. This meant ruling out many of the so-called *nuisance parameters* [21], i.e., user-defined quantities that drive the internal algorithms but that are either perceived of little biological relevance, or whose precise values are difficult to set in advance. As a consequence of this objective, CMStalker only needs two mandatory input parameters, namely a set of DNA sequences to be searched for motif clusters and a set of *Position Weight Matrices* (*PWMs*) describing TF-DNA binding affinities. Such matrices are typically obtained from known and trusted databases, such as TRANSFAC [55] and Jaspar [39]. However, concerning this latter point, we stress that CMStalker is also able to run third party software packages for the de-novo motif discovery (see [29]); these pieces of software can be employed to "synthesize" new PWMs, to be used as an alternative to known matrices from the public databases. PWM have been selected as formalism for modeling the specificity of protein-DNA interactions since it is a widely used standard [44], however, the core algorithm with minor modifications would be able to use also other models for TF binding such as HMM, or the recently proposed transcription factor flexible model (TFFM) of Mathelier and Wasserman [31].

Experiments. We evaluate CMStalker on three different benchmarks of quite different nature: (1) the twelve datasets (from various organisms) considered by Klepper et al. in [26]; (2) the synthetic dataset introduced by Xie et al. in [56] (from human, mouse and chicken genomes); (3) the dataset of cis-regulatory regions in early stage development of Drosophila considered by Ivan et al. in [22]. We compare the results obtained by CMStalker, using various performance metrics, against those of twelve[1] other published tools. These are the eight tools analyzed in [26] (namely CisModule [59], Cister [14], Cluster-Buster [15], Composite Module Analyst [25], MCAST [3], ModuleSearcher [1], MSCAN [23] and Stubb [43]), and moreover COMPO [40], MOPAT [20], CORECLUST[33], and CPModule [18], [46]. The resulting tool/dataset/metrics matrix is clearly "incomplete", since no single tool has been previously tested over all the above mentioned data, but nonetheless quite dense to provide a solid ground for evaluations. Except for MOPAT and CPModule, we only report published results, in order to avoid the risk of sub-optimal use of some tool (e.g., wrong parameter settings).

Overview of results. Extensive experiments with twelve data sets in [26] and eleven state-of-the-art tools are reported in Section V, showing an average value of the correlation coefficient of 0.54 (against a value 0.41 of the closest competitor). The correlation coefficient is a standard measure that takes real values in the range $[-1..+1]$, with $+1$ being the situation of perfect correlation, -1 of perfect anti-correlation, and 0 of statistical independence (random correlation). This improvements in output quality due to CMStalker is statistically significant, as measured with the Friedman aligned test. On a second benchmark set in [56], for increasing level of induced noise, CMStalker has qualitatively roughly the same performance levels as the best of five other methods.

The results computed by CMStalker have quality which is superior, more often than not, to those of the competitors (see Section V), especially if one considers that they have been obtained using a single set of parameter values (i.e., no parameter fitting on the single data sets has been done). CMStalker can be classified as a *conservative* motif discovery tool. Actually, when CMStalker does not gather sufficiently strong evidence to report a combination of TFBS as being potentially functional, it does not report any answers at all. While this phenomenon in general sacrifices sensitivity, it allowed us to adopt a search methodology based on parameter relaxation that we regard as one the key design choices leading to the observed good experimental results.

The problem of detecting functional regions in DNA sequence data is a quite challenging one, and a killer "general" computational solutions might not exist. While more comparisons are needed, against yet other methods and/or using different benchmark data, we feel however that the results obtained give comforting evidence of the merits of the approach embodied in CMStalker in realistic application scenarios.

CHIP-seq variants. Chromatine immunoprecipitation experiments coupled with next generation sequencing (CHIP-seq) [4] [51] produce data useful for the in-vivo identification of functional TFBS in TF-specific and tissue-specific experiments. Processing CHIP-seq data, however, requires specialized algorithmic techniques that are specific to this technology, in order to be effective. In this work we do not rely on this type of additional data, and thus we compare our performance only with those methods that work in a setting similar to the one we adopted. The development of variants of our algorithm that take advantage of CHIP-seq data is left for future research.

Organization of the paper. The rest of the paper is organized as follows: Section II outlines previous results and tools that provide the landscape in which to place our present contribution; Section III introduces the technical notions required in the following Section IV, which describes the algorithm embodied in CMStalker; finally, Section V reports experimental results.

II. Composite motif models and related work

A (simple) motif is a model of binding sites for just a transcription factor. In the following, for ease of language, we will use the terms motif and binding sites interchangeably. Simple motifs are often described by means of *Position Weight Matrices* (*PWMs*) (see Section III for formal definition). Several hundreds of experimentally determined PWMs for identifying TFBS are available in databases such as TRANSFAC [55] and Jaspar [39]. However, the highly degenerate nature of the TFBS implies that, when scanning sequences for PWM matches, many false positive non-functional matches are quite likely to occur. Thus additional information and criteria are needed to filter out false positive matches.

Composite motifs (a.k.a. *combinatorial* or *higher-order motifs*) describe sets of simple motifs, sometimes called *boxes*, separated by stretches of DNA of limited size (say, < 20 bp). The distance constraint seems typically more important

[1] See Table 2 in Suppl. Materials.

than motif ordering; while alterations in the distances between simple motifs usually kill transcription, a different motif order may be simply associated with a different transcriptional behavior (see [42]).

Before proceeding, we must address an important issue, which is only apparently terminological. Given a composite motif and a set of corresponding binding sites in different input sequences, some authors identify the regions spanned by those sites as *Cis-Regulatory Modules* (hereafter *CRMs*) [48], [11], [26], [33]. This is all but a universally accepted viewpoint. Many influencing works in the fields clearly distinguish the two concepts of composite motif and CRM (see, e.g., [6], [22], [45]), attaching to the latter a broader semantics.

Such broader characterization of the notion of CRMs has to do more with functional output (i.e., the "battery of gene they orchestrate" [22]) than with site affinities, as is the case of composite motifs. The already mentioned paper by Su et al. [45] even equates the notions of CRM and that of a whole promoter. A practical consequence of this viewpoint is that the functional regions that must be sought by computational methods are much longer than those typically spanned by composite motif clusters (compare the 500-1000 bp target size in [22] against the maximum 100 bp target size of [48]). CMStalker has been designed to detect composite motifs in the above sense rather than CRMs; hence, in this paper, we generally avoid any reference to Cis-Regulatory Modules. However, it is true that the detection of recurrent motif clusters is one of the techniques adopted to locate such functional modules. It is precisely for this reason that we have tested CMStalker also on a dataset of Cis-Regulatory Modules in the Drosophila genome (see Section V).

In CMStalker we adopt the simple set model to describe composite motifs. According to this, a higher-order motif is just characterized by its component simple motifs, with just a mild uniformity constraints on the total span of the region containing the composite motif. We made this choice for our tool to require minimal information about the motif structure; actually, while inter-box distances are important for the characterization of the mechanisms of functional sites, the need for their specification clearly limits the tools applicability when only detection is sought. However, as we shall see in Section IV, our algorithm uses information on the overall span of the prospective composite motifs to enforce some regularities in the overall length of the reported output set.

The set model is simple and yet quite a reasonable one; it has been adopted in some of the earliest attempts to identify clusters of binding sites in the Drosophila genome [6] and in the muscle and liver datasets that we also consider in the present paper [2]. More recently, the set model has been adopted in COMPO [40] and CPModule [18], [46].

Another popular model still regards composite motifs as sets of simple motifs, but includes ordering and distance information on their occurrences. *Structured motifs* or *tuple motif models* are the often used names for these cluster of site abstractions. A pioneering work on structured motif discovery is that by Marsan and Sagot [30]. One of principal merits of this influential contribution is to highlight the power of a direct approach to finding clusters of motifs, namely to make it possible even for a single "subtle" motif to be captured when considered in combination with other signals (however, see [13] for an in-depth comparison of direct methods with algorithms based on the construction of structured motifs by simple motif combination). This model is also very often used in researches where simple *dyad motifs* (*i.e.*, composite motifs made of just two boxes) are sought [19], [12], [57], [58].

A third model often adopted to represent composite motifs is the *Hidden Markov Model* (*HMM*). This is very powerful and much a richer abstraction than the previous two. Essentially, a HMM for composite motif description is a Markov chain whose states are associated to positions in the simple motifs or in the background; in each state the HMM "emits" one of the four nucleotides according to a probability distribution (computed either from the PWM representation of the motif or from the background frequencies). Different HMMs models adopt different definitions for the transition probabilities and differ in the way they are computed. By accurately defining these probabilities, a HMM may model sets of unstructured motif as well as structured motifs. One of the first attempts to use HMMs to discover clusters of motifs is the one described in [34]. Other popular tools that model composite motifs as HMMs include *Cister* [14], *Cluster-Buster* [15], *Stubb* [43], and *CORECLUST* [33]. Among these, *Stubb* is especially worth mentioning because it computes the transition probabilities using expectation-maximization, without requiring any user-supplied parameters.

III. Technical definitions

In this Section we briefly define/recall the fundamental notions used in the rest of the paper (a summary can be found in Table 1 of Supplementary Materials).

