

Integrative Methods for Discovering Generic *Cis*-Regulatory Motifs

Thesis

Submitted for the degree of
Doctor of Philosophy

Edward WIJAYA
(*MSc, LSE U.K.*)

School of Computing
National University of Singapore
2008

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Dr. Sung Wing-Kin for his guidance and countless insightful suggestions throughout my research. Also through him I learnt about the importance of pursuing excellence rather than settling for mediocrity in research. I will work hard to live to your aspiration throughout my future research.

My heartfelt gratitude to Dr. Kanagasabai Rajaraman, whom in the first place took me as his student. I am grateful to him for his patience with my shortcomings and his enlightening advices for me throughout my Ph.D. work.

I would also like to extend my sincere thanks to our collaborator Dr. Siu Yiu-Ming from Hongkong University for his continued guidance, encouragement and support, particularly at many critical junctures in my research. I am also grateful to my committee members Dr. Leong Hon Wai and Dr. Anthony Tung for providing advices and suggestions throughout my thesis proposal.

I would also like to thank my friends whom have helped me in research and technical discussion: Ngo Thanh Son, Hendra Setiawan, SPT Krishnan, and Jose Martinez.

My thanks to my parents and aunt Martha, for giving me support at the critical points of my work. At last, my eternal gratitude to my wife Yumiko for her steadfastness and patience in times of difficulties, especially in taking care of our children when I was not around.

Summary

One of the important problems in molecular biology is to understand the mechanisms that regulate the expressions of genes. A crucial step in this challenge is the ability to identify *cis*-regulatory motifs, e.g. binding sites in DNA sequences. Studying them can give us important clues in unraveling regulatory interactions of genes. The prediction of such regulatory elements is a problem where computational methods offer a great hope.

This thesis presents a new class of algorithms for *in silico* discovery of regulatory elements. Firstly, we address the problem of motif finding for generic spaced motifs. Spaced motifs, an important class of transcription factors binding sites, consists of several short segments separated by spacers of different lengths. Existing motif finding algorithms are either designed for monad motifs or have assumptions on the spacer lengths or can handle at most two segments. To address this issue, we propose a new method called **SPACE**. The key idea is to obtain the motif as an integration of the submotifs as defined by the frequent pattern.

Our method makes use of a novel scoring technique to measure the statistical significance of generic spaced motifs. With this measure we overcome the difficulty in handling biased samples by incorporating background sequence from

various *species*. Based on experiments on real biological datasets and Tompa’s benchmark datasets, we show that our algorithm outperforms the existing tools for spaced motifs in both sensitivity by 20.3% and specificity by 76%. And for monads, it performs as well as other tools.

Secondly, although many tools have been developed for motif finding, they vary in their definitions of what constitute a motif and in their methods for finding statistically overrepresented motifs. There is no clear way for biologist to choose the motif finder that is most suitable for their task. There is an immediate need for a more effective method that allows the biologist to make use of these diverse motif finders for finding novel transcription factor binding sites accurately. However there are two main difficulties in this direction. First, multiple motif finders may report similar spurious motifs. The challenge lies in how to distinguish these spurious motifs from the real overrepresented motifs. Second, even if the reported motif can approximate the real motif, they still contain false positive that have high similarity with the real binding sites. For this reason, we propose a method called **MotifVoter** to identify regulatory sites by integrating results found by multiple motif finders. It applies a variance based statistical measure to remove the spurious motifs and then refines the prediction by filtering the noisy binding sites using a novel voting scheme. We show that these two steps help to overcome the two difficulties by removing spurious predictions at both motif and binding site levels. Validation of our method on Tompa’s benchmark, real *metazoan* and *E. Coli* datasets (186 datasets in total) show that it can improve the sensitivity by 120% and precision by 77% over stand alone motif finders. MotifVoter can locate almost all the binding sites found by the individual motif finders used and is able to distinguish the real binding sites from noise effectively.

We conclude that our integrative approach towards motif finding offers a practical alternative for biologists to study novel regulatory sites.

Publications and Softwares

Publications

- **Edward Wijaya**, Siu-Ming Yiu, Ngo Thanh Son, Kanagasabai Rajaraman and Wing-Kin Sung, MotifVoter: a novel ensemble method for fine-grained integration of generic motif finders, *Bioinformatics*, 24(20):2288-2295, 2008.
- **Edward Wijaya**, Kanagasabai Rajaraman, Siu-Ming Yiu and Wing-Kin Sung, Detection of Generic Spaced Motifs Using Submotif Pattern Mining, *Bioinformatics*, 23(12):1476-1485, 2007.
- Bijayalaxmi Mohanty, Balasubramanian Ashok, and **Edward Wijaya**, Modelling and detection of transcription termination sites of genes induced during low oxygen response in Arabidopsis, in *Proc. 9th Conference of the International Society for Plant Anaerobiosis*, 2007.
- **Edward Wijaya**, Kanagasabai Rajaraman and Wing-Kin Sung, Detection of Regulatory Elements using Constrained Submotif Pattern Mining, in *6th Singapore-Korea Joint Workshop on Bioinformatics Invited Seminar*, February 12th 2007.
- **Edward Wijaya** and Kanagasabai Rajaraman, Identification of spaced regulatory sites via submotif modeling, in *Proc. 3rd RECOMB Workshop on Regulatory Genomics*, 2006.
- **Edward Wijaya**, Kanagasabai Rajaraman and Manisha Bramahchary, A Hybrid Algorithm for Motif Discovery from DNA Sequences, *3rd Asia-Pacific Bioinformatics Conference - Satellite Symposium and Poster*, 2005.

Softwares

In conjunction with the works presented in this thesis. The following softwares have been made available as webservers for public use:

- **SPACE** available at:

<http://www.comp.nus.edu.sg/~bioinfo/SPACE-Web>

This webserver allows users to find generic spaced motifs, by online submission of FASTA sequences. Result will be dispatched through email.

- **MotifVoter** available at:

<http://www.comp.nus.edu.sg/~bioinfo/MotifVoter>

This webserver implements ensemble motif finding proposed in Chapter 3 of the thesis. It allows user to perform online submission of FASTA sequences and select their preferred component motif finders. Result will be dispatched through email.

Contents

Acknowledgements	i
Summary	ii
Publications and Softwares	v
Nomenclature	ix
List of Tables	xi
List of Figures	xiii
1 Introduction	1
1.1 Biological Background	2
1.1.1 Gene Regulation	2
1.1.2 <i>Cis</i> -Regulatory Elements	3
1.1.3 Role of Transcription Factor in Gene Regulation	3
1.1.4 Challenges in the Discovery of Regulatory Motifs	5
1.2 Literature Review	7
1.2.1 Motif Models	7
1.2.2 <i>De novo</i> Motif Finders	10
1.2.3 Methods Using Genomical Data	24
1.2.4 Motif Evaluation and Benchmarks	25
1.3 Motivations	27
1.3.1 Challenges from Real Biological Data	27
1.3.2 Challenges from Current Practice	28
1.4 Contributions of the Thesis	29
1.5 Organization of the Thesis	31
2 Detection of Generic Spaced Motifs Using Submotif Pattern Mining	32
2.1 Generation of Motif Candidates	38

2.2	Refining Motif Candidate into Spaced Motif	39
2.3	Significance Testing and Scoring	41
2.4	Efficient Generation of Motif Candidates	43
2.5	The Final Ranking of Motifs in SPACE	45
2.6	Experimental Results	47
2.6.1	Results on Datasets with Spaced Motifs	48
2.6.2	Results on Datasets with Monad Motifs	65
2.7	Conclusions	76
3	Variance Based Ensemble Method for Integrating Generic Motif Finders	77
3.1	Performance of Individual Motif Finders with the Inclusion of Lower Rank Motifs	81
3.2	Different Motif Finders Discover Different Binding Sites	83
3.3	MotifVoter - A Method That Utilizes the Sites Predicted by Multiple Motif Finders	84
3.4	Pairwise Similarity Between Motifs	85
3.5	Motif Filtering	86
3.6	Heuristics Used in MotifVoter	88
3.7	Instance Refinement	89
3.8	Position Weight Matrix (PWM) Generation	91
3.9	Experimental Results	91
3.9.1	The performance of MotifVoter versus individual motif finders	91
3.9.2	Performance of MotifVoter on Different Background Sequences and Species.	95
3.9.3	Time Complexity of MotifVoter	96
3.9.4	Robustness of MotifVoter	99
3.9.5	Validation on Metazoan Datasets	101
3.9.6	Comparison of MotifVoter with Other Ensemble Methods .	105
3.10	Effect of Discriminative and Constraint Attributes	118
3.11	Observations on the Binding Sites Missed by MotifVoter	119
3.12	Conclusion	121
4	Conclusion and Future Directions	123
4.1	Conclusion	123
4.2	Future Directions	125
	References	127
	Appendix	139

Nomenclature

L	motif length
ℓ	submotif length
d	number of mutations
q	quorum
$Z[i..j]$	substring of Z starting at position i and ending at position j
$len(s_i)$	length of i -th sequence s_i
$hd(x, y)$	Hamming distance of two equal-length strings x and y
$E(M, e)$	expected frequency of motif M with at most e mutations
$\beta(M)$	occurrence score of motif M
$\sigma(M)$	sequence-specific score of motif M
$sim(x, y)$	similarity between motif x and y
$I(x)$	set of regions covered by the instances of motif x
$I(x) \cap I(y)$	set of regions covered by at least one instance in x and y
$I(x) \cup I(y)$	set of regions covered by any instance of x or y
m	number of component motif finders
n	number of top- n motifs reported by a component motif finder
P	a set of candidate motifs from m motif finders ($P = mn$)
X	a subset of candidate motifs of P
$w(X)$	similarity score of candidate motifs in X
$A(X)$	variance score of X
PPV	positive predictive value
SN	sensitivity
CC	coefficient correlation
PC	performance correlation

Degenerate Consensus Symbols

M	A or C
R	A or G
W	A or T
S	C or G
Y	C or T
K	G or T
B	C,G or T
D	A,G or T
H	A,C or T
V	A,C or G
N	A,C,G or T

List of Tables

2.1	Comparison of SPACE, MITRA and BioProspector on spaced motifs in real biological datasets	49
2.2	Comparison of SPACE, MEME and Weeder on real spaced biological data where motif contain spacers	53
2.3	Performance of SPACE, MITRA and BioProspector (denoted BP) on 4 types of synthetic data (one dataset each).	58
2.4	Comparison of SPACE and MITRA averaged performance on 4 motif finding problems.	59
2.5	Comparison of SPACE and BIOPROSPECTOR averaged performance on 4 motif finding problems.	59
2.6	Detailed comparison of SPACE and MITRA performance on 3 motif finding Test Case 1.	60
2.7	Detailed comparison of SPACE and MITRA performance on 3 motif finding Test Case 2.	60
2.8	Detailed comparison of SPACE and MITRA performance on 3 motif finding Test Case 3.	60
2.9	Detailed comparison of SPACE and MITRA performance on 3 motif finding Test Case 4.	60
2.10	Detailed comparison of SPACE and BIOPROSPECTOR performance on 3 motif finding Test Case 1.	61
2.11	Detailed comparison of SPACE and BIOPROSPECTOR performance on 3 motif finding Test Case 2.	61
2.12	Detailed comparison of SPACE and BIOPROSPECTOR performance on 3 motif finding Test Case 3.	61

2.13	Detailed comparison of SPACE and BIOPROSPECTOR performance on 3 motif finding Test Case 4.	61
2.14	Comparison of SPACE and MEME averaged performance on 4 motif finding problems.	62
2.15	Comparison of SPACE and WEEDER averaged performance on 4 motif finding problems.	62
2.16	Detailed comparison of SPACE and MEME performance on 3 motif finding Test Case 1.	63
2.17	Detailed comparison of SPACE and MEME performance on 3 motif finding Test Case 2.	63
2.18	Detailed comparison of SPACE and MEME performance on 3 motif finding Test Case 3.	63
2.19	Detailed comparison of SPACE and MEME performance on 3 motif finding Test Case 4.	63
2.20	Detailed comparison of SPACE and WEEDER performance on 3 motif finding Test Case 1.	64
2.21	Detailed comparison of SPACE and WEEDER performance on 3 motif finding Test Case 2.	64
2.22	Detailed comparison of SPACE and WEEDER performance on 3 motif finding Test Case 3.	64
2.23	Detailed comparison of SPACE and WEEDER performance on 3 motif finding Test Case 4.	64
2.24	Comparison of SPACE, MITRA and Weeder on monads in real biological datasets	69
2.25	Comparison of SPACE, MITRA and BioProspector on real monad biological data	72
3.1	Average sensitivity and precision (nPPV) of each motif finder in <i>E.Coli</i> and Tompa dataset.	108
3.2	Low binding sites density will have higher percentage of missed binding sites	121
3.3	Higher binding sites density will have lower percentage of missed binding sites	121

List of Figures

1.1	A transcription factor binds upstream of a gene	4
1.2	CTCF motif.	5
1.3	An investigative paradigm to infer regulatory interaction	6
1.4	A consensus model inferred from five occurrences of a motif	8
1.5	A PWM model inferred from five occurrences of a motif	9
1.6	Classification table for stand alone <i>de novo</i> motif finders.	10
1.7	Similarity between instances is modeled using graph	12
1.8	Three different states of HMM to model a set of instances	18
1.9	Performance of motif finders in Tompa's benchmark dataset	29
2.1	Example of length-20 spaced motif with three segments	34
2.2	GAAGAnnnnnnnTAGAAAnn is a spaced motif of the above 5 sequences	36
2.3	Since the number of occurrences is at least q , it is a motif candidate	38
2.4	Note that $\{1, 13, 14\}$ is the frequent itemset which appears 4 times	40
2.5	Comparison of MITRA, BioProspector and SPACE	57
2.6	Comparison between SPACE with 13 other motif discovery tools	65
2.7	Binding sites without gaps reported by SPACE in hm17g (<i>human</i>)	66
2.8	Comparison of SPACE and best performing algorithms on 4 types of organisms	67
3.1	MotifVoter's approach.	79
3.2	Accumulative performance of 10 individual motif finders	82
3.3	Different motif finders discover different binding sites	84
3.4	Comparison of MotifVoter and individual motif finders on Tompa's Benchmark dataset	93

