CMStalker: a combinatorial tool for composite motif discovery
Supplementary materials

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

July 2, 2014

Contents

A Technical definitions

Table 1 quickly summarizes all the main technical definitions used in this paper.

Name	Definition
$\mathcal{D} = \{\texttt{a},\texttt{c},\texttt{g},\texttt{t}\}$	DNA alphabet
$\mathcal{S} = \{S_1, \ldots, S_N\}$	Set of N sequences over $\mathcal{D}^*$
Transcription Factor (TF)	Protein that binds to specific stretches of DNA
Transcription Factor Binding Site (TFBS)	Specific stretch of DNA sequence which TFs bind to
Position Weight Matrices (PWM)	Matrix describing potential TFBSs of a TF
Oligonucleotide (oligo)	Short word in $\mathcal{D}^*$
DNA motif ((simple) motif)	TFBS model for just one TF
Motif mach (occurrence or hit)	Substring of a sequence in $\mathcal{S}$ represented by the motif (a potential binding site)
Motif class (used interchangeably with Transcription Factor or factor)	Set of motifs (describing a set of potential binding sites for a single TF)
Factor match	Motif match of any of the motifs in the class associated to that factor
Mapping alphabet $\mathcal{R}$	Arbitrary alphabet used to map single factors to symbols
Combinatorial group (group)	Set of motif classes having close-by matches in a large enough subset of $\mathcal{S}$ (represented by multisets over $\mathcal{R}$)
Group match	Set of factor matches, one for each motif class in a combinatorial group
Group quorum	Fraction of the set of sequences containing matches for the group
Group match width (span)	Given a sequence, the number of bps between the first and the last factor match of a group match.
Composite Motif (CM)	Set of motif classes having close-by matches

Table 1: Technical notions used/defined in the paper

B Table of Algorithms

Algorithm	Short	year	ref.	Target	Notes
CMStalker	CMS	2014	This paper	cm	Set model
COMPO	C	2008	[14]	cm	Set model
Cluster-Buster	CB	2003	[4]	cm	HMM model
CisModule	CM	2004	[18]	cm	HMM model
Cister	CI	2001	[3]	cm	HMM model
Composite Module Analyst	CMA	2006	[10]	cm	Set Model
MCAST	MC	2003	[2]	cm.	HMM model
ModuleSearcher	MS	2003	[1]	cm	Set Model
MSCAN	MSC	2003	[9]	cm	Set Model
Stubb	Stubb	2003	[15]	cm / crm	HMM model
MOPAT	M	2008	[7]	cm	Set model
CPModule	CP	2010	[6]	cm	Set model
CORECLUST	CoC	2012	[13]	cm	HMM model
D2Z-set	D2Z	2008	[8]	crm	Set model
CSam	CSam	2008	[8]	crm	Set model

Table 2: Table of cited algorithms. For each one we report the full name, the short name, year, reference, target output (either composite motifs (cm) or cis-regulatory modules (crm)), and the underlying mathematical model.

C Operations on multisets

In CMSTALKER multisets are represented as character sorted strings over the mapping alphabet (that we can easily regard as being totally ordered). This means, for instance, that both $\{a, b, a, c, c, a\}$ and $\{c, a, c, b, a, a\}$ are represented as *aaabcc*, given the standard lexicographic ordering of the characters.

Let $\nu_M(a)$ denote the number of occurrences of a in the multiset M. Then, for two multisets M and N we have $M \subseteq N$ if and only if, for any $a \in M$, $\nu_M(a) \leq \nu_N(a)$ holds.

The intersection I of multisets M and N includes all the elements a such that both $\nu_M(a) > 0$ and $\nu_N(a) > 0$ hold. Moreover, for any such element, $\nu_I(a) = \min\{\nu_M(a), \nu_N(a)\}$. As an example: $aaabcc \cap accd = acc$.

The difference D between multisets M and N includes all the elements a such that $\nu_M(a) > \nu_N(a)$. Moreover, for any such element, $\nu_D(a) = \nu_M(a) - \nu_N(a)$. As an example: $aaabcc \setminus accd = aab$.

D Numerical results

In this appendix we first report tables containing numerical values used to draw graphics in the main paper.