Let $\mathcal{D} = \{\texttt{a}, \texttt{c}, \texttt{g}, \texttt{t}\}$ be the alphabet representing the four DNA base pairs (bp). A short word $w \in \mathcal{D}^*$ is called an *oligonucleotide*, or simply *oligo*. Typically $|w| \leq 20$. Let $\mathcal{S} \subseteq \mathcal{D}^*$ be a set of n DNA fragments, *e.g.*, sequences of bps from the promoter regions of some genes. We say that w *occurs* in $S \in \mathcal{S}$ if and only if w is a substring of S.

From a computational point of view, a *DNA motif* (or simply *motif*) is a representation of a set of oligos that describe potential Transcription Factor (TF) binding loci. The representation can be made according to one of a number of models presented in the literature. Here we adopt the well-known Position Weight Matrices (PWMs).

A PWM $M = (m_{b,j})$, $b \in \mathcal{D}, j = 1, \ldots, k$, is a $4 \times k$ real matrix. The element $m_{b,j}$ gives a score for nucleotide b being found at position j in the subset of length-k oligos that M is meant to represent. Scores are typically computed from frequency values. But how can we associate oligos to PWMs? Different answers have been given to this question in the literature (see, for instance, [24], [8], [37]). Here we adopt perhaps the simplest one.

Let M be a PWM and consider a word $w = w_1 w_2 \ldots w_k$ over $\mathcal{D}^k$: We let *score of* w (w.r.t. M) denote the sum of the scores of each nucleotides, i.e., $S_M(w) = \sum_{j=1}^{k} m_{w_j,j}$. The maximum possible score given by M to any word in $\mathcal{D}^k$ is clearly $S_M = \sum_{j=1}^{k} \max_{b \in D} m_{b,j}$. Then we say that M

represents word w iff $\frac{S_M(w)}{S_M} \geq \tau$, for some threshold value $\tau \in (0,1]$. In the following, we will identify motifs with their matrix representation.

A motif M has a *match* (or *occurrence*) in $S \in \mathcal{S}$ if and only if there is a substring of S that is represented by M. We borrow some terminology from [40] and call *discretization* the process of determining the matches of a motif in a set of DNA sequences.

A *motif class* is a set of motifs. Ideally, in CMStalker all the motifs in a class describe potential binding sites for a single Transcription Factor. For this reason, we often freely speak of Transcription Factors to refer to motif classes. A *factor match* in a DNA sequence is thus a match of any of the motifs in the class associated to that factor. Note that motif classes have the ability to represent oligos of different lengths for the same TF, since different matrices usually exist for the same factor that have a different number of columns. Let $\mathcal{F}$ be the set of factors having matches in $\mathcal{S}$. We consider a one-to-one mapping between $\mathcal{F}$ and an arbitrary alphabet $\mathcal{R}$ of $|\mathcal{F}|$ symbols, which we refer to $\mathcal{R}$ as the *mapping alphabet*.

A *combinatorial group* (or just *group*) is a collection of not necessarily distinct TFs that have close-by matches in a sufficiently large fraction of the input sequences (assuming the number N of sequences is clearly understood, we silently equate the fraction $q \in (0,1]$ and the absolute number of sequences $\lceil q \cdot N \rceil$). The minimum fraction allowed for a set of TFs to be considered a combinatorial group is termed *quorum*. The *width* or *span* of a group match in a sequence S is the "distance" (measured in bps) between the first bps of first and last TF match of the group in S.

In set-theoretic terms, groups are multisets. In CMStalker they are represented as character sorted strings over the mapping alphabet $\mathcal{R}$. CMStalker's algorithmic core (see Section IV-C) efficiently implements special union and intersection operations (denoted $\vee$ and $\wedge$, respectively), defined on maximal collections of pairs $\langle M, n\rangle$, where M is a multiset and n is a positive integer. Maximality means that if P is one such collection and $\langle M, n\rangle \in P$, then there is no other $\langle \bar{M}, \bar{n}\rangle$ in P with both $\bar{M} \supseteq M$ and $\bar{n} \geq n$.
The definition of $\vee$ is easy[2]:

$$\text{Union:} \quad P \vee Q = \{p \in P \cup Q : p \text{ is maximal in } P \cup Q\}$$

where $\cup$ denotes union over sets. As for $\wedge$, we first define it for collections containing just a single pair:

$$\begin{aligned}\{\langle M_1, n_1\rangle\} \wedge \{\langle M_2, n_2\rangle\} = \\ \{\langle M_1 \cap M_2, n_1 + n_2\rangle\}_{\text{if } M_1 \cap M_2 \neq \emptyset} \cup \\ \{\langle M_1, n_1\rangle\}_{\text{if } M_1 \setminus M_2 \neq \emptyset} \cup \{\langle M_2, n_2\rangle\}_{\text{if } M_2 \setminus M_1 \neq \emptyset}\end{aligned}$$

Then, for arbitrary sets $P_1 = \{p_i^{(1)}\}_{i=1,\ldots,h}$ and $P_2 = \{p_j^{(2)}\}_{j=1,\ldots,k}$:

$$\text{Intersection:} \quad P_1 \wedge P_2 = \vee_{i,j}\left(\left\{p_i^{(1)}\right\} \wedge \left\{p_j^{(2)}\right\}\right).$$

Our last definition is that of *Composite Motif* (*CM*). A CM is a set of close-by TF matches in some input sequence. CMs represent CMStalker's *best guess* for functional TF binding regions. Note that no quorum constraint is imposed to composite motifs. Indeed, as collection of factor matches, composite motifs are clearly unique objects. As we shall see in Section IV, CMStalker builds composite motifs by extending the matching of some combinatorial group.

IV. Algorithm

CMStalker main operation mode, which we describe in this paper, is composite motif discovery in a set $\mathcal{S} = \{S_1, \ldots, S_N\} \subseteq \mathcal{D}^*$ of DNA sequences, using a collection of PWMs. However, CMStalker is also able to run a number of third-party motif discovery tools, the output of which can then be used either to directly discover putative composite motifs, or to create a number of PWMs to be later used under the main operation mode.

In many cases, the number of matrices available, which describe the binding affinities of the factors involved in the experimental protocol upstream data analysis, is much larger than the number of such factors. This often happens when many PWMs are loaded from an annotated database or when the input matrices are produced by third-party motif discovery tools. In particular, a single factor may be described by many different matrices of possibly different lengths. CMStalker is able to handle this latter state of affairs, provided that the user has some additional knowledge on the biological experiment.

1) The user knows number and identities of the TFs involved. In this case s/he may submit to CMStalker different input files with different sets of matrices, each one describing affinities for a single TF.
2) The user only knows the presumed number of TFs involved, but either does not knows their detailed identities or, even if s/he does, s/he is not able to provide clean sets of PWMs for each.

If none of the above applies, CMStalker will treat any input matrix (or matrix synthesized by an external tool) as describing a distinct factor.

Overall, CMStalker's main operation mode amounts to the following five steps, which we describe in details in the rest of this section.

1) *PWM clustering* (optional), to organize the matrices in groups (factors) of close-by PWMs;
2) *Discretization*, to detect factor matches in the input sequences;
3) *Group finding*, which is the crucial combinatorial group detection step;
4) *Group filtering*, for the screening of groups according to various filtering criteria;
5) *Composite motif prediction*, to form the putative composite motifs out of groups.

A. PWM clustering

This step is performed if (and only if) case 2) above applies, i.e., if the user provides the number of relevant TFs together

[2] A quick recap of the standard operations on multisets used in the definitions of $\vee$ and $\wedge$ can be found in the Supplementary materials.

with one "undistinguished" set of matrices (i.e., a single input PWM file). Note that in case 1) the "clustering" is implicitly performed by the user by submitting to CMStalker multiple input PWM files.

To perform the clustering, CMStalker first builds a weighted adjacency graph whose nodes are the matrices and edges the pairs (M_1, M_2) such that the similarity between M_1 and M_2 is above a given threshold. Currently, CMStalker uses pairwise normalized correlation [47].

Then, CMStalker executes a *single-linkage* partitioning step of the graph vertices, which essentially reduces to a variation of Kruskal's algorithm for the construction of a maximum cost spanning forest. More precisely, let N_m be the number of matrices and let N_F be the number of TFs; then CMStalker performs at most $\min\{|E|, N_m - N_F\}$ steps of Kruskal's algorithm, where E is the set of edges in the similarity graph. The returned clusters are the graphs induced by the vertices in distinct trees of the forest. Finally, CMStalker identifies the dense cores in each set of the partition, via pseudo-cliques enumeration [50], returning them as the computed clusters.

B. Discretization

Even with the most accurate PWM description of a motif, the problem of determining the "true" motif matches in the input sequences is all but a trivial task. Actually, whatever the algorithm adopted, there is always the problem of setting some threshold σ to separate matches from non-matches, a choice that may have a dramatic impact on the tool's performance. In general, low thresholds improve sensitivity while high thresholds may improve the rate of positive predicted values (PPVs). The strategy adopted in CMStalker is to moderately privilege sensitivity during discretization, with the hope to increase the positive predicted rate thanks to the combinatorial effect of close-by matches. We turned this general goal into precise threshold values by taking TF representation issues as well as computational efficiency concerns into account.