3.5	The sensitivity of MotifVoter versus the maximum possible sensitivity (using 10 selected motif finders)	94
3.6	Performance of MotifVoter on various types of background sequences	95
3.7	The performance of MotifVoter on various species	96
3.8	Running time of 10 motif finders on 1.5KB dataset	97
3.9	Running time of heuristic with respect to changes in m and n	98
3.10	he performance of MotifVoter when we use 10 motif finders together with 1-5 random motif finders	98
3.11	Performance of MotifVoter based its N fastest basic motif finders	100
3.12	Contribution of component motif finder to the output given my MotifVoter	100
3.13	Upper bound analysis on Metazoan datasets	102
3.14	Comparison of MotifVoter and individual motif finders on metazoan dataset	103
3.15	Performance of MotifVoter on various species in <i>metazoan</i> dataset	103
3.16	Examples of the binding sites found by MotifVoter and stand-alone motif finders on real metazoan datasets	104
3.17	Comparison of MotifVoter with SCOPE and EMD	107
3.18	Evaluation of MotifVoter with other stand-alone motif finders in <i>E.Coli</i> dataset.	107
3.19	Comparison on yeast ChIP [54] experiments, with BEST, Web-motifs and SCOPE in terms of predicting percentage of correct motifs	109
3.20	Binding sites comparison of MotifVoter on <i>yeast</i> ChiP experiments.	110
3.21	Binding sites comparison of MotifVoter on mammalian ChiP experiments.	116
3.22	Importance of discriminative measure and constraint.	118

CHAPTER 1

Introduction

Since the dawn of 21st century, genomic research has entered a new era, due to the introduction of high-throughput experiments in molecular biology [76]. Large scale genomics became an important tool for understanding the organisms. Access to these genomic sequences helps biologists to define and test hypotheses about how genomes are organized and evolved, as well as how a genome encodes the observed properties of a living organisms. The major questions being pursued include: what parts of our genome encode the mechanisms for major cellular function like metabolism, differentiation, proliferation, and programmed death? How do multiple genes act together to perform specialized functions? How is our non-protein coding DNA organized, and which parts are functionally important? How do selective pressures act on the random processes of gene duplication and mutation to give rise to complex organs like eyes, wings and brains? Why do humans appear so different from worms and flies, despite sharing so many of the same genes?

Nevertheless, the task of discovering the function of these genomic sequences is expensive and time consuming. Given the wealth of sequence data nowadays, functional analysis in the wet lab can only be applied to a small percentage of

new data.

On the other hand since genomic sequence data has been accurately represented in a database, this provides an opportunity for computer scientists. Computationally aided analysis can provide insight into the function to the genes, both by analyzing the genes themselves or by comparing similarities of genes of other organisms. Computational analysis of genomic sequence may never replace the wet lab techniques of the molecular biologist. However, by mining statistically significant trends from genomic data, the computer scientists can direct the attention of molecular biology, uncovering biologically significant functional information that might otherwise remain undiscovered.

It is within this framework of genomic sequence analysis our thesis work is situated.

1.1 Biological Background

1.1.1 Gene Regulation

Most cells of a multi-cellular organism contain all genetic information at all times, but only a fraction of it is active. We are only beginning to understand how do cells determine the active state of its component [32, 108]. About 10% of human and fruitfly genes are estimated to be used only to control the expression of other genes [1, 140]. Understanding the regulation of gene expression is therefore undoubtedly one of the most interesting challenges in molecular biology today.

To express a gene, control mechanisms appear at many different levels. The most important control level is the first step of gene expression which produce the primary transcript RNA. The primary transcript RNA eventually goes through RNA processing to generate messenger RNA.

Complexes of transcription factors, RNA polymerase complex and other proteins bind to regulatory DNA regions called *promoters* of genes. It is intuitively clear that errors occurring in this machinery leading to false-expression of genes that are important link to genetically based diseases [76]. It is thus important to find exact regulatory regions to be able to examine in detail, either computationally or by experiments, and learn the mechanism that control the expression of genes.

1.1.2 *Cis*-Regulatory Elements

Regions of DNA or RNA that regulate the expression of genes are called *cis*-regulatory elements [30, 108]. These elements are often binding sites of one or more trans-acting factors. There exist many categories of *cis*-regulatory elements [97]. The most important is the class of *transcription factor binding sites* (TFBS). These are short DNA sequence patterns that are targeted by specific auxiliary proteins called *transcription factors* [76].

There are many other examples of motifs including motifs in enhancers, ribosome and splicing sites [71]. For a more complete discussion on cellular regulatory mechanism, we refer to standard books on this topic, e.g. [19, 76]. For illustration, we consider the transcription factor binding sites (TFBS) as an exemplar of regulatory motifs, in the next subsection.

1.1.3 Role of Transcription Factor in Gene Regulation

The study of transcriptional regulation is crucial to our understanding of the cell. Whether it is a routine function which controls a cell to grow and replicate, or the information processing and response mechanism that are deployed by the cell

to deal with external stimulus, transcriptional regulation is heavily utilized as the building block of elaborate cellular mechanism [142].

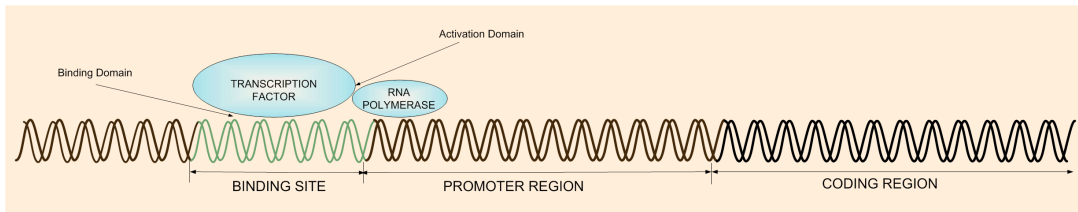


Figure 1.1: A transcription factor binds upstream of a gene and activates the RNA polymerase, thereby promoting transcription initiation.

The most common mechanism to regulate the gene expression operates at the stage of transcription. For a mRNA transcript to be produced from a gene template, the RNA polymerase complex first needs to be recruited near the 5' end of the gene, at a position called TSS. In some cases, a protein molecule binds to the DNA near the TSS, and then interacts with the RNA polymerase complex, either inducing or inhibiting the latter's DNA-binding capacity (See Figure 1.1). It is easy to see that such DNA-binding molecule would then have the ability to either promote or suppress gene expression, by affecting the recruitment of RNA polymerase. These molecules are called *transcription factors*, and there are many distinct proteins that serve as transcription factors in the cell. Sometimes, a transcription factor interacts with other proteins (including other transcription factors), influencing transcription indirectly. It has two important domains in its structure - the DNA binding domain, which is often specialized to recognize a very specific DNA specific sequence, and the activation domain, which interacts with the RNA polymerase or other proteins. The DNA binding domain can recognize and bind specific target sites that are located near a gene, "switching" the gene on or off, without directly affecting the expression of other genes. The

regulated gene may itself code for another transcription factor, which in turn regulates another gene.

The DNA-binding domain of a transcription factor is specialized to recognize a very specific target site in the DNA. These transcription factor binding sites (or “regulatory elements”) range between 6 and 25 bp in length. Usually the bases at all positions in the site are not equally important for binding specificity. A *motif* is a characterization of the binding sites of a transcription factor. For example, a well known transcription factor CTCF has CCGCGnGGnGGCAG as its motif [69] (see Figure 1.2). The transcription factor has high affinity for sequences that exactly or approximately match the motif while relatively low affinity for sequences different from the motif.



Figure 1.2: CTCF motif.

The study of transcription factor binding sites can give us important clues in unraveling regulatory interactions of genes. Once the motif of the binding sites of a transcription factor is known, it enables one to look for occurrences of this motif in promoters of other genes. The presence of motif is circumstantial evidence that the gene is regulated by the transcription factor.

1.1.4 Challenges in the Discovery of Regulatory Motifs

We outline the motif discovery problem from the setting of the transcription factor binding sites. We start with the hypothesis that a set of genes is regulated by the same transcription factor (*co-regulated*). We can then look at the interesting

motifs that are shared by promoters of these genes. If any such motif is found, we can experimentally verify if there exist a transcription factor that has high specificity for the motif, and if so, that transcription factor is a potential regulator of the set of genes that we started with. This kind of paradigm is the most relevant application scenario for the work we are presenting. Figure 1.3 depicts the schematic flow of these steps.

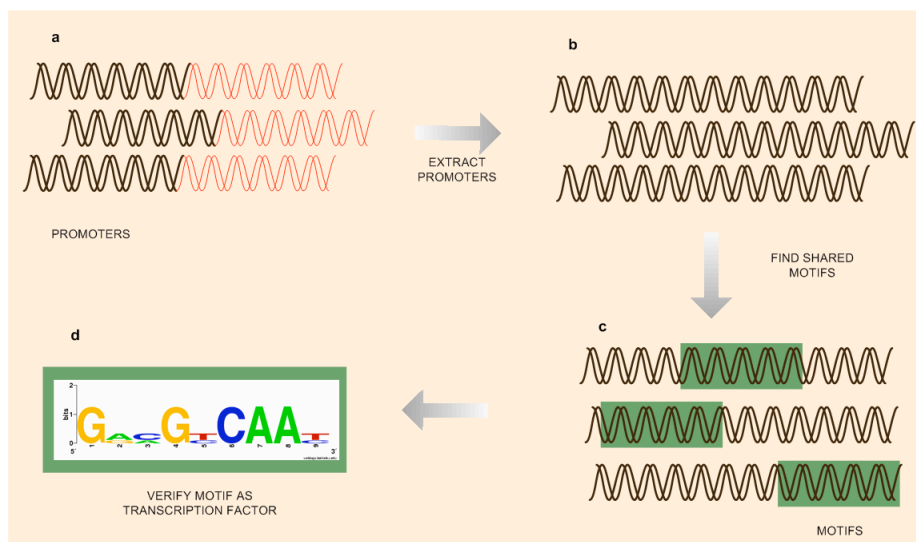


Figure 1.3: An investigative paradigm to infer regulatory interaction. (a) Begin with a set of potentially co-regulated genes, (b) Extract the promoter sequences of these genes, (c) Identify interesting motifs shared by promoters, (d) Experimentally verify if detected motifs are specifically bound by any transcription factor

Motif-finding in general is a difficult problem, and the one that is not yet well-solved [104, 135]. There are several reasons for the difficulty:

1. As shown by Ming Li [77] and Litman [45] the motif finding problem is inherently NP-hard.
2. There can be a great variability in the binding sites of a single factor, and the nature of the allowable variations is not well understood. Depending on

the model of variability that we assume binding sites, the space of possible motifs to be searched can be very large.

3. There may be multiple binding sites for a single factor in a single gene's regulatory region. The regulatory elements are not always the same orientation as the coding sequence or each other.

1.2 Literature Review

In this section we will first describe two general classes of motif models used by existing motif finders. Subsequently, we will elaborate on representative motif finders for the respective models.

1.2.1 Motif Models

Consensus Model

The consensus model is a simple combinatorial description of a motif. In this model, the motif is represented as consensus sequence. Each occurrence (instance) of the motif is a copy of the consensus sequence, perhaps with a few substitutions. The consensus at each position of multiple sequences is defined as the base which occurs most often at the position. In the case that two or more bases have equal highest occurrence, the consensus will be represented by IUPAC symbol. Figure 1.4. below illustrate the example of consensus model.

One could measure the conservation of a motif by the number of substitutions between each occurrence and the consensus sequence.

Consensus model is somewhat a simpler model. Given multiple occurrences, it extract a single pattern - consensus sequence. In most cases, it is effective in

5 occurrences of a motif	Consensus Sequence
CATCAAT	
TGCTAAT	
TGTACAT	TGTwAAT
TGGCACT	
TGTTGAT	

Figure 1.4: A consensus model inferred from five occurrences of a motif. The most frequent base in each position of the occurrences becomes the base of the consensus at the position. If two or more bases appear equally often in a given position, as with T and C in the fourth position, the consensus is represented with IUPAC symbol w.

the sense that the base that appears most frequently in each position has the highest likelihood to be original base of the motif. However, consensus model risks missing the actual motif. This happens in the situation that the base at any position of the motif is weakly conserved.

Position Weight Matrix (PWM) model

The consensus model is not informative, because it does not reveal either how strongly the consensus base in each position is conserved or the distribution of non-consensus bases. However, all this information are described in the weight matrix model (PWM), also called profile model. PWM is a probabilistic model. It models a motif of length l as a $4 \times l$ matrix W , where the entry at position $W[p, q]$ gives the probability that an occurrence of the motif contains a base p ($p = \{\mathbf{A}, \mathbf{T}, \mathbf{C}, \mathbf{G}\}$) in its q -th position. Each column of the matrix therefore sums to one as illustrated in Figure 1.5. The distribution of bases in different positions are independent of each other. Given a length- l sequences, let $s[i]$ denotes the base at its i -th position. Based on the weight matrix, the probability that M produces a particular length- l instance m is: $Pr[m|W] = \prod_{i=1}^l W[m[i], i]$. Given a set of motif occurrences M , the weight matrix $W[M]$ can be easily computed by

calculating the frequency of each base in each position.