D.1 Numerical values for Figure 2 in Main Text

Dataset	CMSTALKER Two PWM files	CMSTALKER One PWM file
Dataset	Separated PWMs	Mixed PWMs
AP1-Ets	0.446	0.518
AP1-NFAT	0.109	0.106
AP1-NfkB	0.759	0.755
CEBP-NfkB	0.736	0.736
Ebox-Ets	0.587	0.587
Ets-AML	0.491	0.491
IRF-NfkB	0.917	0.917
NfkB-HMGIY	0.221	0.255
PU1-IRF	0.921	0.921
Sp1-Ets	0.203	0.203

Table 3: Numerical values for Figure 2 in Main TexT : (left) nCC when PWMs are given in separated files by TFs membership; (right) nCC when all PWMs in a file.

D.2 Numerical values for Figure 3 in Main Text

Dataset/Alg	CMS	C	CB	CI	MSC	MS	MC	Stubb	CMA	CM	M	CP	CoC
AP1-Ets	**0.52**	0.19	0.24	0.00	0.11	0.30	0.20	0.15	0.22	-0.03	0.23	0.20	
AP1-NFAT	0.11	0.06	0.04	0.00	0.00	0.05	0.14	-0.01	0.15	-0.02	**0.30**	0.14	
AP1-NFkB	**0.76**	0.59	0.49	0.19	0.36	0.29	0.26	0.35	0.55	0.05	0.06	0.27	
CEBP-NFkB	0.74	0.70	0.72	0.45	0.56	0.56	0.60	0.36	0.60	-0.03	0.40	**0.89**	
Ebox-Ets	**0.59**	0.55	0.16	0.26	0.44	0.20	0.23	0.14	0.18	0.05	0.31	0.24	
Ets-AML	**0.49**	0.42	0.30	0.07	0.31	0.38	0.26	0.23	0.33	0.03	0.26	0.22	
IRF-NFkB	**0.92**	0.73	0.77	0.62	0.91	0.85	0.41	0.41	0.69	0.04	0.61	0.73	
NFkB-HMGIY	0.26	0.31	0.35	0.10	0.30	**0.40**	0.23	0.07	0.15	-0.03	0.07	0.23	
PU1-IRF	**0.92**	0.28	0.16	0.27	0.00	0.43	0.16	0.17	0.24	-0.01	0.37	0.37	
Sp1-Ets	**0.20**	0.05	0.09	0.20	0.00	0.00	0.13	0.19	0.15	0.02	0.02	0.00	
Liver	0.49	0.56	**0.59**	0.31	0.51	0.42	0.50	0.48	0.36	-0.01	0.34	0.31	
Muscle	**0.56**	0.47	0.41	0.36	0.50	0.46	0.30	0.24	0.46	0.29	0.28	0.20	**0.56**
Avg nCC	**0.54**	0.41	0.36	0.24	0.33	0.36	0.29	0.23	0.34	0.03	0.27	0.32	

Table 4: Numerical values for Figure 3 in Main Text (nCC meaures), with best results reported in bold-face. Legend for algorithms: CMS=CMStalker, C = COMPO, CB = Cluster-Buster, CI = Cister, MSC= MSCAN, MS = ModuleSearcher, MC = MCAST, CMA = Composite Module Analyst, CM = CisModule, M=MOPAT, CP = CPModule, CoC = CORECLUST.

D.3 Numerical values for Figure 4 in Main Text

Datasets	Tool	PPV	Sensitivity	PC	ASP	CC
TRANSCompel	CMSTALKER	**0.67**	**0.54**	**0.42**	**0.60**	**0.58**
	COMPO	0.40	0.47	0.28	0.43	0.41
Liver	CMSTALKER	0.67	**0.43**	0.35	0.55	0.49
	COMPO	**0.85**	0.42	**0.55**	**0.64**	**0.57**
Muscle	CMSTALKER	**0.60**	0.65	**0.45**	**0.62**	**0.56**
	COMPO	0.52	**0.69**	0.42	0.60	0.52

Table 5: Nucleotide level numerical results for Figure 4 in Main Text.

E Comparing COMPO and CMStalker at increasing levels of noise

The authors of [11] provide additional PWM files for the TRASCompel datasets, which are characterized by an increasing number of "noise" (or "decoy") matrices. This is very similar to the approach already analyzed for the XIE benchmark (Section 5.4 in Main Text). In each file of the so-called *noise 50* benchmark, the authors include an equal number of good matrices and decoy matrices. For instance, the AP1-Ets PWM files include 33 matrices known to be associated with either AP1 or Ets Transcription Factors as well as additional 33 randomly selected TRANSFAC matrices that have no annotated binding sites in the AP1-Ets dataset. Similarly, in each file of the *noise 75* benchmark, there are three decoy matrices for each good one.