We observe that, when different "accurate" matrices are known to describe functional loci for a single TF, it is nonetheless very likely that, for any particular site, some of them do dot produce high scores. Moreover, many matrices almost unavoidably imply a lot of matches, which in turn have a strong impact on the computational cost of our algorithm (and not only ours), as we will point out in Section IV-F. For the above reasons, we allow for the adoption of more than one threshold value; in particular, in our experiments we start with the quite high 0.7 value adopted by TAMO[3] for TFs represented by five or more matrices and decrease it gradually to the minimum value of 0.5 for motifs represented by one matrix. The latter allows us to possibly catch weak signals without paying too much in terms of computational cost.

All the experiments of Section V were performed using fixed threshold values. These can be varied in the configuration file (which means they are essentially hidden to the typical user), and what one can reasonably expect is that different thresholds on different datasets may return very different results. The average good quality of CMStalker's predictions, across different datasets, suggests that the above criterion (rather than the particular threshold values adopted) has indeed some merits.

C. Group finding

The previous two steps result in a set of motif classes (factors) and a set of factors matches, which are the "input" to the group finding step. To this end, CMStalker uses a simple search strategy, with the aim of trading computation time for accuracy: it progressively relaxes two internal parameters until each motif class is possibly included in at least one group. These parameters are the maximum allowed combinatorial group width and the minimum quorum for combinatorial groups and can be set in the configuration file.

Formally, let $\{W_1, \ldots, W_r\}$ be a set of window sizes and let $\{q_1, \ldots, q_s\}$ be a set of quorum values, with $W_1 < W_2 < \ldots < W_r$ and $1 \geq q_1 > q_2 > \ldots > q_s > 0$. For a given window size value W and sequence S_i, we say that a multiset m over $\mathcal{R}$ is *feasible* iff each factor of m corresponds to a match in S_i and the span of all the matches in S_i is bounded by W.

Intuitively, for each pair W, q, CMStalker computes all maximal feasible groups with respect to window size W and quorum q (*i.e.*, that have matches in at least $\lceil q \cdot N \rceil$ sequences). If all letters of $\mathcal{R}$ are included in the found groups, CMStalker stops. Otherwise, CMStalker relaxes, in alternate order, one of the constraints on width (considering a larger value) and quorum (smaller value).

Algorithm IV-C describes the group finding step in details. Note that, thanks to the properties of the $\vee$ and $\wedge$ operators, the pairs $\langle M, n \rangle$ included in G_N are maximal, with n satisfying the last fixed quorum value. Clearly, even with the weakest parameter values (*i.e.*, widest window and smallest quorum), some factors may not be represented in G. This is not necessarily a problem, since the user may have provided PWMs for irrelevant factors.

In current CMStalker implementation, the relaxation step (step 15) involves the window size and the quorum value in an alternate order. Note that a verbatim implementation of Algorithm IV-C might be quite inefficient. For instance, when relaxing the quorum value, step 3 needs not be computed. On the other hand, the pairwise intersections in step 8 can be performed quite efficiently thanks to the character sorted string representation of multisets of factors.

D. Group filtering

The filtering stage aims at picking groups that: (1) are strong enough from a conservation point of view; (2) further exhibit a regularity in terms of span of the matches; (3) satisfy a simple statistic criterion.

1) "Group filtering": This phase aims at eliminating those groups that are not "strong enough". As outlined in Section IV-B, a weak TF match may be allowed, due to possibly low thresholds for poorly represented TFs. At the group level, though, we impose stronger requirements, with the aims of both improving PPVs

[3] Our software uses TAMO [17] for "low-level" motif representation and manipulation.

```
1:  $W \leftarrow W_1$ and $q \leftarrow q_1$
2:  for $i = 1, \ldots, N$ do
3:      Compute the maximal multisets $M_1^{(i)}, \ldots, M_{n_i}^{(i)}$ that are feasible for $W$ and $S_i$
4:      $P_i \leftarrow \{\langle M_1^{(i)}, 1\rangle, \ldots, \langle M_{n_i}^{(i)}, 1\rangle\}$
5:  end for
6:  $G_1 \leftarrow P_1$
7:  for $i = 2, \ldots, N$ do
8:      $G_i \leftarrow G_{i-1} \wedge P_i$
9:  end for
10: From $G_N$ discard all pairs $\langle M, n\rangle$ such that $n < \lceil q \cdot N \rceil$
11: if multisets in $G_N$ include all the letters of $\mathcal{R}$ or $W = W_r$ and $q = q_s$ then
12:     $G \leftarrow \{M : \langle M, n\rangle \in G_N\}$
13:     return $G$
14: else
15:     Relax $W$ or $q$ (or both)
16:     Jump to step 2
17: end if
```

and possibly limiting the number of candidates, which highly affects the cost of the group finding step (see Section IV-F). For this purpose, a threshold value τ is again specified in the CMStalker configuration and use to discard those group matches that exhibit less than $\tau \times 100$ percent of conserved bps over all TF matches of the group. Here conserved simply means that the particular bp it's the one that scores the highest in the corresponding PWM column. Also the value of this parameter was kept constant in all the experiments performed.

2) "P-value filtering". CMStalker computes the p-value of each combinatorial group according to the methodology adopted in [40] for the modules, and discards those groups with p-value higher than a user specified threshold (which defaults to the "usual" value 0.05).
3) "Group clustering". This step is performed only under the ZOOPS (Zero or One Occurrence per Sequence) model of motif distribution, which is the default in CMStalker (the alternative model is usually termed ANR, which stands for "Any Number of Repetitions"). For each remaining maximal group, CMStalker performs a clustering of all its matches with respect to the widths of the matches themselves. For each sequence where the group occurs, it then returns the match whose width is closest to the most recurring group width. Note that each sequence may still contain more than one group, but not of the same "type" (i.e., factor composition).

   The choice of this clustering step is motivated by the observation that there is a wide literature on the so called *structured motifs* (see, *e.g.*, [36]) where the order of the TF matches and the inner spacings between the single motifs play a crucial role. We do not use any of these pieces of information, but simply note that, if the spacings are fixed, then the width of the composite motif is fixed (or exhibits small oscillations) as well. Hence, this information might be captured by a clustering strategy.

E. Composite motif prediction

For any "surviving" group g in G, CMStalker first retrieves its actual matches from the input sequences; then tries to merge overlapping or close-by groups of matches provided that the resulting span of the factor matches does not violate the window constraint. These merged groups are the candidate composite motifs being predicted. However, under the ZOOPS model, only one composite motif is returned, namely the one that contains more factor matches.

Note that it is precisely this step that makes composite motifs unique objects, in the sense that they do not have to satisfy (after merging) any quorum constraints.

F. Computational complexity

The cost of the bare CMStalker algorithm is dominated by the module finding step or, more precisely, by the combinatorial group finding sub-process. It is easy to see that this can be exponential in the length of the longest group g (regarded as a string over $\mathcal{R}$) in any of the initial sets M_i's, simply because g may have an exponential number of maximal subgroups that satisfy also the quorum constraint. In turn, the length of g may be of the order of module width and hence of sequence length.

The above worst-case cost can indeed be achieved, especially if one gives many PWMs in input to CMStalker (say, all the PWMs available in the TRANSFAC database), that are very likely to incur in a huge number of (non functional) factor matches. One possibility to keep running time under control is to bound the cardinality of the groups and/or the number of matches in each sequence. For this reason, there are two additional parameters that control these complexity related quantities, which are stored in the configuration file. We shall say more on this in Section V, where we show the actual running times obtained on input the "synthetic benchmark" (that we used for size scalability reasons).

At the other extreme, there is the situation where we only have few TFs and look for sites where all of them bind (as for the TRANSCompel datasets of Section V). In this case the cost of the sub-process is linear in the number of sequences.

If clustering is required the overall cost is also bounded from below by a quadratic function of the number of input PWMs, since the algorithm computes all the possible distances among pairs of matrices. This cost indeed dominates the computation time on the TRANSCompel datasets.

V. Experiments

In this section we present the results obtained in experiments performed on three different benchmark datasets, each described in one of the subsequent subsections. We first describe the parameter set up for CMStalker, and the functions used to measure the algorithms' output quality. Next we describe in details the three mentioned benchmarks, which we refer to briefly as the COMPOSITE, XIE, and REDFLY benchmarks, respectively.

A. *CMStalker's setup*

All the experiments were performed with a almost fixed parameter configuration, which we describe in this section. The overall good experimental results obtained suggest that such standard setup can guarantee good performances across very different datasets. All the parameters are set in the configuration file only, which essentially means that they are hidden from the typical end user.