5 occurrences of a motif	Position Weight Matrix							
CATCAAT								
TGCTAAT								
TGTACAT								
TGGCACT								
TGTTGAT								
	1	2	3	4	5	6	7	
A	0	0.2	0	0.2	0.6	0.8	0	
C	0.2	0	0.2	0.4	0.2	0.2	0	
G	0	0.8	0.2	0	0.2	0	0	
T	0.8	0	0.6	0.4	0	0	1	

Figure 1.5: Unlike the consensus model, the PWM captures the frequencies of both consensus base and non-consensus bases, and it remains well-defined even when the consensus base is ambiguous at 4-th position.

The matrix $W[M]$ is the best description of M in the sense of maximum likelihood. It is the weight matrix W that maximizes the likelihood $L[W[M]|M] = \prod_{m \in M} Pr[m|W]$. And the likelihood $L[W[M]|M]$ is also a useful score by which to measure the extent of conservation of the motif. If the motif occurs in random background sequences with a base distribution P , then the scoring function for the set M of motif occurrences is the likelihood ratio $LR(M)$, defined as:

$$LR(M) = \frac{L[W[M]|M]}{L[P|M]}$$

where

$$L[P|M] = \prod_{m \in M} Pr[m|P]$$

While the likelihood ratio is not, strictly speaking, a measure conservation, it represents a principled way to account for the background base distribution when scoring a motif.

Since PWM motif model captures the frequency of each base in each position. It will best describe the motif (M) in the sense of maximum likelihood. In addition, the impact of the background distribution can be taken into account

for measuring the conservation.

Instead of extracting a specific motif, a PWM provides only information to infer the likelihood that any length l -string is the actual motif. It is possible that the model is biased on wrong bases in some positions in the situation that the mutations occur preferentially on a small subset of positions of its occurrences. To overcome this issue, some works have tried to makes use of mixture of several PWMs that capture different information sources [4, 53] or incorporating some forms of weighted measure into the base counting procedure [124]. Finally, to get the model that best reflect the actual motif, the initial model need to be refined using one of the following probabilistic methods derivatives: *Expectation-Maximization*, *Gibbs Sampling* and *Hidden Markov Model*.

1.2.2 *De novo* Motif Finders

In this section we describe *de novo* methods that use the above motif models. Although the division is clear for most algorithms, there also exist methods that try to combine both methodologies. Figure 1.6 depicts the general overview of the classification for de novo motif finders.

Consensus		PWM			Hybrid
Graph	Enumeration	Expectation Maximization	Gibbs Sampling	Hidden Markov Model	
MotifCut SP-STAR Trawler Winnower cWinnower	Consensus MITRA Oligo-Dyad QuickScore SMILE TEIRESIAS Weeder YMF	Dragon Motif Builder Improbizer MEME Ortho-MEME Random Projection	ANN-Spec AlignACE BioProspector GLAM GibbsILR MotifSampler SeSiMCMC	YEBIS	HMD MDScan

Figure 1.6: Classification table for stand alone *de novo* motif finders.

Consensus Based Approaches

In this approach the algorithm starts from the representation of a motif as a string. These methods begin from basic counting, where the frequency of the motif in a given sequence set is compared to the expected number of occurrences. The advantage of these approaches is that it can guarantee to find the best pattern (motif). However they are not expressive, i.e. they cannot differentiate between conserved and unconserved positions, also they cannot represent positions where multiple bases can occur.

Graph Based Methods Among the class of string based methods, graph-based approaches are the most popular among computational biologists. Pevzner and Sze [103] proposed two methods using this approach called Winnower and SP-STAR. Winnower represents motif instances as vertices in a graph and the edges represent similarity between the instances (see Figure 1.7). It then try to delete spurious edges and recover the motif with the remaining vertices. A variant of this approach is CWinnower [78]. It imposes a consensus constraint enabling it to detect weaker signals compare to Winnower. SP-STAR is a greedy algorithm which iteratively improves the sum of pair score of the motif generated by Winnower.

Another recent approaches that use graph-theoretic framework are MotifCut [46] and Trawler [41]. The main idea of MotifCut is to search for a best motif based on its maximal density subgraph, which is a set of k -mers that exhibit a large number of pairwise similarities. Trawler's approach is to cluster subgraphs based on their density. Graph based method has also been extended to find RNA motif [110] and network motifs e.g. the works of Grochow [50] and Przytycka

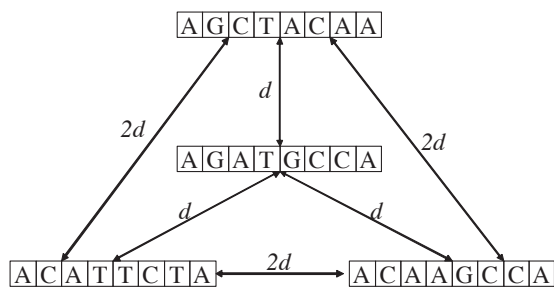


Figure 1.7: Similarity between instances is modeled using graph. The vertex **AGATGCCA** is a motif with **AGCTACAA**, **ACATTCTA**, **ACAAGCCA** as its instances. Note that the distance (edges) between the motif and instances is at most d mismatch (where $d = 2$).

[107].

Enumeration Based Methods The most basic approach in this string based approach is to search for overrepresented strings in a set of co-regulated genes. Using such approaches, over-representation is typically measured by exhaustive enumeration of all oligonucleotides of a specific length without allowing any mismatches. The observed number of occurrences of a given motif is compared to the expected number of occurrences. The expected number of occurrences and the statistical estimate is done in many ways. Here we give an overview of the different methods.

It was van Helden et.al [138] who provided an initial version of the enumeration methods. They presented a simple and fast method for the identification of DNA binding sites in the upstream regions from families of co-regulated genes in yeast (*S. Cerevisiae*). First, for each oligonucleotide of a given length, the expected frequency is computed from all non-coding, upstream regions in the genome of interest. Based on this frequency table, the expected number of occurrences of a given oligonucleotide in a specific set of sequences can be estimated. Then, a significance coefficient is computed taking into account the distinct number of all possible oligonucleotides. Finally the retrieved oligos are grouped together to extend the motifs. Later work by Apostolico [7] improved this approach to enable the finding protein motifs by allowing extensible wildcard in their motif model. The most crucial parameters here is the choice of probabilistic model for the significant occurrences. Their method is limited to searching short motifs of five to seven base pairs long. The following are some other approaches that follow this direction.

Consensus [57] is an algorithm that uses greedy enumeration method to first find pairs of sequences that share motif with greatest information content, then finding the third sequence that can be added the motif resulting in greatest in-

formation content and so on.

Tompa [134] proposed an exact method to find short motifs in DNA sequences. In principle it computed the statistical significance of motifs exhaustively. First for each k -mer s with certain number of mismatches, the number of sequences containing s is calculated. Next the probability of p_s of finding of at least one occurrences of s in a sequence drawn from a random distribution is estimated. Then the associated z -score is computed as follows:

$$z_s = \frac{N_s - Np_s}{\sqrt{Np_s(1 - p_s)}}$$

z_s gives a measure of how unlikely it is to have N_s occurrences of s given the expected number of occurrences Np_s . They proposed an algorithm to estimate the expected frequency of p_s of a word from a set of background sequences based on a Markov chain. This method was later enhanced by YMF [127] and Quickscore [111].

Enumeration method is also applied for finding *spaced dyads*. Spaced dyads are motifs consisting of *two* short conserved boxes separated by a region of *fixed* size and variable content. The earliest work on this extension is by van Helden [139]. In his approach the length of the conserved box is fixed to 3 nucleotides but the length of the spacer is different for each motifs. Different spacer lengths are systematically examined. MITRA [40] improves this approaches by allowing box length to be greater than 3bp (monad segments). MITRA relies on a specially designed data structure (mismatch tree data structure) to quickly identify possible monad segments. Another approach that aims to speed-up finding dyads is TEIRESIAS [113], by using convolution strategy to stitch the monads. The greatest shortcoming of these methods is that they only handle spaces with

only two segments.

Another approach to overcome the computational cost of enumeration methods (for both monad and dyads) is using suffix tree as a data structure. Weeder [101] is the primary example of monad motif finders that uses suffix tree. SMILE [87] is the example of motif finder that uses suffix tree to find dyad motifs.

PWM Based Approaches

Instead of the string based approaches, the problem of motif finding can also be tackled by learning a matrix model that describes the binding sites [94]. There exist three main implementations for this approach, namely *Expectation-Maximization*, *Gibbs Sampling* and *Hidden Markov Model*.

Expectation Maximization Based Methods Within the maximum likelihood estimation framework, Expectation Maximization (EM) is the primary choice of optimization algorithm. EM is a two-step iterative procedure for obtaining the maximum likelihood parameter estimates for a model of observed data and missing values [90].

EM for motif finding was first introduced by Lawrence and Reilly [73]. Although it was primarily intended for searching motifs in related proteins, the described method could also be applied in DNA sequences. Their proposed model conforms to the assumptions outlined above. Each sequence contains exactly one instance of the motif. The starting position of each motif instance is unknown and is considered as being a missing value from the data. If the motif positions are known then the observed frequencies of the nucleotides at each position in the motif are the maximum likelihood estimates of model parameters. To find the starting positions each subsequence is scored with the current estimate of

the motif model. These updated probabilities are used to re-estimate the motif model. this procedures is repeated until convergence. EM often suffers badly from local minima for short DNA motifs.

Since assuming there is exactly one copy of the motif per sequence does not hold for binding sites in DNA sequences, Bailey and Elkan proposed an advance EM implementation for motif finding called MEME [8,9]. To overcome the problem of initialization and getting stuck in local minima, MEME proposes to initialize the algorithm with a motif model based on a contiguous subsequence that gives the highest likelihood score. Therefore, each substring in the sequence set is used as a starting point for one-step iteration of EM, then the computed motif models are ordered in decreasing order of likelihood. The best motif is retained and used for further optimization steps. After the convergence the corresponding motif positions are masked and the procedure is restarted with the next motif model in the list.

Apart from MEME, many algorithms have been proposed to tackle the initialization problem in EM. They include Random Projection [21], Improbizer [5], Ortho-MEME [106] and Dragon Motif Builder [60].

Gibbs Sampling Based Methods The applicability of Gibbs sampling to solve missing value problem [131] has lead to the implementation of a Gibbs sampler for motif finding. The derivation of the exact algorithm was presented by Lawrence et.al [72]. Subsequently we observed that there are several works that proposed methods to fine-tune the Gibbs sampling algorithm for motif finding. Here we will give short description of these methods.

A version of Gibbs sampling algorithm that was especially tuned towards finding motif in DNA sequence is AlignACE [61,115]. The modification on Gibbs

sampling is done in two ways. First, one motif at the time was retrieved and the positions were masked instead of simultaneous multiple motif searching. Second, they were implemented with a fixed single nucleotide background model based on base frequency in the sequence set. Finally, the *maximum a posteriori* likelihood score was used to judge the quality of different motifs.

ANN-Spec [147] has its origin in the Gibbs sampling framework but approaches the representation of the motif model rather differently. It models the DNA binding specificity of a transcription factor using weight matrix. And uses Gibbs sampling to fit the parameter with gradient descent method. MotifSampler [133] uses Gibbs sampling to find the position probability matrix that represents the motif. The probabilistic framework is further exploited to estimate the expected number of motif instances in the sequence. BioProspector [79] modifies the motif model used in classical Gibbs samplers motif finder to allow for the modeling of gapped motifs and motifs with palindromic patterns. Frith, et.al [47] implemented GLAM that uses Gibbs sampling to automatically optimizes the alignment width and evaluates the statistical significance of its output. Gibb-sILR [93] uses Gibbs Sampling to produce a motif that exhibits locally optimized ILR (incomplete data likelihood ratio) score. Finally there is SeSiMCMC [42] which is a modification of Gibbs sampling algorithm to find structured motifs of symmetric types, as well as motifs without any explicit symmetry, in a set of unaligned DNA sequences.

The main goal of these algorithms is to get a generative probabilistic representation of the overrepresented motifs. The major advantages of this framework are: it is able to represent the motif in a very powerful way and the scoring function is motivated by the underlying probabilistic model. Additional information in the motif search like: background statistic, expression data, aligned genomes,

functional categories and position information can easily be incorporated. A major drawback is that finding the best matrix or profile is difficult (not guaranteed).

Hidden Markov Model Based Methods One of the current implementation that uses Hidden Markov Models (HMMs) for extracting motif in DNA is by Yada [6, 148] called YEBIS, even though the conceptual application of HMMs in this area has begun much earlier [62]. HMMs is used as a model for a family of sequences. There are three aspects which need to be addressed here. First is the *Topology of HMM*, it specifies the layout of the model which we use to represent a sequence family.

The model consist of three kind of states (see Figure 1.8). *Match states* model conserved parts of sequences (motifs). It specify probability distribution of characters on each conserved position. There can be any number of match states, which is normally given by the user. *Insert states* model possible gaps in between match states. Gaps can be arbitrarily long. Probability assigned to a self-loop in an insertion state models probability distribution of possible gap length. Finally, *delete states* allow to bypass some of the match states. For detailed description of HMMs and related algorithms we refer to [37].