To make it possible to average over the possibly very different effects of the random selections, for each dataset and noise level the authors provide ten PWMs files that include different random selections from TRASNFAC matrices.

Figure 6 reports the results obtained by CMSTALKER and COMPO (by far the best performing tools, among those considered in this paper, on the TRANSCompel data) on input the PWM sets with 50% and 75% noise levels.

At level of noise 50% CMSTALKER leads on 7 data set over 10, as well as on average. At level of noise 75% The behavior of the two algorithms is quite similar (5 over 10 each) on the average (actually, a double average, since the results for each dataset are in turn the average of the results obtained on input the ten PMW sets with the corresponding noise level) but COMPO seems less sensitive (on these two benchmark) to noise level increase.

F Further Comparisons on the XIE et al. Benchmark

In Figure 1 we report the results obtained by CMStalker and the tools analyzed in [6] in a shorter range of noise matrices, namely from 10 to 40. We observe that, in this range, CMStalker is roughly equivalent to the two best performing methods (Cister, CisterBuster).

	Noise_50		Noise_75	
Dataset	CMSTALKER	COMPO	CMSTALKER	COMPO
AP1-Ets	**0.409**	0.200	**0.366**	0.157
AP1-NFAT	**0.098**	0.064	0.031	**0.064**
AP1-NFkappaB	0.354	**0.543**	0.108	**0.411**
CEBP-NFkappaB	**0.724**	0.699	**0.724**	0.699
Ebox-Ets	0.545	**0.548**	0.440	**0.488**
Ets-AML	**0.488**	0.428	0.294	**0.394**
IRF-NFkappaB	**0.917**	0.734	**0.842**	0.734
NFkappaB-HMGIY	0.086	**0.299**	0.031	**0.253**
PU1-IRF	**0.830**	0.255	**0.566**	0.244
Sp1-Ets	**0.184**	0.077	**0.135**	0.045
Average	**0.424**	0.385	**0.354**	0.349

Table 6: nCC values obtained by CMSTALKER and COMPO at noise level 50 and noise level 75 on the TRANSCompel benchmarks.

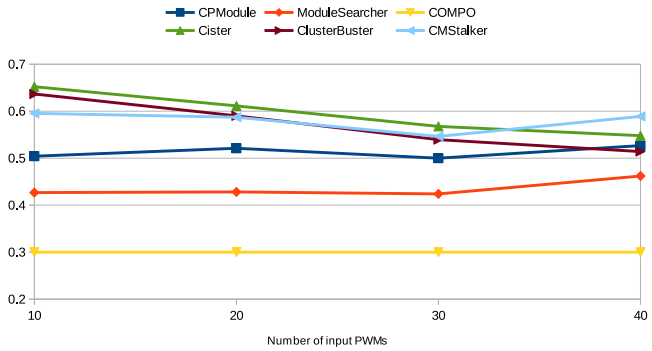

Figure 1: Nucleotide level CC results on the XIE benchmark.

G Experimental setup for MOPAT and CPModule

The results reported for the MOPAT[7] and CPModule[6] tools on the COMPOSITE benchmark were obtained according to the protocols described below.

G.1 MOPAT

For each of the twelve possible input datasets in the COMPOSITE benchmark, we run MOPAT with the following parameter setting: (1) windowsize = 50, (2) gene number = 50% of the input sequences, (3) pvalue = 0.001. Moreover, for each possible input PWM file, we performed ten runs. For the noise 50 and noise 75 sets of matrices, this amounts to 100 runs for each possible dataset (since, as descibed in Appendix C, there are 10 possible input PWM files for each noisy set, which are constructed by randomly sampling TRANSFAC matrices).

In certain cases MOPAT did not return any results. If, for a given dataset, MOPAT remained silent in 50% of the runs (or more), we discarded all the results for that dataset and proceeded by relaxing some input parameter. First we relaxed the window size (to 100 and 150, when required), then the pvalue (to 0.005), and last the gene number (actually, this was never necessary).

G.2 CPModule

We run CPModule using standard parameters. The crucial aspect that may influence the tool's performance, which we discuss here, is the choice of the background sequences used by CPModule to calculate an enrichment score of the motifs found in the input sequences.

We used different sets of background sequences. For the liver and muscle datasets we used the following ones:

Random.Hs.1000 a set of 1000 human promoter sequences (composed of 1100 bps each) sampled uniformly at random from the set of promoters in the Mammalian Promoter Database of the Cold Spring Harbor Laboratory [12];

Hs.Hg.18 a set of control sequences extracted from non-coding regions of the human genome [5] with lengths in the range 500-1000.