- The threshold parameter were fixed as reported in Section IV-B.
- The p-value adopted in the filtering stage (Section IV-D) was 0.05. However, in case of the TRANSCompel data, where we were just looking for pairs of motifs corresponding to exactly two different TFs, then we simply did not perform any p-value filtering (i.e., set p-value=1). This reflects our viewpoint about the merits of statistical parameters in motif finding, namely that they might render evident "relative" rather than "absolute" quality of potential binding locations. Here we have just one candidate motif, which is returned provided that it is strong enough by combinatorial evidence.
- the two crucial "optimization" parameters, namely quorum q and window size W (Section IV-C), were fixed as follows: $q = \{0.9, 0.8, 0.7, 0.6., 0.5., 0.4, 0.3, 0.2, 0.1\}$, $W = \{50, 75, 100, 125, 150\}$. However, for the REDFLY dataset we used different values of W, to conform to the experimental setting of [22] (see Section V-E).
- To limit the possibly exponential growth in running times (see Section IV-F), we included two additional parameters upper bounding: (1) the number of hits returned by the discretization stage in any input sequence, and (2) the cardinality of the combinatorial groups (see also [40]). We call these MAXHITS and MAXGROUP, respectively. MAXGROUP was kept fixed to 8 in all the experiments: this is usually sufficient to model the clusters of functional TFBSs that occur in practice. Bounding the number of hits in any input sequence seems to be more delicate. Recall that we set a low value for the threshold parameter, with the aim at not loosing weak signals. However, doing so may produce a large number of hits that severely impact on the computational cost (Section IV-F). This does not happen in the COMPOSITE benchmark, but it is definitely the case for a number of datasets in the other two benchmarks. To set MAXHITS in general, we performed a study using the XIE benchmark, which allows a better control of the input parameters (in particular, the number of PWMs). Figure 1 shows the observed running times for different datasets (with varying number of input PWMs) of the XIE benchmark. The exponential nature of the running time as a function of the number of hits (confirming the worst-case theoretical analysis and the impact of an accurate/inaccurate choice of the input PWM) suggested us to set MAXHITS=20 for XIE and (at least initially also) for REDFLY datasets. However, in an attempt to catch more results, we then set MAXHITS=30 for the latter but payed (on input some hard datasets) a high computation cost, with runs that lasted several days.

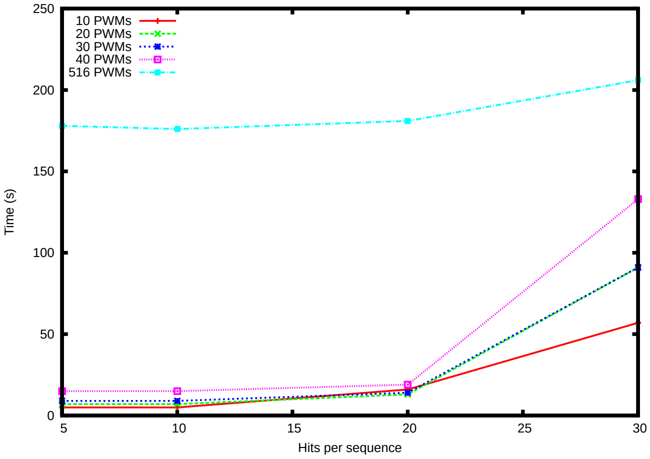

Fig. 1. Running times of CMStalker for XIE datasets with a variable number of PWMs. The graph highlights the exponential rise of the run time function starting at MAXHITS=30.

B. *Measures used to assess the tool's output quality at the nucleotide level*

Each composite motif prediction algorithm will output a set of TF labels and a set of corresponding TFBS (intervals on the input sequences) which, together, form the algorithm's prediction of the most likely composite motif for the given input. The use of benchmarks as input allows us to measure the quality of each algorithm's output in terms of a measure of similarity between the true solution and the reported prediction. Among the well known methodological frameworks for this task we follow that of Burset et al. [7] and Tompa et al. [49], which we outline here below.

For each nucleotide in the input sequences we can label it with one element from the binary domain $\{\text{Positive}, \text{Negative}\}$, where Positive (abbr. P) indicates that the nucleotide belongs to a TFBS in the algorithm's prediction, and Negative (abbr. N) indicates that the nucleotide does not belong to a TFBS in the algorithm's prediction. Each prediction can be either True (abbr. T) or False (abbr. F) thus we can group (and count) the nucleotides into four classes according to the prediction label, and to the fact that it corresponds to reality or not.

a) nTP is the number of nucleotide labeled $Positive$, that are really part of a TFBS.
b) nFP is the number of nucleotide labeled $Positive$, that are not really part of a TFBS.
c) nTN is the number of nucleotide labeled $Negative$, that are not really part of a TFBS.
d) nFN is the number of nucleotide labeled $Negative$, that are really part of a TFBS.

Clearly we have high similarity between prediction and reality when nTP and nTN are large numbers relative to nFP and nFN. In order to capture this intuition several synthetic indices have can be devised. Denote with nPr the number of nucleotides in predicted TFBS, we have: $nPr = nTP + nFP$. Denote with nGs the number of nucleotides in real TFBS (also called the "golden standard"), we have: $nGs = nTP + nFN$. A basic measure is *Sensitivity*[4] (abbr. Sn) defined as the ratio between the number of true predictions of nucleotides in TFBS over the number of nucleotides in real TFBS:

$$Sn = \frac{nTP}{nGs} = \frac{nTP}{nTP + nFN} \tag{1}$$

A second basic measure is *Precision*[5] (abbr. Pr) defined as the ratio between the number of true predictions of nucleotides in TFBS over the total number of nucleotides in predicted TFBS:

$$Pr = \frac{nTP}{nPr} = \frac{nTP}{nTP + nFP} \tag{2}$$

It has been noticed, however, that certain pairs of measures are antagonistic, in the sense that it is easy to devise relatively trivial algorithmic strategies to inflate one at the expense of the other[6]. For this reason other functions have been proposed which often can be seen as "mean" values of the antagonistic pairs. For example in information retrieval it is often used the F1 measure that is the harmonic mean of the precision and the sensitivity measures.

Burset et al. [7] introduced the *correlation coefficient* (abbr. CC) specifically with the aim of having one such balanced measure in the area of gene structure prediction, by using all 4 values nTP, nFP, nTN and nFN. The correlation coefficient has an easy statistical interpretation as a correlation between two random variables.

$$CC = \frac{nTP \cdot nTN - nFN \cdot nFP}{\sqrt{(nTP + nFN)(nTN + nFP)(nTP + nFP)(nTN + nFN)}}$$

A second balanced measure has been proposed by Pevzner and Sze [35], called the *Performance Coefficient* (abbr. *PC*) and defined as:

$$PC = \frac{nTP}{nTP + nFN + nFP} \tag{3}$$

Tompa et al. [49] report also the *Average Site Performance* (abbr. *ASP*), the arithmetic mean of precision and sensitivity:

$$ASP = \frac{Sn + Pr}{2} \tag{4}$$

In our experiments we will use CC as the main performance measure of accuracy, and the others as ancillary measures.

Ivan et al. [22] propose the following evaluation scheme for evaluating CRM prediction tools. For each data set the mean value of the length m of the true CRM in the input sequences is pre-computed and given to the prediction tools as an additional parameter. The prediction tools are required to output a prediction in which the predicted CRM in each sequence is of length m. In this framework, summing the contributions over the full set of sequences, we have forced the constraint $nGs = nPr$, thus several measures define above become redundant and, in this context it becomes safe to use just the PPV value as a measure of accuracy.

This type of analysis is termed "at nucleotide level" since the initial step is a classification of the nucleotides into four groups. A similar type of analysis can be carried out also at the motif level, when we provide the corresponding classification of the predicted TF (see the Supplementary Materials for an in-depth discussion).

C. COMPOSITE benchmark

1) COMPOSITE datasets: The first benchmark was presented in [26]. It is composed of 12 datasets from various organisms. This benchmark is composed of three subgroups:

a) ten data sets from TRANSCompel in [26]. Each data set contains a module made up of two TF with two binding sites, for different TFs from the following set: *AP1, Ets, NFAT, NFκB, CEBP, Ebox, AML, IRF, HMGIY, PU1*, and *Sp1*. Any dataset is named after the two component TFs: AP1-Ets, AP1-NFAT, AP1-NFκB, CEBP-NFκB, Ebox-Ets, Ets-NFκB, NFκB-HMGIY, PU1-IRF, and Sp1-Ets. In [26], all the matrices corresponding to the same TF were grouped to form an "equivalence set", and treated as if they were one.

b) One data set from [27] on *liver specific* transcription; this data set includes modules with up to nine binding sites of four different TFs.

c) One data set from [53] on *muscle specific* transcription. Modules are composed of a number of TFs ranging between eight sites and five.

We report in Table I basic statistics on the length and number of sequences, and on number and length of the composite modules. Further information can be found in Additional Files of [26].

2) Experimental set up: On the COMPOSITE data set we performed the most accurate analysis of CMStalker's behaviour (with respect to the other two benchmarks considered), especially thanks to the results already available in [26], which we will refer in the following as to the *assessment paper*. We were able to compare CMStalker against the eight tools considered in the assessment paper (CisModule [59], Cister [14], Cluster-Buster (CB) [15], Composite Module Analyst (CMA) [25], MCAST [3], ModuleSearcher (MS) [1], MSCAN [23] and Stubb [43]), as well as three other more recent tools, namely COMPO, developed by the same research group that performed the assessment [40], MOPAT [20], and CPModule [18].