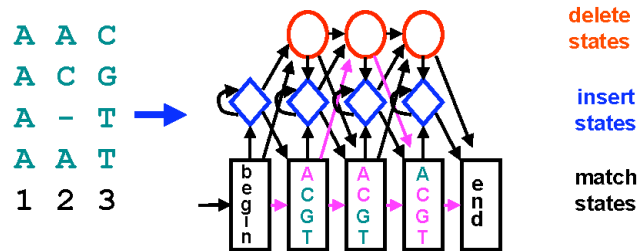


Figure 1.8: Three different states of HMM to model a set of instances. Match states model the conserved position in these instances. Insert states aims to capture the possible insertion these instances. Finally, deletion states models the deletion in the 3rd instance.

Hybrid Approaches

There also exists approaches that use the combination of the above two approaches. One of the most important tool that follows this path is MDScan [80]. Using consensus based approach, MDScan first search for motif candidates appearing in the subset of input sequences that are more likely to contain the motif (highly ChIP-enriched sequences). Subsequently, motif candidates in each similarity group are used find initialization PWM matrices. Then matrices is evaluated using maximum *a posteriori* scoring function. The highest ranking matrices (seed) is then used to scan the remaining input sequences to update the motif candidates.

HMD [145] algorithm consists of a sequence filtering component that uses a probabilistic strategy, and a graph-theoretic motif finding component that utilizes a deterministic algorithm. Sequence filtering uses the idea of *locality sensitive hashing* from computational geometry. This is based on the idea that simple hashing functions can be used to map objects in multidimensional space to buckets that have high probability of containing objects close to each other than those which are far apart. The aim of this step is to filter out corrupted sequences. For motif finding it applies CMF algorithm that is based on the concept of constraint rules [35].

Ensemble Motif Finders

In machine learning terminology, ensemble learning is a method that combines individual classifiers in some way to classify new instances. It has been theoretically shown that ensemble methods often perform better than any single classifier [34]. The difficulty in general is how to determine the suitable classifier. Inclusion of

bad performing classifier will degrade the performance. The central challenge of ensemble method therefore is how to combine the individual classifiers when their predictive quality is unknown.

In bioinformatics, ensemble methods have been applied in several prediction methods such as gene prediction [2], protein tertiary structure prediction [44, 48, 82], protein domain prediction [118] and protein secondary structure prediction [3, 95]. The success of ensemble method in these areas has been attributed to several factors. Albrecht *et.al* [3] referred their success to the noise-filtering properties of the ensemble approach, which damp the training errors of single methods. Lundström *et.al* [82] discussed that the key reason for the success of an ensemble approach is to properly measure the similarity between the different models. Furthermore, works by Harbison [54], Kihara [58], and MacIsaac [83] hinted that possible improvement can be made in motif discovery by combining output of several programs.

Ensemble methods in motif finding refers to the method of combining *de novo* motif finders for discovering regulatory motifs. In the literature, there are three existing approaches for performing ensembles for motif finding:

1. Re-rank collection of motifs returned by individual motif finders using some form of scoring function and finally report one motif.
2. Cluster collection of motifs returned by individual motif finders, find representative motif from the cluster and re-score them.
3. Cluster motifs from the same rank and select sites from the cluster.

Below we describe, in detail, of the methods for each approach.

Re-ranking Approach is taken by SCOPE [27] and cBEST [36, 63]. The distinctive difference between them is on the scoring function they use.

SCOPE uses BEAM [24], PRISM [25], and SPACER [28] for its component motif finders. These three component motif finders use semi-greedy algorithm in their approach. In particular BEAM is aimed at the identification of non-degenerate motifs, PRISM for identification of degenerate motifs with contiguous critical residues and SPACER for highly degenerate motifs.

In SCOPE, first motif reported by these component motif finders are filtered out based on its redundancy, subsequently the filtered motifs are scored and ranked based on SCOPE's scoring function.

In principle SCOPE uses p -value as the basis of its scoring function. It measures the motif significance based on probability of a motif m will have the sufficient occurrences within a particular null hypothesis. Let M be a random variable over the full space of IUPAC word. The p -value of a particular motif m denoted by $p(M = m)$ determines the significance of the occurrence of motif m over some random motif M in background sequence of the given species. Hence, the final scoring function of SCOPE is to find a motif that maximize:

$$Sig = -\log(p(M = m).N)$$

where a normalization factor N is the total number of length $|m|$ oligos in the input sequence.

The main intuition behind this scoring function is that the higher the Sig score, the probability of accepting the hypothesis that motif m is more significant than any random motif M in background sequence is also higher.

cBEST uses AlignACE, BioProspector, CONSENSUS, and MEME as its com-

ponent motif finders. In principle cBEST employ Bayesian model to improve the motif score from any generic motif finders. The main hypothesis is that if motif M is good it will have several similar-looking motifs – from all the motif finders’ output – present within input sequence S . Hence, the idea is to maximize the probability of motif M having such unknown number of similar-looking motifs. Given an input sequence S , motif M , an unknown motif matrix Φ , motif’s matrix θ (i.e. motif’s nucleotide composition), and known parameters θ_0 (vector of nucleotide composition of background) and a pre-specified parameter p_0 (*a priori* probability that a particular string being a motif site), the Bayesian model that describe the probability of motif M occurs together with some unknown motif is described as: $p(\Phi, M|S, \theta_0, p_0)$. The final scoring function of BEST is to maximize posterior distribution of the probability.

Motif Cluster Approach This approach is taken by Webmotifs [49, 114], it uses AlignACE, MDScan, MEME, and Weeder as its component motif finders. Initially set of motifs returned by these component motif finders will be clustered with k -medoids clustering method using the inter-motif distance metric:

$$D = \frac{1}{w} \sum_{i=1}^w \frac{1}{\sqrt{2}} \sum_{L \in \{A, C, G, T\}} (a_{i,L} - b_{i,L})^2$$

where w is the motif width, and $a_{i,L}$ and $b_{i,L}$ are the estimated probabilities of observing base L at position i of motifs a and b respectively. The centroid motif for each cluster is then scored using enrichment score formulated as:

$$p = \sum_{i=b}^{\min(B,g)} \frac{\binom{B}{i} \binom{G-B}{g-i}}{\binom{G}{g}}$$

where B is the number of input sequences and G is the total number of sequences represented in microarray or genome. The quantities b and g represent

the subset of B and G that match the motif.

The advantage of Re-ranking and Motif Cluster approaches is that they can select the best motif out of all the motif finders. However, these methods only select correct binding sites of one motif predicted by one individual motif finder. It will fail to discover correct binding sites found by more than one motif finders.

Sites Voting Approach Finally, EMD [59] follows this last approach. It uses AlignACE, BioProspector, MDScan, MEME and MotifSampler as its component motif finders. Initially, each motif finder M_i report top K scoring motifs. Subsequently the motifs will be clustered into K -groups based on its ranking. For each of the K -groups, it computes the number of times each position of a site occurs (this count is denoted as V_p). These sites is further smoothen by only selecting those falls within 8-15bp length. The final stage is to select sites that has the highest V_p count in each of the cluster.

The benefit of this approach is that it can find more binding sites from multiple motif finders. However, it will miss the true binding sites that come from motifs of different ranking since true binding sites most likely come from different motifs of different rank.

From these three approaches we observe that two key issues in ensemble method are not addressed. Firstly, among the motifs reported by multiple motif finders, there are many false motifs. How do we filter those false motifs? Secondly, even for a motif which can approximate the true motif, some of the instances (sites) of the motif are real while the rest are noise. How can we remove those false sites?

In our thesis we propose a novel methods that aim to overcome the limitation of existing ensemble methods. Specifically we believe that an effective integration of results is necessary at both motif level and sites level.

1.2.3 Methods Using Genomical Data

There are methods that exploits domain knowledge for motif finding. This domain knowledge can provide a powerful information to improve the performance of *de novo* methods. Some works that follow this path include: PhyME [128], it exploits the comparative sequence analysis by combining interspecies overrepresentation and interspecies conservation for motif finding.

It consists of two steps: *alignment* and *motif finding* step. In alignment step PhyME extract blocks of high sequence similarity between reference species and each of the other species. Its main assumption is that the motif that occurs in such locally conserved region are deemed orthologous. At the end of this step we obtain a regulatory regions of potentially co-regulated genes along their orthologs from other species. This region (sequences) is then used for the motif finding step.

In motif finding step, PhyME uses an Expectation Maximization (EM) algorithm to search for motif that best explain the data. When evaluating the motif, its orthologous occurrences are assumed related to each other by a probabilistic model of evolution that takes into account the varying phylogenetic distances among the species. The other algorithm that uses this information are PhyloGibbs [123] and EMnEM [91].

REDUCE [116] uses microarray (expression) data to find cis-regulatory elements. This method takes into account the combinatorial nature of gene expression regulation. REDUCE works by fitting a multivariate predictive model to a single genome-wide expression pattern. The expression level of a gene is modeled as a sum of independent contributions from all transcription factors for which binding sites occur in promoter region. Finally a forward parameter selection strategy is used to select motifs from a large set of candidate motifs. Other

algorithms that uses gene expressions but differs in their method of using correlations statistics include MARS [31,129], RankMotif++ [29], MEDUSA [89], and RegTREE [105].

Other external genomic data has been used for motif finding include nucleosome occupancy [55], protein-DNA interactions [67,92] and familial binding profiles [85].

1.2.4 Motif Evaluation and Benchmarks

Due to the large number of available tools, robust assessment of motif discovery methods becomes important, not only for validation but also for pointing out the most promising directions for future research in the field.

Tompa [135] published an important and timely contribution to the field by providing a benchmark dataset. Up to then there have been only a few small-scale assessment of some of these motif discovery tools [103,126]. Tompa's assessment is the first large-scale effort to measure the performance of 13 motif discovery tools. These tools do not use auxiliary information such as comparative sequence analysis, mRNA expression levels or chromatin immunoprecipitation results.

Tompa's benchmark dataset has been constructed based on real transcription factor binding sites drawn from four different organisms yeast, fruitfly, human and mouse. It consists of 56 datasets in total. Each dataset consists of 1-35 sequences and each sequence is of length up to 3000 bp. The datasets are constructed from three different types of background sequences. They are (i) real promoter sequences, (ii) randomly chosen promoter sequences from the same genome (called generic), and (iii) sequences generated by Markov chain of order 3 (called markov).

The performance of motif discovery tools is measured according the following statistics: *sensitivity* (SN), *positive predictive value* (PPV), *specificity*,

performance coefficient (PC), *average site performance (ASP)* and *correlation coefficient (CC)*. They are formulated as follows:

$$SN = TP/(TP + FN)$$

(*Sensitivity*)

$$PPV = TP/(TP + FP)$$

(*Precision*)

$$Specificity = TN/(TN + FP)$$

$$PC = TP/(TP + FN + FP)$$

$$ASP = (SN + PPV)/2$$

$$CC = \frac{TP.TN - FN.FP}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}}$$

The following abbreviations are used to specify how the scores are calculated: *TP* (true positive) is the number of the overlapped nucleotides both in real and predicted sites, *FP* (false positives) the number of the overlapped nucleotides not in known sites but in predicted sites, *TN* (true negatives) the number of the overlapped nucleotides neither in known sites or in predicted sites, and *FN* (false negatives) the number of the overlapped nucleotides neither in known sites and but not in predicted sites.

This thesis uses Tompa’s datasets for benchmarking. We also adopt the above statistical measures for evaluation.

1.3 Motivations

Our thesis is primarily motivated by two factors. First is the challenges faced while dealing with real biological data, and second is the challenge from the current practice in biological domain.

1.3.1 Challenges from Real Biological Data

An inspection of transcription factor database such as TRANSFAC [146], or relevant literature like [39, 66], reveals that there is significant variation among binding sites of any single transcription factors. Here are some important issues in dealing with real biological data.

1. Many motifs are known to be composite patterns which are groups of monad patterns (short contiguous patterns with some mismatches) that occur relatively near each other [54]. For example, the binding site for *ArcA-P*, a transcription factor for regulating gene related to the respiratory metabolism in *E.coli* [81], can be regarded as two conserved segments, separated by a spacer of length approximately 6 [88]. Another example is *Mcm1* [68] or often called as the early cell cycle box (ECB) [132] which has 3 segments and two spacers. Note that a spacer does not necessarily mean that the characters in the spacer are completely random and arbitrary, but these characters are not very conserved in different instances.

In fact, in some regulatory mechanisms, a single transcription factor may bind to two or more sites that are relatively close to each other – as is fre-

quently the case, for instance, of RNA polymerase [109]. Identifying these sites is similar to finding a spaced motif. Spaced motifs may also be associated with co-regulated genes that share two or more transcription factors and the binding sites are often recognized by different macromolecular complexes that make contact with one another [98,143]. Our focus in this thesis is to find such complex motifs that could contain spacers.

2. Real samples may contain biased nucleotide. This type of samples has a functional significance. For example Bernardi [12] have demonstrated that the genomes of warm-bodied animals (mammals, birds, etc.) are organized heterogeneously, with **G + C**- and gene-rich "isochores" interspersed with regions of lower **G + C** content. The motif finding problems becomes more difficult if the background nucleotides composition in the sample is skewed.
3. One of the fundamental issues in identifying TFBS is the determination of its biological significance. It is difficult to quantify them. Sometimes the signals that have low statistical significance, still can have real, biological significance. For example some of the TFBS for activating protein Hap1 is dictated by its structural environment [74].

1.3.2 Challenges from Current Practice

Though a lot of tools have been developed, little knowledge is known on which motif finder should be used for a particular dataset. Individually, these motif finders perform unimpressively overall based on Tompa's benchmark datasets [135] (see Figure 1.9).