For the TRANSCompel datasets, we used ten additional background sequences, one for each dataset. More specifically, for any given dataset, say AP1-Ets, we formed the AP1-Ets.bkgd file by using those input sequences that were present in some other input file (e.g., IRF-NFkappaB.fasta) but not in AP1-Ets.fasta.

We note that the observed differences in the overall performance on the Transfac benchmark were minimal, if any, across the various background sets. In any case, the results reported in the paper are the ones obtained using the Random.Hs.1000 sequences, the most favourable to CPModule.

CPModule adopts the Any Number motif distribution model, as opposed to ZOOPS. However, we computed the statistics for both models, using the top scoring composite motif (i.e., the one with smallest p-value) for the latter. Once again, the results reported are the most favourable to CPModule, which were obtained precisely under the ZOOPS model.

H Measures used to assess the tool's output quality at the motif level

In [16] data is analyzed at the motif level by taking into account only the labels of the TF predicted to be part of the composite modules. The information relative to the position of the TFBS in the sequences is not used. Thus in this context, given the set of predicted TF and the set of true TF for each experiment:

mTP is the size of the intersection of the two sets of TF labels.

mFP is the size of the difference set of the predicted labels and the real labels.

mTN is the size of the complement of the union of the two sets of TF labels with respect to all the TF labels in Transfac.

mFN is the size difference set of the real labels and the predicted labels.

Starting from these basic classification all derived measures cited in Section 5.2 can be derived for the motif-level analysis. This type of analysis is adopted in [6].
In [17] the motif level analysis is defined differently, since it is based on the concept of a hit which involves the positions of the TSBS. A predicted TFBS hits a real TFBS when the length of their overlap is at least one quarter of the length of the real TFBS. In this context we can define:

mTP is the number of real TFBS with at least one hit.

mFP is the number of predicted TFBS that do not participate in any hit.

mFN is the number of real TFBS that do not participate in any hit.

In this context the notion of True Negative is problematic and for this reason no such definition is given in [17]. In our context, the difficulty in defining true negatives becomes evident when there are many input PWMs, the vast majority of which are not relevant to the experimental data at hand. Accounting for this very lage number of true negatives would clearly make the CC statistics less discriminant as the number of "irrelevant" matrices increases.

We can still use the, *Sensitivity* (Sn), *Positive Predicted Values* (PPV), *Performance Coefficient* (PC), and *Average Site Performance* (ASP) which do not depend on the value for TN. We cannot use the *correlation coefficient* (CC) which instead depends on the value of TN. In our measures at the motif level we uses the scheme in [17].

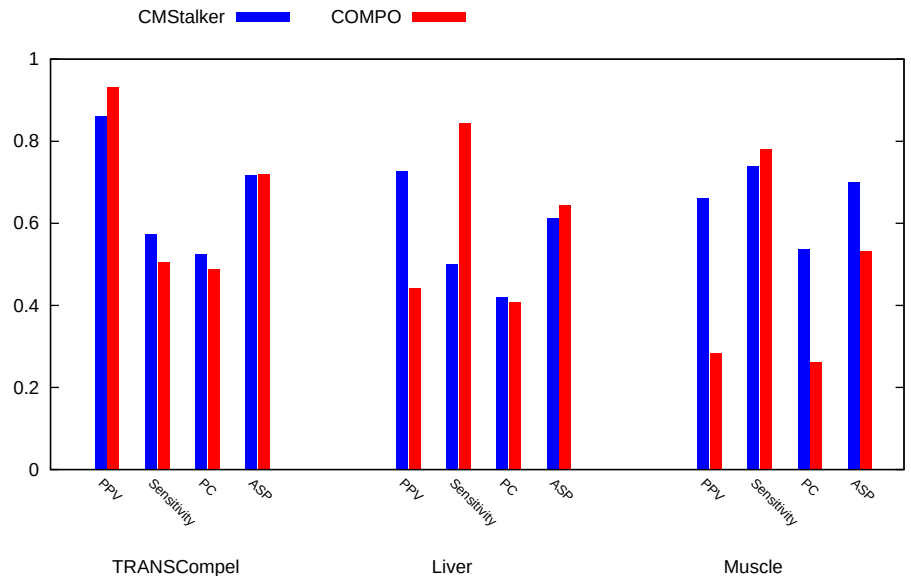

Figure 2: Motif level results for CMStalker and COMPO on TRANSCompel as well as liver and muscle datasets.