Statistics for the tools already evaluated in [26] were downloaded from the authors' site. Regarding COMPO, we computed the statistics for liver-specific and muscle-specific datasets starting from the prediction files made available by the authors at the address `http://tare.medisin.ntnu.no/compo/`. For

[4] Sensitivity is also called "recall".

[5] Precision is also called PPV (positive predicted value).

[6] Note that, taken separately, precision and sensitivity use only two of the four values defined above, which makes it easier to increase one of them by relatively trivial means.

Dataset	Seqs	Total Size (bp)	Modules	Module size min,max,avg
AP1-Ets	16	14860	17	14,99,27
AP1-NFAT	8	6893	11	14,19,16
AP1-NFκB	7	6532	8	18,135,53
CEBP-NFκB	8	7308	8	44,118,84
Ebox-Ets	4	3489	6	16,50,25
Ets-AML	5	4053	5	13,30,19
IRF-NFκB	6	5344	6	23,71,43
NFκB-HMGIY	6	5393	7	10,32,13
PU1-IRF	5	4530	5	12,14,13
Sp1-Ets	7	5787	8	16,117,37
Liver	12	11943	14	26,176,112
Muscle	24	20427	24	14,294,120

TABLE I
A BRIEF OVERVIEW OF THE TEN TRANSCOMPEL SEQUENCE SETS AND THE LIVER AND MUSCLE DATASETS TAKEN VERBATIM FROM [26]. FURTHER INFORMATION CAN BE FOUND IN ADDITIONAL FILES OF [26].

the TRANSCompel data, we directly used the results provided at the same address. Predictions for MOPAT and CPModule were obtained as described in the Supplementary Materials.

3) Results at the Nucleotide level: The first set of experiments involves only CMStalker over the TRANSCompel datasets. For each dataset we performed two runs, one with matrices already separated by TF (*i.e.*, giving CMStalker two PWM input files), and one with mixed matrices. In this second case we pass to CMStalker only the information on the number of TFs involved (just two). As Figure 2 shows, the results obtained are essentially identical, indeed suggesting that the clustering phase was able to “recognize” the true motif classes. The fact that sometimes the “mixed PWMs” runs may produce better results (e.g., for the AP1-Ets and NFκB-HMGIY datasets of TRANSCompel) is not a contradiction. Actually, in such cases the clustering stage may have removed some matrices that produced spurious (non functional) matches. Of course, also the opposite situation could occur on different datasets.

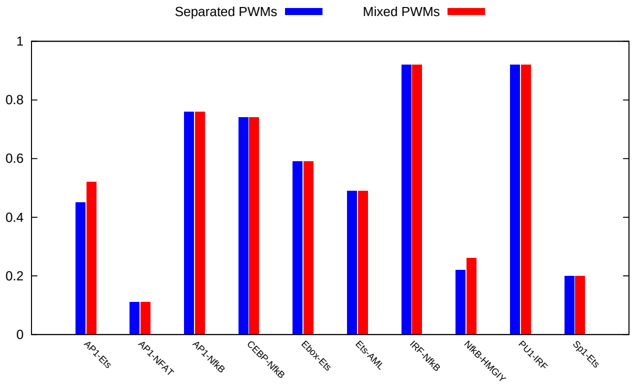

Fig. 2. nCC results for CMStalker on the TRANSCompel datasets, with separated (left/blue) or mixed (right/red) PWMs.

In the second set of experiments we compared CMStalker (fed with just one single PWM file) against the eleven competitor algorithms on the whole collection of twelve COMPOSITE datasets by measuring the Correlation Coefficient. The results are shown in Figure 3 and in Table 4 of the Supplementary materials.

In terms of ranking, CMStalker ranks first in 8 cases while Cluster-Buster, Module-Searcher, Mopat, and CPModule are first in one case for each method. CMStalker has a higher average nCC value of all the methods testes on this data set.

We conclude that, though CMStalker does not beat all other competing methods on all the data sets, still in terms of ranking and average nCC values has a far superior performance to all of them in this extensive experiment. However to make our statement more robust we perform also a statistical analysis, which at the best of our knowledge has not been done yet in this context.

4) Statistical significance: In the attempt to assess the statistical significance of these results, we first performed a Friedman aligned non-parametric test using the overall methodology detailed in [16]) that involved CMStalker and eleven other tools[7].

First we test the null hypothesis that all the considered algorithms behave similarly, and hence that the average ranks over the all data sets are essentially the same. This can be safely rejected, with a p-value about $8.9 \cdot 10^{-9}$.

Next we performed a post hoc test associated to the Friedman statistics, by considering CMStalker as the new proposed methods to be compared against the other eleven tools. Here the null hypothesis is that CMStalker has no better performance that each of the other tested methods and that the observed differences are caused by chance.

Table II shows the p-values of the eleven (Friedman aligned) comparisons, adjusted according to the Benjamini and Hochberg procedure [5] (also known as *False Discovery Rate*, *FDR*). This methodology takes into account possible type-I errors in the whole set of comparisons [16]. For all of the competing algorithm the null hypothesis can be safely rejected at a threshold well below 0.05. The best competitor is COMPO, against which the null hypothesis can be rejected with a p-value around 0.00005. All the other methods fall behind by several orders of magnitude.

COMPO $5{\cdot}10^{-5}$	MS $5{\cdot}10^{-12}$	CB $2{\cdot}10^{-12}$	CMA $5{\cdot}10^{-16}$	MSCAN 10^{-16}	CP $2{\cdot}10^{-24}$
MCAST $9{\cdot}10^{-31}$	MOPAT $3{\cdot}10^{-36}$	Cister $5{\cdot}10^{-51}$	Stubb $6{\cdot}10^{-53}$	CM $4{\cdot}10^{-103}$	

TABLE II
ADJUSTED p-VALUES FOR POST HOC COMPARISONS OF CMSTALKER AGAINST OTHER 11 TOOLS: MS = MODULESEARCHER, CB = CLUSTERBUSTER, CMA = COMPOSITE MODULE ANALYST, CP=CPMODULE, CM = CISMODULE

5) Comparison of CMStalker and COMPO: In this Section we provide further results to compare CMStalker and COMPO

[7] We excluded CORECLUST because of the limited availability of homogeneous data for the comparisons.

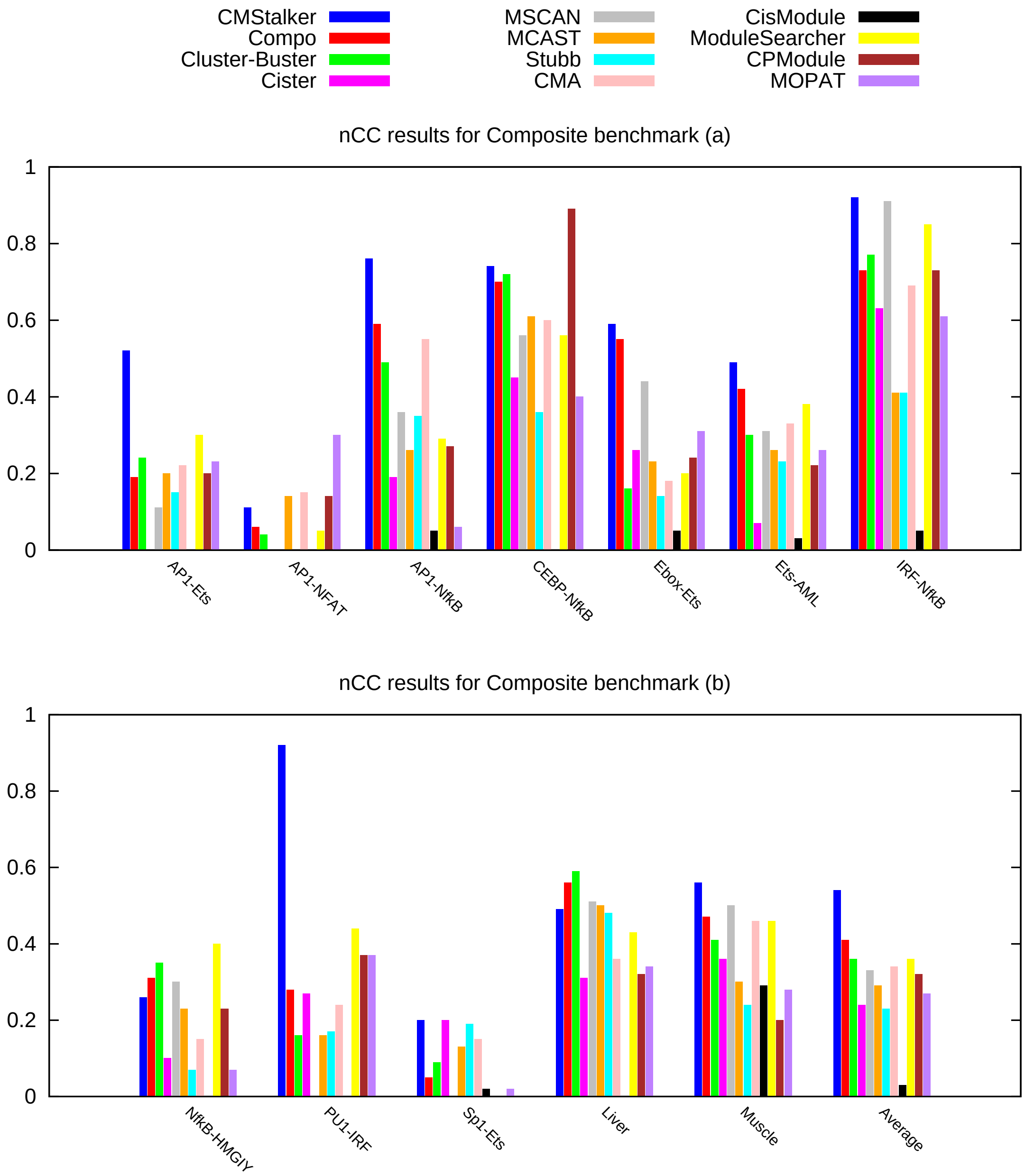

Fig. 3. Nucleotide level Correlation Coefficient values obtained by 12 tools on the whole collection of COMPOSITE datasets (negative values treated as 0).

on the COMPOSITE benchmark, in order to gain additional insights.