Moreover, these motif finders vary in their definitions of what constitute a motif, and in their methods for finding statistically overrepresented motifs. This

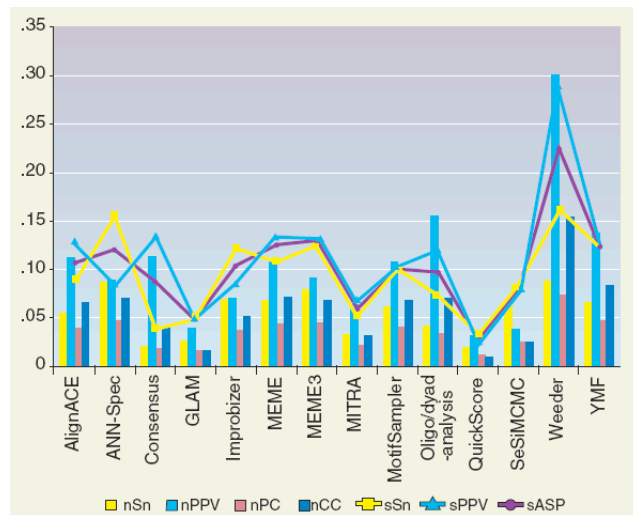


Figure 1.9: Performance of motif finders in Tompa's benchmark dataset is unimpressive overall with sensitivity ≤ 0.086 and precision ≤ 0.300 [135].

makes different motif finders perform well for identifying binding sites of certain types of datasets only. There is no clear ways for biologists to choose the motif finder that is most suitable for their task. Hence, we can see that there is an immediate need for a more effective and efficient methods that allows the biologist to make use these diverse motif finders for finding novel regulatory sites accurately.

1.4 Contributions of the Thesis

This section describes the significant contributions of the thesis:

1. We address the problem of motif finding for generic spaced motifs. Spaced motifs, an important class of transcription factors, consists of several short segments separated by spacers of different lengths. Existing motif finding algorithms are either designed for monad motifs or have assumptions on the spacer lengths or can handle at most two segments [18,134,138]. To address this issue, we propose a new method called SPACE. We introduce the notion

of *submotifs* to capture the ungapped segments in the spaced motifs and formulate the motif finding problem as frequent submotif mining problem.

2. We also propose a novel scoring technique to measure the statistical significance of generic spaced motifs. With this measure we overcome the difficulty in handling biased samples by incorporating background sequence from various *species*. Based on experiments on real biological datasets and Tompa's benchmark datasets, we show that our algorithm outperforms the existing tools for spaced motifs in both sensitivity by 20.3% and specificity by 76%. And for monads, it performs as well as other tools.
3. We address two main difficulties in performing ensemble methods in motif finding. First, multiple motif finders may report similar spurious motifs. The challenge lies in how to remove these motifs. Second, even if the reported motif can approximate the real motif, they still contain false positive that have high similarity with the real binding sites.

To address these difficulties we propose MotifVoter. It applies a variance based statistical measure to remove the spurious motifs and then refines the prediction by filtering the noisy binding sites by using a novel voting scheme.

Validation of our method on Tompa's benchmark, real *metazoan* and *E. Coli* datasets (186 datasets in total) show that it can improve the sensitivity by 120% and precision by 77%. MotifVoter can locate almost all the binding sites found by the individual motif finders used and is able to distinguish the real binding sites from noise effectively.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows.

Chapter 2 We present our method for detection of generic spaced motifs using submotif pattern mining. Then, we describe the efficient methods for generation of motif candidates, refining motif candidate into spaced motifs, and finally scoring method to measure the statistical significance of generic spaced motifs. We perform experiments that shows the results for both spaced and monad motifs on Tompa’s benchmark dataset, real biological data and synthetic datasets.

Chapter 3 In this chapter we present our ensemble method, called MotifVoter, for integrating generic motif finders. We also show our study of the performance of individual motif finders when we include lower rank of motifs, as well as our discovery that every motif finders finds different binding sites. For this method we perform extensive experiments on 186 datasets (Tompa’s benchmark, real *metazoan* and *E.coli* datasets) examining the performance of MotifVoter: 1) in comparison with individual motif finders and other ensemble methods, 2) on different background sequence and species, 3) time complexity, and 4) its robustness.

Chapter 4 In this final chapter we will provide discussion and give our preliminary conclusion. Then we propose our future work that target at solving limitations of our approaches as well as looking at other important issues for handling real biological data.

Detection of Generic Spaced Motifs Using Submotif Pattern Mining

This chapter describes a novel approach for identifying spaced motifs with any number of spacers of different lengths. We introduce the notion of *submotifs* to capture the segments in the spaced motif and formulate the motif finding problem as a frequent submotif mining problem. We provide an algorithm called SPACE to solve the problem. Based on experiments on real biological datasets, synthetic datasets and the motif assessment benchmarks by Tompa et al., we show that our algorithm performs better than existing tools for spaced motifs with improvements in both sensitivity and specificity and for monads, SPACE performs as well as other tools.

As pointed out by Eisen in a recent survey [39], regulatory motifs could be highly complex in the biological context. Many motifs are known to be composite patterns which are groups of monad patterns (short contiguous patterns with some mismatches) that occur relatively near each other [54]. For example, the binding site for *ArcA-P*, a transcription factor for regulating gene related

to the respiratory metabolism in *E.coli* [81], can be regarded as two conserved segments, separated by a spacer of length approximately 6 [88]. Another example is *Mcm1* [68] or often called the early cell cycle box (ECB) [132] which has 3 segments and two spacers. Note that a spacer does not necessarily mean that the characters in the spacer are completely random and arbitrary, but these characters are not very conserved in different instances.

In fact, in some regulatory mechanisms, a single transcription factor may bind to two or more sites that are relatively close to each other – as is frequently the case, for instance, of RNA polymerase [109]. Identifying these sites is similar to finding a spaced motif. Spaced motifs may also be associated with co-regulated genes that share two or more transcription factors and the binding sites are often recognized by different macromolecular complexes that make contact with one another [98,143]. Our focus in this chapter is to find such complex motifs that could contain spacers.

Most of the existing algorithms are mainly designed for monad motifs. Applying these algorithms to locate spaced motifs may not be effective. By treating a spaced motif as a single monad pattern, the motif instances may not be very similar, i.e., the signal may not be strong to be detected, due to the many random (non-conserved) characters in the spacers. Or if we try to locate the individual segments of a spaced motif using these algorithms, some of the segments may be too short and may not be easily detected.

On the other hand, there are algorithms designed for spaced motifs. The methods used by existing algorithms can be classified into the following approaches. The first and the most common approach is to assume that all the spacers in the same motif are all of the same fixed length (e.g. SesiMCMC [43], OligoDyad [139]). However, in real cases, this is not the case. Another approach

to handle spacers is to enumerate all possible spacer lengths between two composite segments (e.g. YMF developed in [125] and BioProspector [79]). Although this approach can find motifs with spacers of varying length, it is inherently inefficient and is difficult to extend to more than two segments. And it may not be practical for long motifs. The third approach to locate spaced motifs is to find the monad segments first (e.g. MITRA in [40]), then based on the locations of monad segments, locate a set of possible dyads (spaced motifs with two segments). The algorithm of MITRA relies on a specially designed data structure (mismatch tree data structure) to quickly identify possible monad segments. There are other methods (e.g. [26, 87]) that make use of data structures such as suffix tree to store the regularly spaced motif before finally identifying the motif pairs to speed up the process. Almost all existing approaches only handle spaced motifs with two segments.

In this chapter we propose a new approach for finding spaced motifs, and develop a novel motif-finding algorithm that offers flexibility in handling spacers with different lengths, the number of segments, and variations in segment lengths. We formulate the motif finding problem as a frequent itemset mining problem and present an algorithm called SPACE for finding these motifs. Experimental results show that the approach is promising.

```

M=CAGTTTCAnACGTCnnGACGT
I1=TAGTTTAtATGTCcgGACAT
I2=CACTTTAtATGTCcgCACGT

```

Figure 2.1: Consider $L = 20$, $\ell = 5$, and $d = 1$. M is an example of length-20 spaced motif with three segments separated by two spacers. Then, I_1 is an instance of M since all length-5 submotifs in the three segments of M have less than 1 mismatches when comparing with I_1 . On the other hand, I_2 is not an instance of M since $M[2..6]$ and $I_2[2..6]$ have two mismatches.

Our approach is similar to TEIRESIAS [113] in building longer motif from shorter blocks. However, TEIRESIAS is computationally expensive. It uses a *convolution* strategy to stitch the shorter blocks exhaustively to find maximal patterns. It also does not handle mismatches. On the other hand our novel approach provide further advantages. We allow flexibilities in terms of allowing mismatches and provide an efficient method to find the pattern.

In this section, we provide the formal definition of a generic spaced motif and discuss the notion of *submotif* which is the core concept of our approach. We generalize the string representation of motifs as follows.

Definition 2.1 *For some pre-defined coverage ratio $r \leq 1$, a spaced motif (or simply a motif) is a length- L string formed by characters of $\{A, C, G, T, n\}$ with at least $\lfloor r \times L \rfloor$ characters in $\{A, C, G, T\}$. Each maximal substring of consecutive “ n ” represents a spacer and each maximal substring of other characters represents a segment.*

Figure 2.1 shows an example of a spaced motif M which is of length 20 and has three segments separated by two spacers. Note that the segments, as well as the spacers, can be of different lengths. The number of segments is also not fixed. Let $Z[i..j]$ be the substring of Z starting at position i and ending at position j . Any length- ℓ substring $M[i..i + \ell - 1]$ within any segment of M is called its *submotif*. Below, using submotifs, we define an instance of a spaced motif (see Figure 2.1 for an example).

Definition 2.2 *Consider a length- L spaced motif M and any length- L string I formed by characters of $\{A, C, G, T\}$. I is called an instance of M if, for every submotif $M[i..i + \ell - 1]$ and $I[i..i + \ell - 1]$ have at most d mutations, for some pre-defined constant d .*

Now, we define the *spaced motif finding problem*. Let $\mathbf{S} = \{s_1, s_2, \dots, s_t\}$ be a given set of DNA sequences. Our task is to identify spaced motifs with at least q instances in \mathbf{S} , for some predefined constant q (we call this the *minimum support*). Figure 2.2 shows an example.

```

TTGATACCGAAGATACCGATTAGAAATCACTCA
ACTACAGAAAAGCAGTAGTAAAAACGTACAGTC
GAAGACCGTCATGAGAAATCGCATAACAGGCA
TTCACCCGATAAAAATAAGGCTGTCTGGACTAA
TCGGAACAATTACGAAGAAAAGCAGTAGAAAAA

```

Figure 2.2: Consider $L = 20$, $r = 0.5$, $\ell = 5$, $d = 1$, and $q = 4$. GAAGAnnnnnnTAGAAAnn is a spaced motif of the above 5 sequences. All its instances are underlined.

By formulating the motif finding problem in this way, we have the following advantages:

1. The lengths of the segments in the motif need not be known *even if* we prefix the length of the submotif. This follows because union of an overlapping set of submotifs can represent an arbitrary length segment. This property implies that motifs with segments of arbitrary lengths could be found. Note that this does not depend on whether the motif has spacers or not.
2. The spaced motif uses multiple segments to model the functional parts, which are more conserved, and the spacers to model the non-functional parts. However, monad motif (or dyads) only has one segment (or two segments) for modeling both conserved and non-conserved regions. Hence, spaced motif can fit the conserved regions better. In other words, it yields higher specificity. We confirm this in our experiments on several datasets including the Motif Assessment Benchmark and some real biological datasets.

-
3. It provides a natural extension for finding motifs with multiple spacers, in which neither the spacer length nor the number of spacers (and segments) is known.

However, there could be too many submotifs (many of them are spurious) and the challenge is in how effectively the submotif-compositing can be done to return “good” motifs. To tackle this situation, we formulate this task as a constrained frequent submotif mining problem and propose a new algorithm for solving it.

Let $\mathbf{S} = \{s_1, s_2, \dots, s_t\}$ be the given set of t sequences. Our solution for finding spaced motifs is called *SPACE*. It consists of three main steps. Step 1 finds motif candidates, which is defined below. Step 2 refines the motif candidates into spaced motifs. Lastly, Step 3 computes the significance of the spaced motifs based on our scoring function and reports the ranked list of motifs.

We do not assume any knowledge about the number and the locations of the spacers in the motif. To identify a possible candidate for the motif, we look at each length- L substring u in \mathbf{S} , based on the definition of a spaced motif, we define an occurrence of u as follows. Let $hd(x, y)$ be the Hamming distance of two equal-length strings x and y .

Definition 2.3 *Let u be a length- L substring in \mathbf{S} . Consider another substring w of the same length in \mathbf{S} . For some pre-defined constants d and $r \in [0, 1]$, for every i , the substring $w[i..i + \ell - 1]$ is called a submotif occurrence of $u[i..i + \ell - 1]$ if $hd(u[i..i + \ell - 1], w[i..i + \ell - 1]) \leq d$. The number of characters spanned by all submotif occurrences is called the coverage of w on u . The substring w is called an occurrence of u if the coverage of w on u is at least $\lfloor r \times L \rfloor$.*

The length- L substring u is called a *motif candidate* if there exist at least q occurrences of u in \mathbf{S} . Fig. 3 shows an example of a motif candidate.

```

TTGATACCGAAGATACCGATTAGAAATCACTCA
ACTACAGAAAAGCAGTAGTAAAAACGTACAGTC
GAAGACCGTCATGAGAAATCGCATACACGAGCA
TTCACCCGATAAAAATAAGGCTGTCTGGACTAA
TCGGAACAATTACGAAGAAAAGCAGTAGAAAAA

```

Figure 2.3: Consider $L = 20$, $r = 0.5$, $\ell = 5$, $d = 1$, and $q = 4$. For the same set S of 5 sequences in Figure 2.2, GAAGATACCGATTAGAAATC has 5 occurrences. All its occurrences are underlined. Since the number of occurrences is at least q , it is a motif candidate.