I Motif level analysis for CMStalker vs COMPO

We report here (Figure 2) the results of a comparison between CMStalker and COMPO using motif level statistics. The goal of such analysis is to evaluate prediction accuracy with respect to *module composition*, *i.e.*, the ability of a tool to correctly tell the matrices (possibly within an equivalence set) whose matches belong to the predicted modules in any given sequence/gene. As already pointed out above, we do not base our comparisons on statistics that involve the notion of true negative predictions. Hence, we only consider Sensitivity, Positive Predicted Value, Performance Coefficient, and Average Site Performance as motif level statistics.

Figure 2 shows the results obtained, which seem to suggest an essentially comparable behavior at the motif level between the two algorithms (perhaps with the exception of the muscle dataset, where CMStalker leads in both PC and ASP).

I.1 Numerical values for Figure 1 in Suppl. Materials

Datasets	Tool	PPV	Sensitivity	PC	ASP
TRANSCompel	CMSTALKER	0.861	**0.574**	**0.525**	**0.718**
	COMPO	**0.932**	0.506	0.488	**0.719**
Liver	CMSTALKER	**0.727**	0.500	**0.421**	0.613
	COMPO	0.443	**0.844**	0.409	**0.643**
Muscle	CMSTALKER	**0.661**	0.740	**0.536**	**0.700**
	COMPO	0.283	**0.780**	0.262	0.531

Table 7: Motif level numerical results for Figure 2.

J Motif level analysis for CMStalker on on Xie et al. Data

Figure 3 shows the motif level statistics for the results computed by CMStalker. As expected, as the number of input PWMs increases, we observe a negative trend of the PC and ASP statistics.

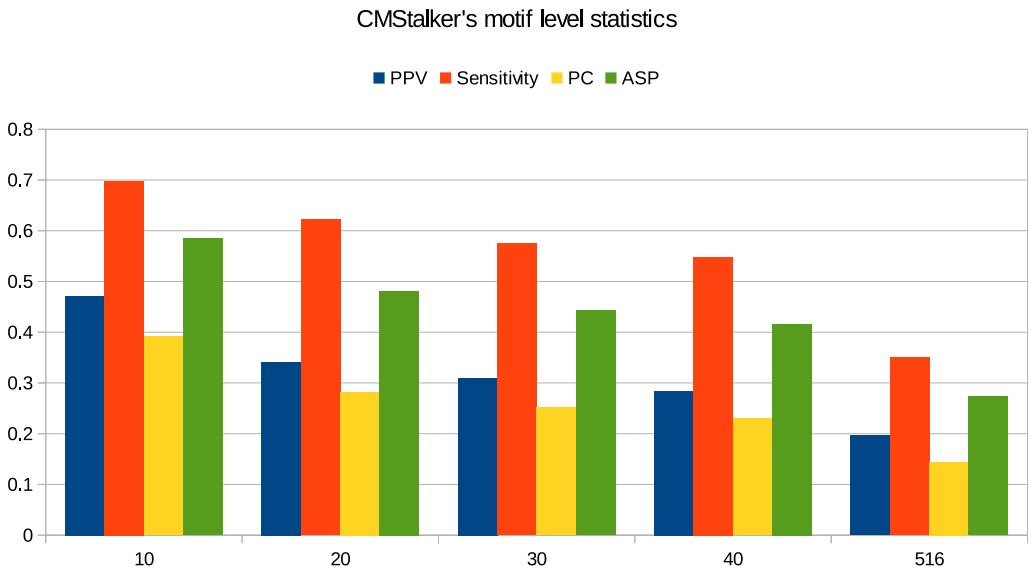

Figure 3: Motif level statistics for CMStalker on XIE benchmark.

K Hardware and Sofware

The code for CMStalker, CPModule and MOPAT have been executed on an Apple Mac Mini with Intel core i7 processor (4 cores) at 2.6GHz, with 16Gb RAM memory. The Operating System is Mac OS X 10.8.5. The HD holds 1Tb (with 128SSD). The software has been developed in Python 2.7 (with some external calls to Perl code).

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

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

[4] 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.

[5] H. Z. Girgis and I. Ovcharenko. Predicting tissue specific cis-regulatory modules in the human genome using pairs of co-occurring motifs. *BMC Bioinformatics*, 13(25), 2012.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] C. S. H. Laboratory. Comprehensive regulatory element analysis and discovery.

[13] 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.

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

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

[16] 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.

[17] 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.

[18] 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.