Figure 4 shows the results obtained on a wider sets of statistics on the COMPOSITE datasets, separating liver and muscle from the TRANSCompel's data. For the latter, the results shown in Figure 4 combine the results obtained on the single datasets (*i.e.*, counting the total numbers of positive, positive predicted, negative, and negative predicted nucleotides over the all datasets). Regarding liver and muscle, we point out that we have reported the statistics most favorable to COMPO among those obtained from the three different prediction files provided by the authors. CMStalker has slightly better performance than COMPO on muscle data on all three balanced measures (PC, ASP and CC), worse performance on Liver data, and better performance on TRANSCompel's data. In Supplementary Materials we report the precise numerical values used to draw the figures presented in this section, as well as the results obtained on additional TRANSCompel datasets, in which the "true" matrices corresponding to the TFs involved have been mixed with other (not relevant) PWMs.

6) Comparison of CMStalker and Coreclust: We compared CMStalker against CORECLUST [33] on the liver-specific dataset, which is the only one for which compatible data are available. It turns out that both methods are characterized by exactly the same nCC value, namely 0.56 [32].

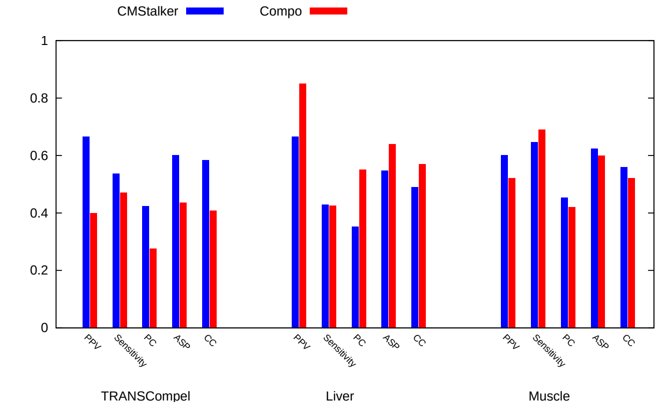

Fig. 4. Further nucleotide level comparisons between CMStalker and COMPO.

D. XIE benchmark

1) XIE dataset: The second benchmark is that constructed by Xie et al. [56], which is composed of 22 genomic sequences sampled from upstream regions of 22 randomly chosen genes in the mouse, chicken and human genomes, and of 516 TFs from TRANSFAC. The 22 genomic sequences are of same length (1000bps). For 20 sequences, binding loci for transcription factors Oct4, Sox2 and FoxD3 have been inserted within a region of at most 164bps, with inter-distances among TFBS sampled from a Poisson distribution of expected value 10. The order of the TF is preserved in each of the 20 sequences. For two sequences no insertion has been done.

We downloaded the whole benchmark from the companion site of [18], [46], which also included various other files: (1) one file with the correct positions and composition of the inserted modules (the *answer* file); (2) a file with a set of 516 TRANSFAC matrices corresponding to vertebrate TFs; (3) four collections composed of 10 PWM files each. Any file in the first collection stores 10 matrices, namely the ones corresponding to the TF loci inserted in the genomic sequences together with 7 "noisy" PWMs. The noisy matrices have been sampled from the above mentioned set of 516 TRANSFAC matrices. The other collections are characterized by an increasing amount of noisy matrices (17/20, 27/30, and 37/40, respectively).

2) Experimental set up: We compared CMStalker on this dataset against the results reported in [18], [46], that have been obtained by five different tools, namely the already mentioned Cister, Cluster-Buster, ModuleSearcher, and COMPO, as well as CPModule itself [18], [46].

Figure 5 shows the nucleotide level CC statistics for CMStalker and the tools already evaluated in [18], [46] as a function of the number of PWMs given in input. CMStalker's good results can be better appreciated when considering the fact that (differently from the other tools and, in particular, CPModule) it did not use any prior knowledge of the module's size (the proximity constraint of [18], [46]). As for the COMPOSITE benchmark, CMStalker simply used parameter relaxation in order to detect both size and quorum of the prospective combinatorial groups. We observe that CMStalker is roughly equivalent to the best performing methods in the range from 16 to 216 matrices. Afterwards there is a natural decay as the signal to noise ratio decreases. From this perspective, the behavior of Compo seems quite odd, as pointed out also in [46]. In the supplementary material we also report a zooming of the figures in the range from 10 to 40 PWMs.

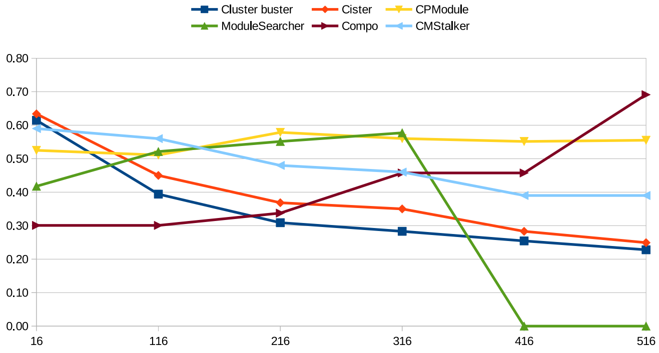

Fig. 5. Nucleotide level CC results on the XIE benchmark.

E. REDFLY benchmark

1) REDFLY dataset: The third benchmark contains a number of cis-regulatory regions which are provably functionally active during the blastoderm development stage of Drosophila. The benchmark is available as supplementary material of the paper [22], which we will frequently refer for both the experimental setting and the result comparisons. It is composed of 53 potentially relevant PWMs and of 33 datasets (collections of sequences), with a number of sequences per dataset ranging from a minimum of 4 to a maximum of 77 and summing to a total of 719 sequences, for approximately 5.7M bps. Each sequence of a given dataset includes a single CRM, whose length is typically different from those of other sequences in the same dataset. However, the experimental setting in [22] requires that any CRM discovery software being evaluated returns, for each sequence in a dataset, a single region of fixed length. This value is computed as the average real CRM length in that dataset and it is passed as a parameter to the prediction algorithm. Such average values range from 442 bps to 1248 bps.

The average lengths of the regulatory regions of the REDFLY dataset are significantly greater than those of the composite motifs typically detected by our software, and by most tools designed for the detection of clusters of binding sites considered in this paper. This feature makes it the less favorable to CMStalker (see also Section II). We made no attempt to adapt CMStalker to this state of affairs since we wanted to understand whether CMStalker "as is" can be suitably employed for the detection of regulatory elements spanning from few tens to a thousand bps. We ran CMStalker with the same set of parameters already adopted in the previously described experiments. However, to conform to the experimental setting of [22], we set the final window size parameter, for a given dataset, to the average CRM value for that dataset, by enlarging (or shrinking) the actual output. To avoid incurring in very high computation times, the values of MAXGROUP and MAXHITS parameters were set as described in Section V-A.

In [22] he results obtained by three different tools on this benchmark (the already mentioned Stubb, as well as D2Z-set and CSam, described in the same paper [22]) are reported, against which we compare CMStalker. The results obtained are by no means definitive. CMStalker can be regarded as quite "conservative" a motif discovery software; in many cases (30 over 33), according to its internal logic, CMStalker does not gather sufficient evidence for reporting a cluster of sites as a potential motif, and hence remains silent. In three cases the search was instead successful and the performance competitive against the other tools on the PPV metrics.

2) Experimental set up: We performed two sets of experiments, varying the MAXHITS parameter (Section V-A). In the first set, with MAXHITS=20 (and set all the other parameters to default values, with the exception of window size, for the reasons explained above), CMStalker returned just one answer, for the mapping1.cardiac.mesoderm dataset. We then set MAXHITS=30[8] and got two additional answers. Table IV shows the results obtained and the corresponding values reported in [22].

	Sequences		CRMs	CRM length		
Dataset	Seqs	Tot (bp)		min	max	avg
D1	5	28,085	5	126	927	561
D2	16	92,723	16	220	1373	579
D3	16	87,140	16	105	1415	544

TABLE III
DROSOPHILA DATASETS FOR WHICH CMSTALKER DETECTS A CRM: D1=MAPPING1.CARDIAC.MESODERM, D2=MAPPING1.ENDODERM, D3=MAPPING1.MESODERM.