Step 1 tries to find all motif candidates. A straight-forward implementation is given in Section 2.1. Note that a spaced motif is highly correlated with a motif candidate. For a motif candidate that is a real spaced motif, the locations of sub-motif occurrences in each occurrence of the motif candidate define the locations of the segments for the candidate. A different occurrence may define a different set of segments for the same candidate. By finding the set of common segments defined by the occurrences, we can generate a spaced motif. The refinement process is done in Step 2 based on frequent itemset mining, which is detailed in Section 2.2. Step 3 and our scoring function is discussed in Section 2.3. The naive implementation shown in Section 2.1 is a bit slow, Section 2.4 shows how to speed up the process.

2.1 Generation of Motif Candidates

To find all motif candidates and their occurrences, a straight-forward implementation is as follows. Fix a constant L for the motif length, for each sequence S_i , for each substring u of length L in S_i , check the coverage of all other substrings in \mathbf{S} of length L on u . If there are q occurrences of u , report u and all its occurrences. This naive procedure runs in $O(Ln^2)$ time where n is the length of a

sequence ¹. The actual running time is about 2 minutes for a dataset of 5K bp with 10 sequences on a 3.6Ghz Xeon Linux workstation with 4 processors and 8GB RAM.

At the end of this step we have a set of motif candidates, each associated with a set of occurrences. Recall that some of these occurrences may be noise or some of the submotif occurrences in them may be spurious. Our next step is to eliminate these noise to identify the spaced motifs.

2.2 Refining Motif Candidate into Spaced Motif

Given a motif candidate u and its occurrences w_1, w_2, \dots, w_c , this section discusses the way to refine u into a spaced motif. Our idea is to transform the problem into frequent itemset mining [64].

Before describing the transformation, recall that, by Definition 2.3, u and w_i share a set of submotif occurrences.

Definition 2.4 *Suppose that w is an occurrence of u . Then, $\{j \mid 1 \leq j \leq L - \ell + 1, hd(w[j..j + \ell - 1], u[j..j + \ell - 1]) \leq d\}$ is called the itemset of w with respect to u .*

Figure 2.4 demonstrates the itemset concept. For an itemset J , we can construct a spaced motif $M_{u,J}$ of length L such that $M_{u,J}[i] = u[i]$ if $0 \leq i - j < \ell$ for some $j \in J$; otherwise, $M_{u,J}[i] = n$. An itemset is called a frequent pattern if it has at least q occurrences. The following lemma states the relationship between frequent itemset and spaced motif. Note that there is an assumption behind this transformation. While we allow different gaps to have different lengths in

¹Because the sequence length n dominates over t (number of sequences), in this analysis we exclude it.

```

GAAGATACCGATTAGAAATC:
  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
GAAAAGCAGTAGTAAAAACG:
  {1, 13, 14}
GAAGACCGTCATGAGAAATC:
  {1, 11, 12, 13, 14, 15, 16}
CCCGATAAAAATAAGGCTGT:
  {3, 4, 12}
GAAGAAAAGCAGTAGAAAAA:
  {1, 2, 13, 14}

```

Figure 2.4: Consider $L = 20$, $r = 0.5$, $\ell = 5$, $d = 1$, and $q = 4$. With respect to the sequence set S in Figure 2.3, this figure shows the 5 occurrences of GAAGATACCGATTAGAAATC and their corresponding itemsets. Note that $\{1, 13, 14\}$ is the frequent itemset which appears 4 times. Hence, GAAGAnnnnnnTAGAAAn is a spaced motif of the set S .

the same motif, for a gap in the motif, the length of this gap is the same in all instances.

In practice, generic frequent itemset mining methods (e.g. LCM [136], MAFIA [22]) can be used for this procedure. Since the running time each of this algorithm varies, we do not evaluate the time complexity of this step.

Lemma 2.1 *Let J be a frequent pattern of u with at least q support. If $M_{u,J}$ has coverage at least $\lfloor r \times L \rfloor$, $M_{u,J}$ is a spaced motif.*

Proof 2.1 *Since J is a frequent pattern with at least q support, $M_{u,J}$ has at least q instances. Also, $M_{u,J}$ has coverage at least $\lfloor r \times L \rfloor$, so $M_{u,J}$ is a spaced motif.*

■

Hence, given a motif candidate u and its occurrences, we can refine u as a spaced motif as follows.

1. Generate the itemsets for all occurrences of u .
2. Find the frequent itemsets F which appear at least q times.

3. Report the spaced motif corresponding to F with sufficient coverage.

Algorithm 1 shows the complete scheme of the algorithm.

Algorithm 1 SPACE

Require: $L, r, q, l, d, \mathbf{S}$

Ensure: Ranked motifs

- 1: from \mathbf{S} generate the set of motif candidates (\mathbf{D}), each associated with a set of occurrences
 - 2: **for** each motif candidate u in \mathbf{D} **do**
 - 3: Let \mathbf{W} be the set of occurrences of u
 - 4: Find all frequent patterns that appear in \mathbf{W}
 - 5: **for** each frequent pattern **do**
 - 6: construct the corresponding spaced motif M
 - 7: If M has enough coverage, keep and score M
 - 8: **end for**
 - 9: **end for**
 - 10: return ranked spaced motifs
-

2.3 Significance Testing and Scoring

We adapt the motif scoring technique introduced in Weeder [100] to compute the significance of spaced motifs. Intuitively, a motif is significant if (1) the total number of its occurrences in all input sequences is a lot more than expected with respect to the background and (2) the pattern is either very conserved or occurs in quite a number of the input sequences. So, Weeder’s scoring mechanism computes two values to capture these two properties.

Let M be the motif, $E(M, e)$ be the expected frequency of M with at most e mutations based on a set of background sequences (we will show how to compute E later in this section). Then, $E(M, e) \cdot \sum \text{len}(s_i)$, where $\text{len}(s_i)$ denotes the length of i -th sequence s_i , represents the expected frequency of M with at most e mutations in all input sequences. To capture property (1), we count the total number of observed occurrences of M (with at most e mutations), $Occ_s(M, e)$, in

all input sequences and compute the *occurrence score*, $\beta(M)$ as follows.

$$\beta(M) = \log \frac{Occ_s(M, e)}{E(M, e) \cdot \sum len(s_i)} \quad (2.1)$$

To capture property (2), for a sequence s'_i with an occurrence of M , we consider the most conserved pattern of M and let e_i be the number of mutations of this best pattern. The value of $E(M, e_i) \cdot len(s'_i)$ represents the expected frequency of the occurrences of this motif in s'_i . This value is smaller if the motif is more conserved. Then, we compute the *sequence-specific score*, $\sigma(M)$ as follows. If the pattern is very conserved and/or occurs in many sequences, $\sigma(M)$ is large.

$$\sigma(M) = \sum_i \log \frac{1}{E(M, e_i) \cdot len(s'_i)} \quad (2.2)$$

Finally the score of each motif, $MotifScore(M)$, is $\sigma(M) + \beta(M)$.

The value of $E(M, e)$ is computed by summing the expected frequency $E(M')$ of M' in the background sequences for all M' with at most e mutations from M . When M' contains no spacer and is of length shorter than or equal to 8, the expected frequency value $E(M')$ is pre-computed from background sequences obtained from Regulatory Sequence Analysis Tool (RSAT) database site ² [137]. These background sequences of the organisms are taken from 1000bp upstream regions of all their annotated genes.

When M' contains spacers and is of length shorter than or equal to 8, $E(M')$ equals the sum of $E(M'')$ among all possible M'' with the spacers n 's replacing by $\{A, C, G, T\}$.

When M' is of length longer than 8 and with or without spacers, we are unable to precompute the frequency values since it is long. Instead, the expected

²<http://rsat.ulb.ac.be/rsat>

frequency of M' is modeled using seventh order Markov chain. Suppose $M' = p_1 p_2 \dots p_k$ with k greater than 8. $E(M')$ can be computed as follows:

$$E(M') = E(p_1 p_2 \dots p_8) \prod_{i=9}^k P(p_i | p_{i-7} \dots p_{i-1})$$

The conditional probability $P(p_i | p_{i-7} \dots p_{i-1})$ of having nucleotide p_i preceded by nucleotides $p_{i-7} \dots p_{i-1}$, is computed by using the expected frequency of 8-mers:

$$P(p_i | p_{i-7}, \dots, p_{i-1}) = \frac{E(p_{i-7} \dots p_i)}{E(p_{i-7} \dots p_{i-1} n)}$$

2.4 Efficient Generation of Motif Candidates

This section shows how to speed up Step 1, the motif candidate generation step. The observations that lead to the speed up are as follow. Recall that ℓ is the length of a submotif.

Lemma 2.2 *Let the coverage of $S_a[b..b + L - 1]$ on $S_i[j..j + L - 1]$ be \mathcal{C} . Then, the coverage \mathcal{C}' of $S_a[b + 1..b + L]$ on $S_i[j + 1..j + L]$ can be computed as follows.*

$$\mathcal{C}' = \begin{cases} \mathcal{C} + 1 & \text{if } \alpha = 0 \text{ and } \beta = 1 \\ \mathcal{C} & \text{if } \alpha = \beta = 0 \text{ or } \alpha = \beta = 1 \\ \mathcal{C} - 1 & \text{if } \alpha = 1 \text{ and } \beta = 0 \end{cases}$$

where $\alpha = 1$ if the prefix $S_i[j..j + \ell - 1]$ of $S_i[j..j + L - 1]$ is a submotif occurrence, that is, $hd(S_i[j..j + \ell - 1], S_a[b..b + \ell - 1]) \leq d$, otherwise $\alpha = 0$. Similarly, $\beta = 1$ if the suffix $S_i[j + L - \ell + 1..j + L]$ of $S_i[j + 1..j + L]$ is a submotif occurrence, that is, $hd(S_i[j + L - \ell + 1..j + L], S_a[b + L - \ell + 1..b + L]) \leq d$, otherwise $\beta = 0$.

Proof 2.2 Note that when considering all length- ℓ substrings of $S_i[j + 1..j + L]$ and $S_i[j..j + L - 1]$, the only substrings that they are different are $S_i[j..j + \ell - 1]$ which is in $S_i[j..j + L - 1]$, but not in $S_i[j + 1..j + L]$, and $S_i[j + L - \ell + 1..j + L]$ which is in $S_i[j + 1..j + L]$, but not in $S_i[j..j + L - 1]$.

If $\alpha = 1$, it means that $S_i[j..j + \ell - 1]$ is a submotif of $S_i[j..j + L - 1]$ with respect to $S_a[b..b + L - 1]$. This submotif will not be in $S_i[j + 1..j + L]$. If $\beta = 1$, then $S_i[j + L - \ell + 1..j + L]$ is a submotif of $S_i[j + 1..j + L]$ with respect to $S_a[b + 1..b + L]$ which is not in $S_i[j..j + L - 1]$.

So, the result follows. ■

Based on Lemma 1, once we have calculated the coverage of $S_a[b..b + L - 1]$ on $S_i[j..j + L - 1]$, to calculate the coverage of $S_a[b + 1..b + L]$ on $S_i[j + 1..j + L]$, it only takes $O(1)$ time. To calculate the coverage of all substrings on one sequence against all potential motif candidates in another sequence, the time complexity can then be reduced to $O(n^2)$.

Since we are only interested in the substrings that can have a coverage at least $\lfloor r \times L \rfloor$, we can further prune the computation according to the following lemma.

Lemma 2.3 Let the coverage of $S_a[b..b + L - 1]$ on $S_i[j..j + L - 1]$ be \mathcal{C} . Let y be the length of the longest suffix of $S_i[j..j + L - 1]$ that is not covered by a submotif occurrence. The coverage \mathcal{C}' of $S_a[b + p..b + p + L - 1]$ on $S_i[j + p..j + p + L - 1]$ is upper bounded by $\mathcal{C} + \min\{y, \ell - 1\} + p$ for any $p > 0$.

Proof 2.3 We try to upper bound the value of \mathcal{C}' as follows. Comparing $S_i[j + p..j + p + L - 1]$ with $S_i[j..j + L - 1]$, There are p new characters. Assuming that all these characters are covered by submotifs, the coverage can be increased at most by p . For the suffix of $S_i[j..j + L - 1]$ that is not covered by any submotif occurrence. If $y < \ell - 1$, then when considering $S_i[j + p..j + p + L - 1]$, these

y characters may all be covered by a submotif, so the coverage can be increased by at most *y*. On the other hand, if $y \geq \ell - 1$, then at most the last $\ell - 1$ characters, which can form a submotif with one new character, can be covered by a submotif occurrence when considering $S_i[j + p..j + p + L - 1]$, so the coverage can be increased by at most $\ell - 1$. So, the result follows. ■

By Lemma 2, after computing the coverage of $S_a[b..b+L-1]$ on $S_i[j..j+L-1]$, based on the upper bound calculation, we can skip the computation of coverage for some substrings and jump to the substrings $S_a[b + p..b + p + L - 1]$ and $S_i[j + p..j + p + L - 1]$ with the smallest p such that $\mathcal{C} + \min\{y, \ell - 1\} + p \geq \lfloor r \times L \rfloor$. From our experiments, we found that the running time for generating the motif candidates and their occurrences have been reduced from 2 minutes to less than 1 second on the same dataset of 5K bp with 10 sequences on a 3.6Ghz Xeon Linux workstation with 4 processors and 8GB RAM. So, it is feasible for large datasets.

2.5 The Final Ranking of Motifs in SPACE

We follow a similar idea as Weeder in producing the final ranked list of motifs based on the intuition that the real motifs will constantly have higher ranking compared to spurious motifs even on different parameter settings. For each dataset, we run SPACE using 12 parameter settings:

- $L = 8, r = 0.5, q = t, \ell = 5, d = 1$
- $L = 8, r = 0.5, q = 0.5t, \ell = 5, d = 1$
- $L = 8, r = 0.8, q = t, \ell = 5, d = 1$
- $L = 8, r = 0.8, q = 0.5t, \ell = 5, d = 1$

- $L = 15, r = 0.5, q = t, \ell = 5, d = 1$
- $L = 15, r = 0.5, q = 0.5t, \ell = 5, d = 1$
- $L = 15, r = 0.8, q = t, \ell = 5, d = 1$
- $L = 15, r = 0.8, q = 0.5t, \ell = 5, d = 1$
- $L = 20, r = 0.5, q = t, \ell = 5, d = 1$
- $L = 20, r = 0.5, q = 0.5t, \ell = 5, d = 1$
- $L = 20, r = 0.8, q = t, \ell = 5, d = 1$
- $L = 20, r = 0.8, q = 0.5t, \ell = 5, d = 1$

Where L is the motif length, r the coverage ratio, q the minimum support, ℓ the submotif length, d number of substitution of submotif, and t is total number of input sequences.

Motif candidate length (L) is chosen in range of 8-20 bp because in general the transcription factor binding site is short. The quorum (q) is chosen to be at least 0.5 is because we assume the dataset is well enriched with the binding sites. And submotif parameter (ℓ, d) is chosen to be (5,1) because we believe that this setting is less likely to give a spurious occurrences in a given motif candidate.

We collect the top 10 motifs based on their significant scores from each run. By one run we mean one dataset applied with one parameter listed above. Here we make use of a redundancy measure to give the final ranking of the motifs, using the observation that if a motif is real, it will have more related redundant motifs in the output list,

A motif M' is said to be a redundant motif of M if:

$$\frac{\text{align}(M, M')}{|M|} \geq 0.6$$

where $\text{align}(M, M')$ is the maximum number of matched bases in an ungapped alignment between M and M' . Two aligned n characters are considered as matched.

For each run, we select the motifs that have at least 2 other redundant motifs in the same run. The motifs that are not selected in this round will be discarded from the list. For the remaining motifs, we further discard those that do not have 2 other redundant motifs in different runs. Then, the motifs are ranked in decreasing order of the total number of its redundancy motifs in all the lists of all runs. In case of a tie, the motif with higher significant score will be reported first.

2.6 Experimental Results

We perform experiments on four classes of datasets namely, 1) 9 biological datasets that are known to contain spacers, and 2) 4 synthetic test cases consisting of different variations of spaced motifs, and 3) The datasets from 4 different species, proposed by [135] for the assessment of motif discovery algorithms and 4) 10 real biological datasets consisting monad motifs.

The real biological datasets of the spaced and monad motifs are constructed from -1000 to -1 upstream region of all the genes co-regulated by respective transcription factor, truncating the region if it overlaps with an upstream open reading frame (ORF). They are obtained from RSAT [139] and ABS [14] database respectively. The assessment results are reported below.

For performance evaluation, we use the same four measures proposed in [135] namely, *sensitivity* (nSn), *positive predictive value* ($nPPV$), *performance coefficient* (nPC), and *correlation coefficient* (nCC). Index n is used to denote that the assessment is done at the nucleotide level instead of site level³. All experiments have been performed on a 3.6Ghz Xeon Linux workstation with 4 processors and 8GB RAM.

2.6.1 Results on Datasets with Spaced Motifs

We first evaluate the performance of SPACE for spaced motifs, that is, motifs with at least one spacer.

Real Biological Datasets

In the literature, we found 9 transcription factor binding sites whose motifs have gaps. They are: GAL4P [65], ARCA-P [81], MCM1 [68] or ECB [132], and 6 transcriptional regulators of C6 Zinc cluster family [121].

Comparison is done with MITRA⁴ [40] and BioProspector⁵, both of which can handle motifs with spacers. We let MITRA search for motifs up to 12bp (the maximum possible) and we require it to find the motif on the given strands only. For BioProspector we allow the algorithm to search for motifs with block size ranging from 4 to 10 and gap size ranging from 0 to 12. In the comparison, instead of picking the motif of rank 1, we pick the first motif among the top 20 that can give a better nSn and $nPPV$ than the motif of rank 1. Table 2.1 summarizes the comparison results. From the table, we see that the selected

³Note that the consensus of the motifs reported may be the same for different algorithms, but the predicted binding sites may be different, thus yielding a difference in the performance measures

⁴<http://fluff.cs.columbia.edu:8080/domain/mitra.html>

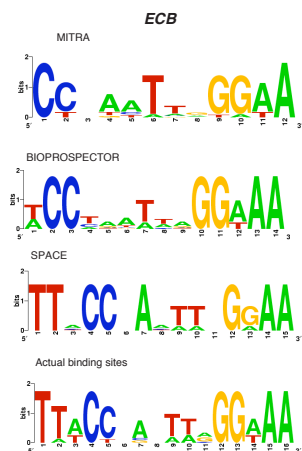
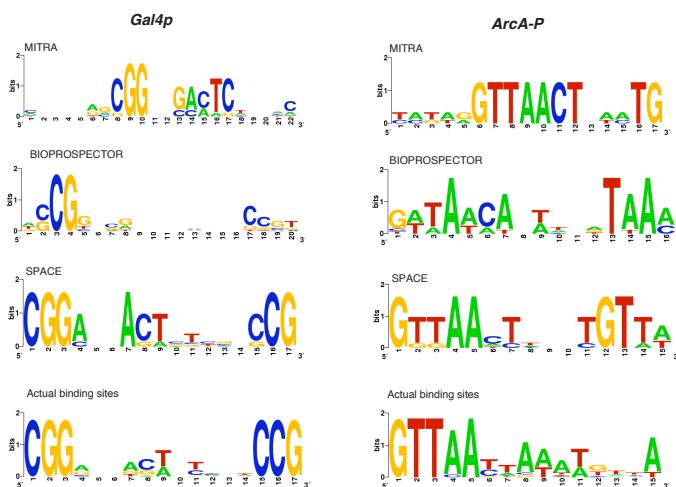
⁵<http://ai.stanford.edu/~xslui/BioProspector/>

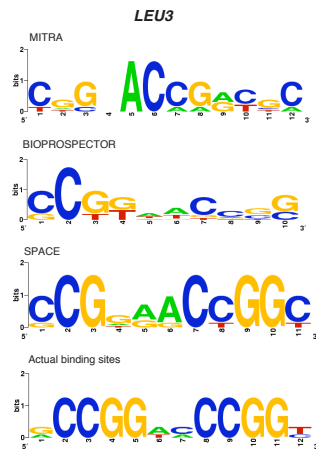
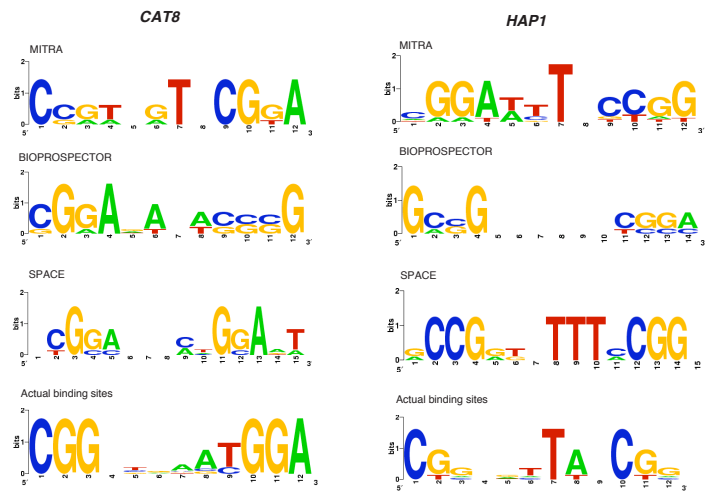
motifs of SPACE are usually of higher rank than the other two and the averaged performance is better across all measures.

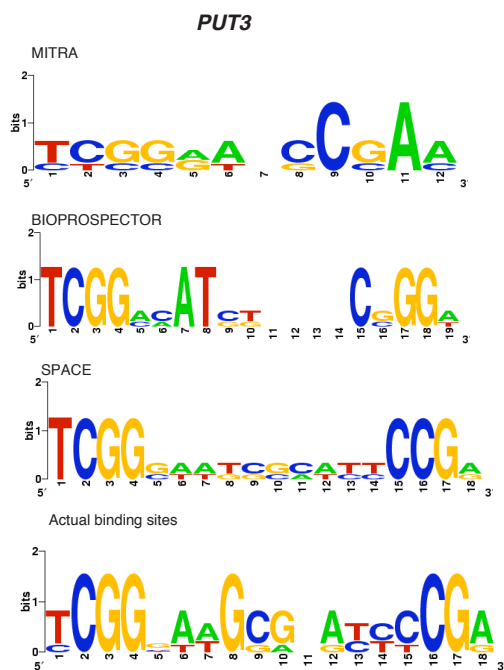
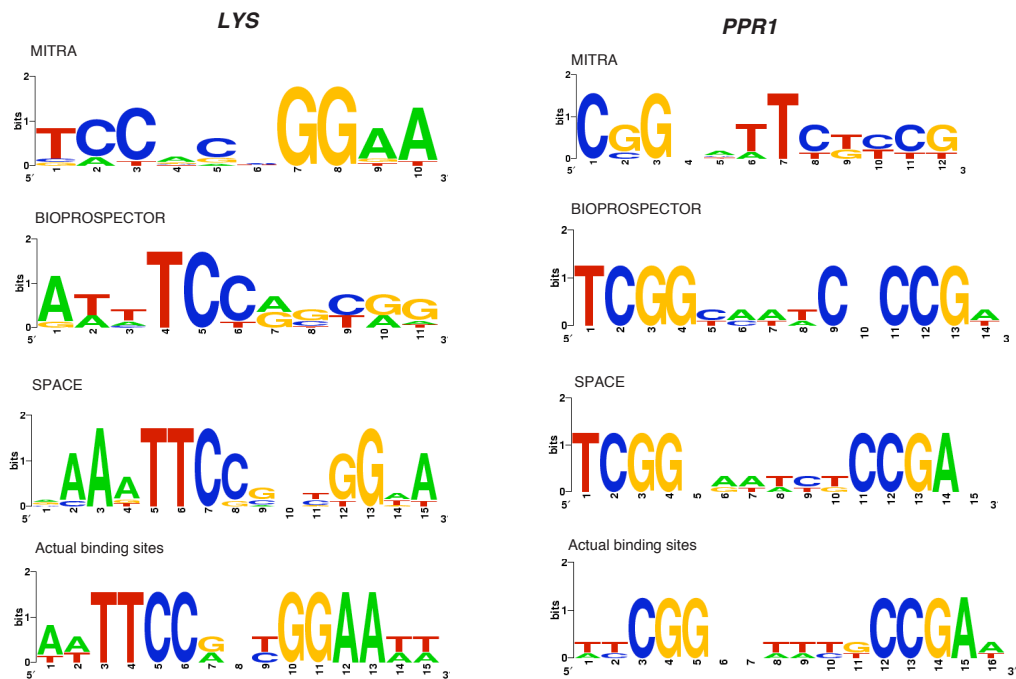
TF		Motif	RANK	nSn	nPPV	nCC	nPC
GAL4P [65]	Actual	CGGRnnRCYnYnCnCCG					
	SPACE	CGGAnGACTnnnnTCCG	1	0.80	0.55	0.48	0.65
	MITRA	AGCGGnnGACTC	1	0.68	0.40	0.34	0.51
	BioProspector	TCCGGnnnnnnnnnnCCGT	1	0.63	0.26	0.23	0.39
ARCA-P [81]	Actual	GTTAAAnnnnnnGTTAA					
	SPACE	GTTAnnnnnnATGTTA	1	0.80	0.59	0.52	0.68
	MITRA	GTTAACT	15	0.60	0.32	0.26	0.42
	BioProspector	GTTATnnnnnnnTAAA	4	0.66	0.25	0.22	0.38
ECB [68, 132]	Actual	TACCnAATTnGGTAA					
	SPACE	TTACnnAATTnGGAA	1	0.70	0.58	0.46	0.61
	MITRA	CCAAAnTTGnGAA	2	0.61	0.48	0.36	0.51
	BioProspector	TCCTAnnnnGGAAA	2	0.72	0.33	0.30	0.47
CAT8 [139]	Actual	CGGnnnnnnGGA					
	SPACE	CGGAnnnnnGGAAT	1	0.74	0.52	0.44	0.62
	MITRA	CCGTnGTTCCGGA	5	0.57	0.31	0.25	0.40
	BioProspector	CGGAnnnnCGGG	1	0.64	0.40	0.33	0.50
HAP1 [121, 130]	Actual	CGGnnnTAnCGGnnnTA					
	SPACE	CCGGnVTTTnCGGH	2	0.67	0.67	0.50	0.66
	MITRA	CGGATnTnCCGG	1	0.67	0.18	0.17	0.33
	BioProspector	GCGGnnnnnnCGGA	5	0.85	0.15	0.14	0.34
LEU3 [130]	Actual	RCCGGnnCCGGY					
	SPACE	CCGGnnCCGGCT	1	0.85	0.28	0.27	0.48
	MITRA	CGGnACCGAnGC	2	0.46	0.13	0.11	0.22
	BioProspector	CCGGnnCCGG	1	0.71	0.19	0.17	0.35
LYS [10]	Actual	WWTCCRnYGGAWWW					
	SPACE	AATTCCGnnGGAA	4	0.62	0.59	0.43	0.60
	MITRA	TCCACnGGAA	4	0.75	0.33	0.30	0.48
	BioProspector	ATTTcAnAGCGG	3	0.56	0.27	0.22	0.37
PPR [121]	Actual	WYCGGnnWYKCCGAW					
	SPACE	TCGGnnnnnGCCGAAG	1	0.88	0.75	0.68	0.81
	MITRA	CGGGnTTCnCG	9	0.67	0.36	0.30	0.47
	BioProspector	TCGGCnnTCTCCGA	1	0.78	0.52	0.45	0.63
PUT3 [121]	Actual	YCGGnAnGCGnAnnnCCGA					
	SPACE	TCGGGAnnnnnnnTCCG	1	0.89	0.76	0.69	0.81
	MITRA	TCGGnAnCCGAA	2	0.75	0.62	0.52	0.66
	BioProspector	TCGGAnnnnnnnnnCCGGA	2	0.64	0.64	0.47	0.62
AVERAGE	SPACE			0.77	0.59	0.50	0.66
	MITRA			0.64	0.35	0.29	0.44
	BioProspector			0.69	0.33	0.28	0.45

Table 2.1: Comparison of SPACE, MITRA and BioProspector on spaced motifs in real biological datasets (the first motif among the top 20 that gives a better nSn and $nPPV$ than motif of Rank 1 is used for comparison).