	CMStalker	Stubb	D2Z-set	CSam
D1	**1.0**(0.06)	0.22(0.08)	0.28(**0.03**)	0.19(0.12)
D2	**0.39**(0.12)	0.24(**0.01**)	0.12(0.31)	0.26(**0.01**)
D3	**0.97**(**0.001**)	0.21(0.02)	0.17(0.09)	0.13(0.22)

TABLE IV
PPV RESULTS FOR CMSTALKER AND OTHER TOOLS ON SOME DROSOPHILA DATASETS (WITHIN PARENTHESIS THE EMPIRICAL P-VALUE). FOR THE ACTUAL DATASET NAMES REFER TO TABLE III

Note that Table IV reports the values of the PPV statistics, rather than Sensitivity, as in [22]. Actually, according to the evaluation protocol of [22], the two measures coincide. In our case, though, even when CMStalker reports some answers for a given dataset, it does not claim a CRM for all the sequences in that dataset. Hence sensitivity and PPV coincide in our case only if they are computed with respect to the true positive bps in the sequences for which CMStalker returned an answers. This means that the true CMStalker sensitivity can be much lower. However PPV seems a much better statistics for the purpose of establishing the usefulness of CMStalker for CMR discovery. The values of PPV, and the corresponding P-values, reported in Figure IV for CMStalker have been computed using the evaluation script available for download as supplementary material of [22].

VI. Discussion

In this paper we have presented CMStalker, a novel tool for Composite Module detection whose algorithmic core is based on purely combinatorial ideas. Using well-known benchmark data, of quite different nature, our software proved to be competitive against a number of state-of-the-art other tools.

We are aware that more comparisons are required, however, we think that some interesting findings have emerged from this work, all related to the power of simple motif combinations. First of all, that the good results exhibited by CMStalker have been obtained without using any sophisticated statistical filtering criteria; the combination of "right" simple sites were often strong enough to emerge from a huge pool of candidate motif clusters. Secondly, that the conceptually simple CMStalker architecture, based on a two-stage approach to composite motif finding (*i.e.*, first detect simple motifs, then combine them to form clusters of prospective functional motifs) proved to be competitive against other, more sophisticated approaches (see also [13]). In the third place, that progressive lowering the thresholds that defines *in silico* the DNA occupancy by a transcription factor, is a winning strategy that can be automated and thus be transparent to the user.

Giving the good results obtained, we are encouraged to carry further activities on CMStalker, including the ones listed below.

- Perform further comparisons, including other tools as well as other experimental frameworks (*e.g.*, those considered in [41] and [33]), not only with the goal of better estimating CMStalker's value, but also with the aim at understanding its limitations, *e.g.*, why it fails on input specific datasets, and how to possibly overcome them.
- Perform experiments where the input to CMStalker is produced by third-party motif finding tools. Clearly, whether or not good results can be achieved here will largely depend on the quality of the external tools performance. However, our hope here is to exploit CMStalker's ability to filter out false positives to achieve at least good PPVs.
- Improve the currently limited CMStalker's ability to predict whole regulatory regions, i.e., improving the Sensitivity of the algorithm on "CRM discovery datasets" while preventing a dramatic decrease of PPVs.

VII. Acknowledgments

We wish to thank the anonymous referees for their constructive comments. The present work is partially supported by the Flagship project *InterOmics* (PB.P05) which is funded and supported by the Italian MIUR and CNR organizations, and by the joint IIT-IFC Laboratory for Integrative System Medicine (LISM).

VIII. Author's Biographies

[8] Allowing as many as 30 hits per sequence caused some runs (e.g., on the mapping3.adult dataset) to last for more than one day.

Mauro Leoncini Mauro Leoncini is a professor of Computer Science with the Dipartimento di Scienze Fisiche, Informatiche e Matematiche of the University of Modena and Reggio Emilia. His research interests include computational complexity, parallel and wireless network algorithms, and bioinformatics. He got his Laurea cum Laude in Computer Science from the University of Pisa in 1984. After some years in the industry, he joined (in 1992) the Dipartimento di Informatica of the University of Pisa and then, as an associate professor, the University of Modena and Reggio Emilia in 2001. Since 2007 he is the Chair of the Computer Science program of the University of Modena and Reggio Emilia, where he teaches courses in the area of algorithms and data structures.

Manuela Montangero Manuela Montangero is a research associate (ricercatore) at the Dipartimento di Scienze Fisiche, Informatiche e Matematiche of the University of Modena and Reggio Emilia. Her research interests include distributed algorithms, combinatorics and bioinformatics. She got her Laurea cum Laude in Computer Science from the University of Pisa in 1996. In 2001 she got a PhD in Computer Science from the University of Salerno. From 2001 to 2004 she was at the Istituto di Informatica e Telematica of the National Research Council (CNR) of Pisa, with a PostDoc Fellowship at first and a temporary research associate position later. From 2005 to 2012 she was research associate at the Dipartimento di Ingegneria dell'Informazione and member of the PhD school council in Multiscale, Modelling, Computational Simulation and Characterization in Material and Life Sciences, at the University of Modena and Reggio Emilia. She is currently member of the PhD school council in Mathematics of the University of Ferrara, joint with the Universities of Modena and Reggio Emilia, and Parma. She is a collaborator of the Istituto di Informatica e Telematica, National Research Council (CNR) in Pisa.

Marco Pellegrini Marco Pellegrini received the Laurea degree (magna cum laude) in electronic engineering from Polytechnic of Milan in 1986 and the PhD degree in computer science from the New York University in 1991. From 1991 to 1995, he was a lecturer in the Department of Computer Science, King's College, London. In 1995, he joined the Italian National Research Council (CNR), where he is a senior scientist as of 1998. His research interests are in computational geometry, analysis of algorithms, information retrieval, and computational biology, with about sixty research papers in international journals and conference proceedings. In 1991, he was awarded a postdoctoral fellowship by the International Computer Science Institute, Berkeley. He has been a coordinator of the Computational Mathematics Group, Institute for Informatics and Telematics, CNR, a representative of CNR in the Scientific Committee of the Gran Sasso Consortium, and a member of the InterUniversity Mathematical School (SMI).

Karina Panucia Tillan Karina Panucia Tillan received the Laurea degree in Informatics from the Havana Polythenic Institute Jose Antonio Echeverria (CUJAE), Cuba, in 2005. She obtained the PhD degree in computer science from the International School in Information and Communication Technologies from the university of Modena and Reggio Emilia in 2014. From 2005 to 2007, she was a lecturer in Mathematics, Programming and Software Engineer Departments at Computer Science University (UCI), Cuba. Her main research interests are in bioinformatics, more specific: motif discovery problem.

REFERENCES

[1] S. Aerts, P. Van Loo, G. Thijs, Y. Moreau, and B. De Moor. Computational detection of cis -regulatory modules. *Bioinf.*, 19(suppl 2):ii5–ii14, 2003.

[2] W. Alkema, Ö. Johansson, J. Lagergren, and W. W. Wasserman. Mscan: identification of functional clusters of transcription factor binding sites. *Nucl. Acids Res.*, 32(Web-Server-Issue):195–198, 2004.

[3] T. L. Bailey and W. S. a. Noble. Searching for statistically significant regulatory modules. *Bioinformatics*, 19(suppl 2):ii16–ii25, 2003.

[4] A. F. Bardet, J. Steinmann, S. Bafna, J. A. Knoblich, J. Zeitlinger, and A. Stark. Identification of transcription factor binding sites from chip-seq data at high resolution. *Bioinformatics*, 29(21):2705–2713, 2013.

[5] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B*, 57(1):289 – 300, 1995.

[6] B. P. Berman, Y. Nibu, B. D. Pfeiffer, P. Tomancak, S. E. Celniker, M. Levine, G. M. Rubin, and M. B. Eisen. Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the drosophila genome. *Proceedings of the National Academy of Sciences*, 99(2):757–762, 2002.

[7] M. Burset and R. Guigo. Evaluation of gene structure prediction programs. *Genomics*, 34(3):353 – 367, 1996.

[8] Q. K. Chen, G. Z. Hertz, and G. D. Stormo. Matrix search 1.0: a computer program that scans dna sequences for transcriptional elements using a database of weight matrices. *Comp. appl. in the biosciences : CABIOS*, 11(5):563–566, 1995.

[9] M. Choi and Q. Zhou. Detecting clustering and ordering binding patterns among transcription factors via point process models. *Bioinformatics*, 2014.

[10] E. H. Davidson. *The Regulatory Genome: Gene Regulatory Networks In Development And Evolution*. Academic Press, 1 edition, 2006.

[11] S. B.-T. de Leon and E. H. Davidson. Gene regulation: Gene control network in development. *Ann. Rev. of Biophysics and Biomolec. Struct.*, 36(1):191–212, 2007.

[12] E. Eskin and P. A. Pevzner. Finding composite regulatory patterns in dna sequences. *Bioinformatics*, 18(suppl 1):S354–S363, 2002.

[13] M. Federico, M. Leoncini, M. Montangero, and P. Valente. Direct vs 2-stage approaches to structured motif finding. *Algorithms for Molecular Biology*, 7(1):20, 2012.

[14] M. C. Frith, U. Hansen, and Z. Weng. Detection of cis -element clusters in higher eukaryotic dna. *Bioinf.*, 17(10):878–889, 2001.

[15] M. C. Frith, M. C. Li, and Z. Weng. Cluster-Buster: Finding dense clusters of motifs in DNA sequences. *Nucl. Acids Res*, 31(13):3666–8, 2003.

[16] S. García, A. Fernández, J. Luengo, and F. Herrera. Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064, 2010.

[17] D. B. Gordon, L. Nekludova, S. McCallum, and E. Fraenkel. Tamo: a flexible, object-oriented framework for analyzing transcriptional regulation using dna-sequence motifs. *Bioinf.*, 21(14):3164–3165.

[18] T. Guns, H. Sun, K. Marchal, and S. Nijssen. Cis-regulatory module detection using constraint programming. In *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*, pages 363 –368, dec. 2010.

[19] J. v. Helden, A. F. Rios, and J. Collado-Vides. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Research*, 28(8):1808–1818, 2000.

[20] J. Hu, H. Hu, and X. Li. Mopat: a graph-based method to predict recurrent cis-regulatory modules from known motifs. *Nucleic Acids Research*, 36(13):4488–4497, 2008.

[21] J. Hu, B. Li, and D. Kihara. Limitations and potentials of current motif discovery algorithms. *Nucleic Acids Res*, 33:4899–4913, 2005.

[22] A. Ivan, M. Halfon, and S. Sinha. Computational discovery of cis-regulatory modules in drosophila without prior knowledge of motifs. *Genome Biology*, 9(1):R22, 2008.

[23] Ö. Johansson, W. Alkema, W. W. Wasserman, and J. Lagergren. Identification of functional clusters of transcription factor binding motifs in genome sequences: the mscan algorithm. *Bioinf.*, 19(suppl 1):i169–i176, 2003.

[24] A. Kel, E. Göß ling, I. Reuter, E. Cheremushkin, O. Kel-Margoulis, and E. Wingender. Matchtm: a tool for searching transcription factor binding sites in dna sequences. *Nucleic Acids Research*, 31(13):3576–3579, 2003.

[25] A. Kel, T. Konovalova, T. Waleev, E. Cheremushkin, O. Kel-Margoulis, and E. Wingender. Composite module analyst: a fitness-based tool for identification of transcription factor binding site combinations. *Bioinf.*, 22(10):1190–1197, 2006.

[26] K. Klepper, G. Sandve, O. Abul, J. Johansen, and F. Drabløs. Assessment of composite motif discovery methods. *BMC Bioinformatics*, 9(1):123, 2008.

[27] W. Krivan and W. W. Wasserman. A predictive model for regulatory sequences directing liver-specific transcription. *Genome Research*, 11(9):1559–1566, 2001.

[28] A. T. Kwon, A. Y. Chou, D. J. Arenillas, and W. W. Wasserman. Validation of skeletal muscle cis-regulatory module predictions reveals nucleotide composition bias in functional enhancers. *PLoS Comput Biol*, 7(12):e1002256, 12 2011.

[29] M. Leoncini, M. Montangero, and K. P. Tillan. Investigating power and limitations of ensemble motif finders using CE^3 metapredictor. *Current Bioinformatics*, to appear.

[30] L. Marsan and M.-F. Sagot. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *Journal of Computational Biology*, 7(3-4):345–362, 2000.

[31] A. Mathelier and W. W. Wasserman. The next generation of transcription factor binding site prediction. *PLoS Comput Biol*, 9(9):e1003214, 09 2013.

[32] A. A. Nikulova. Personal communication.

[33] A. A. Nikulova, A. V. Favorov, R. A. Sutormin, V. J. Makeev, and A. A. Mironov. Coreclust: identification of the conserved crm grammar together with prediction of gene regulation. *Nucl. Acids Res.*, 2012. doi: 10.1093/nar/gks235.

[34] P. Pavlidis, T. S. Furey, M. Liberto, D. Haussler, and W. N. Grundy. Promoter region-based classification of genes. In *Proceedings of the 6th Pacific Symposium of Biocomputing (PSB)*, pages 151–163, 2001.

[35] P. A. Pevzner and S.-H. Sze. Combinatorial approaches to finding subtle signals in DNA sequences. In *Proc. 18th Int. Conf. on Intelligent Systems for Mol. Biol.*, pages 269–278, 2000.

[36] N. Pisanti, A. M. Carvalho, L. Marsan, and M.-F. Sagot. Risotto: Fast extraction of motifs with mismatches. In J. R. Correa, A. Hevia, and M. A. Kiwi, editors, *LATIN*, volume 3887 of *Lecture Notes in Computer Science*, pages 757–768. Springer, 2006.

[37] D. S. Prestridge. Signal scan: a computer program that scans dna sequences for eukaryotic transcriptional elements. *Comp. appl. in the biosc.: CABIOS*, 7(2):203–206, 1991.

[38] H. Rouault, M. Santolini, F. Schweisguth, and V. Hakim. Imogene: identification of motifs and cis-regulatory modules underlying gene co-regulation. *Nucleic Acids Research*, 2014.

[39] A. Sandelin, W. Alkema, P. Engstrm, W. W. Wasserman, and B. Lenhard. Jaspar: an openaccess database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research*, 32(suppl 1):D91–D94, 2004.

[40] G. Sandve, O. Abul, and F. Drablos. Compo: composite motif discovery using discrete models. *BMC Bioinf.*, 9(1):527, 2008.

[41] R. Sharan, I. Ovcharenko, A. Ben-Hur, and R. M. Karp. Creme: a framework for identifying cis-regulatory modules in human-mouse conserved segments. *Bioinf.*, 19(suppl 1):i283–i291, 2003.

[42] S. Sinha. *Finding Regulatory Elements in Genomic Sequences*. PhD thesis, University of Washington, 2002.

[43] S. Sinha, E. van Nimwegen, and E. D. Siggia. A probabilistic method to detect regulatory modules. *Bioinf.*, 19(suppl 1):i292–i301, 2003.

[44] G. D. Stormo. Modeling the specificity of protein-dna interactions. *Quantitative Biology*, 1(2):115–130, 2013.

[45] J. Su, S. A. Teichmann, and T. A. Down. Assessing computational methods of cis-regulatory module prediction. *PLoS Comput Biol*, 6(12):e1001020, 12 2010.

[46] H. Sun, T. Guns, A. C. Fierro, L. Thorrez, S. Nijssen, and K. Marchal. Unveiling combinatorial regulation through the combination of chip information and in silico cis-regulatory module detection. *Nucleic Acids Research*, 40(12):e90, 2012.

[47] M. Thomas-Chollier, O. Sand, J. V. Turatsinze, R. Janky, M. Defrance, E. Vervisch, S. Brohee, and J. van Helden. RSAT: regulatory sequence analysis tools. *Nucleic Acids Research*, 36:W119–W127, 2008.

[48] W. Thompson, M. J. Palumbo, W. W. Wasserman, J. S. Liu, and C. E. Lawrence. Decoding human regulatory circuits. *Genome Research*, 14(10a):1967–1974, 2004.

[49] M. Tompa, N. Li, T. L. Bailey, G. M. Church, and et al. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23(1):137–144, 2005.

[50] T. Uno. Pce: Pseudo clique enumerator, ver. 1.0, July 2006.

[51] A. Valouev, D. Johnson, A. Sundquist, C. Medina, E. Anton, S. Batzoglou, R. Myers, and A. Sidow. Genome-wide analysis of transcription factor binding sites based on chip-seq data. *Nature Methods*, 5:829–834, 2008.

[52] P. Van Loo and P. Marynen. Computational methods for the detection of cis-regulatory modules. *Briefings in Bioinf.*, 10(5):509–524, 2009.

[53] W. W. Wasserman and J. W. Fickett. Identification of regulatory regions which confer muscle-specific gene expression. *Journal of Molecular Biology*, 278(1):167 – 181, 1998.

[54] T. Whitington, M. C. Frith, J. Johnson, and T. L. Bailey. Inferring transcription factor complexes from chip-seq data. *Nucleic Acids Research*, 39(15):e98, 2011.

[55] E. Wingender, P. Dietze, H. Karas, and R. Knüppel. Transfac: a database on transcription factors and their dna binding sites. *Nucl. Acids Res.*, 24(1):238–241, 1996.

[56] D. Xie, J. Cai, N.-Y. Chia, H. H. Ng, and S. Zhong. Cross-species de novo identification of cis-regulatory modules with gibbsmodule: Application to gene regulation in embryonic stem cells. *Genome Research*, 18(8):1325–1335, 2008.

[57] Y. Zhang and M. J. Zaki. EXMOTIF: efficient structured motif extraction. *Algorithms for Molecular Biology*, 1, 2006.

[58] J. Zhou, J. Sander, and G. Lin. Efficient composite pattern finding from monad patterns. *International Journal of Bioinformatics Research and Applications*, 3:86–99, 2007.

[59] Q. Zhou and W. H. Wong. Cismodule: De novo discovery of cis-regulatory modules by hierarchical mixture modeling. *Proc. Nat. Ac. of Sciences USA*, 101(33):12114–12119, 2004.