In the following figures, we exhibit the conservation of the binding sites found by MITRA, BIOPROSPECTOR and SPACE on the above real biological datasets:





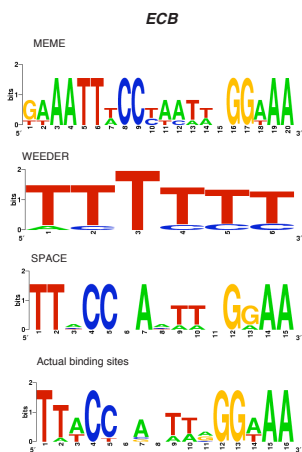
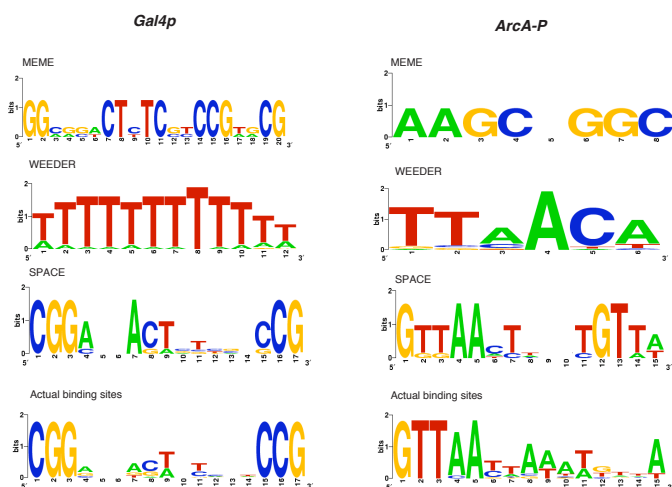


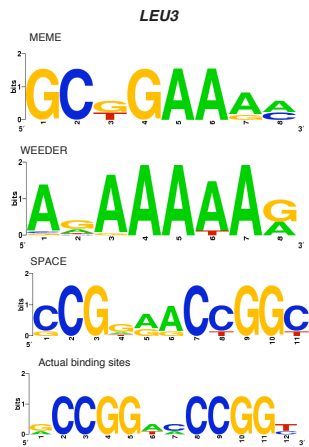
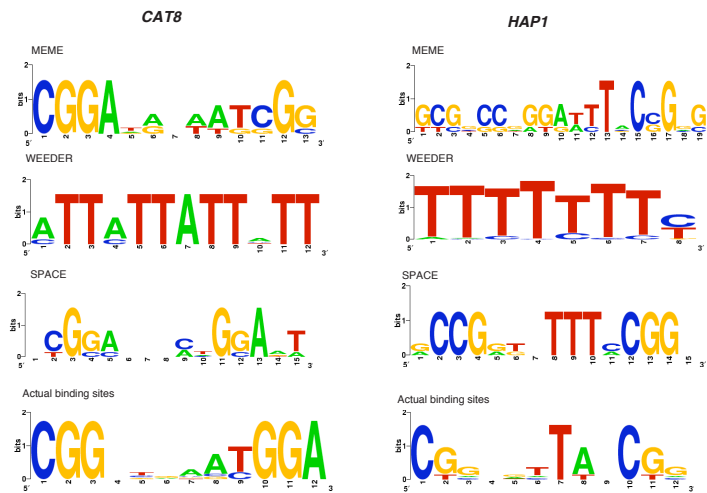
We also compare the performance of monad motif finders MEME and Weeder on the above real datasets.

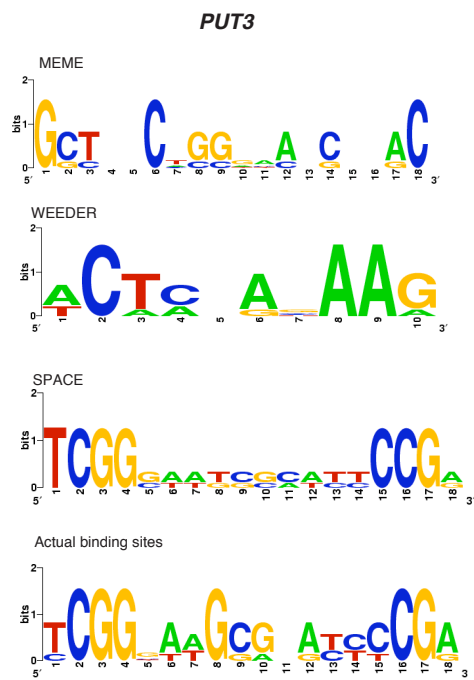
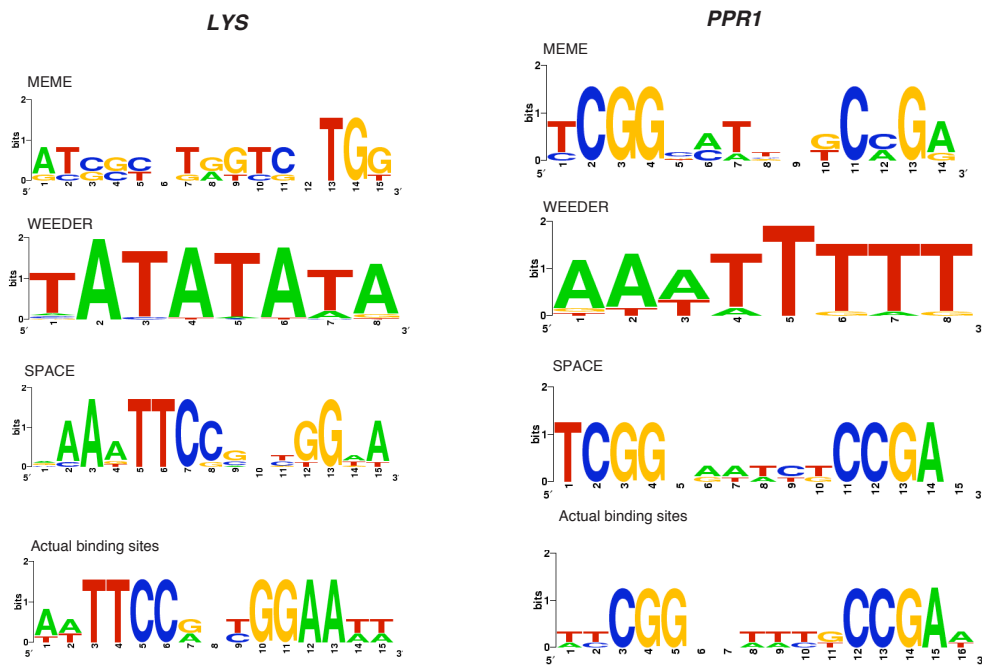
TF		Motif	RANK	nSn	nPPV	nCC	nPC
GAL4P [65]	Actual	CGGRnnRCYnYnCaCCG					
	SPACE	CGGAnGACTnnnnTCCG	1	0.80	0.55	0.48	0.65
	MEME	GGCGACTCTCGTCCGTGCG	1	0.89	0.76	0.70	0.82
	Weeder	TTTTTTTTTTTA	20	0.00	0.00	0.00	0.00
ARCA-P [81]	Actual	GTAA-(6n)-GTAA					
	SPACE	GTTA-(5n)ATGTTA	1	0.80	0.59	0.52	0.68
	MEME	AAGCAGGC	20	0.00	0.00	0.00	0.00
	Weeder	TTAACA	1	0.30	0.23	0.15	0.25
ECB [68] [132]	Actual	TACcNAATtnGGTAA					
	SPACE	TTACnnAATtnGGAA	1	0.70	0.58	0.46	0.61
	MEME	GAAATTCCTAATTAGGAAA	1	0.63	0.55	0.41	0.26
	Weeder	TTTTTT	1	0.26	0.87	0.25	0.46
CAT8 [139]	Actual	CGG-(6n)-GGA					
	SPACE	CGGA-(5n)GGAAT	1	0.74	0.52	0.44	0.62
	MEME	CGGATAAAATCGG	5	0.62	0.31	0.26	0.42
	Weeder	ATTATTATTATT	20	0.00	0.00	0.00	0.00
HAP1 [121] [130]	Actual	CGG-(3n)-TAnCGG-(3n)-TA					
	SPACE	CCGgnVTtnCGGH	2	0.67	0.67	0.50	0.66
	MEME	GGCGCCGGGATTTACCGGG	3	0.67	0.25	0.22	0.39
	Weeder	TTTTTTTC	20	0.00	0.00	0.00	0.00
LEU3 [130]	Actual	RCCGgnCCGGY					
	SPACE	CCGgnCCGGCT	1	0.85	0.23	0.22	0.43
	MEME	GCGGAAAA	20	0.00	0.00	0.00	0.00
	Weeder	AGAAAAAG	20	0.00	0.00	0.00	0.00
LYS [10]	Actual	WWWTCCRnYGGAWWW					
	SPACE	AATTCcGnnGGAA	4	0.62	0.59	0.43	0.60
	MEME	TTTTCCAGCGGAATT	3	0.50	0.42	0.29	0.43
	Weeder	TATATAAA	20	0.00	0.00	0.00	0.00
PPR [121]	Actual	WYCGGnnWYKCCGAW					
	SPACE	TCGG-(6n)-GCCGAAG	1	0.88	0.75	0.68	0.81
	MEME	TCGGCATTCCCGGA	1	0.78	0.68	0.57	0.72
	Weeder	AAATTTT	14	0.12	0.07	0.04	0.07
PUT3 [121]	Actual	YCGGnAnGCGnAnnnCCGA					
	SPACE	TCGGGA-(7n)-TCCG	1	0.89	0.76	0.69	0.81
	MEME	GCTAACTGGGAACCTAAC	1	0.51	0.50	0.34	0.48
	Weeder	ACTCCAGAAG	7	0.42	0.73	0.36	0.52
AVERAGE	SPACE			0.77	0.59	0.50	0.66
	MEME			0.51	0.39	0.31	0.39
	Weeder			0.12	0.21	0.09	0.14

Table 2.2: Comparison of SPACE, MEME and Weeder on real spaced biological data where motif contain spacers (the first motif among the top 20 that gives a better nSn and $nPPV$ than motif of Rank 1 is used for comparison).

In the following figures, we exhibit the conservation of the binding sites found by MEME, Weeder and SPACE on the above real biological datasets:







Synthetic Datasets

We consider 4 synthetic test cases for spaced motifs using randomly created sequences with the base pairs uniformly distributed. For each case, we create 3 datasets, each containing 10 sequences of length 300bp. We run SPACE and report the *averaged* performance. For each dataset the motifs are implanted in 5 of them at random positions. And the motifs are as follow:

1. A 7bp length motif with no spacer and 1 mismatch.
2. A 15bp length motif containing 2 segments of length 5 and 7 with a spacer of length 3, with 1 mismatch for each segment.
3. A 21bp length motif containing 3 segments of length 5 with spacers of length 3, with 1 mismatch for each segment.
4. A 15bp length motif containing 2 segments of length 4 and 5 bp with a spacer of length 6, with 1 mismatch for each segment.

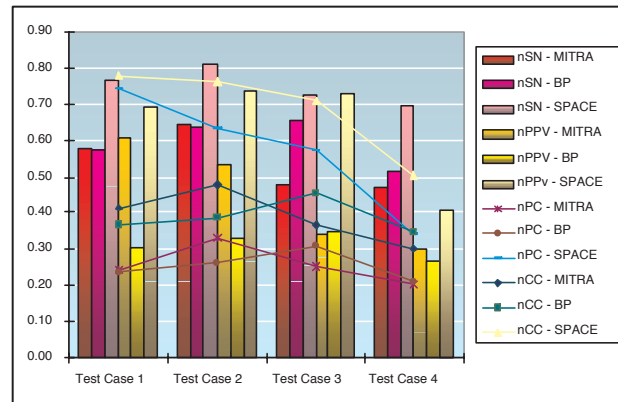


Figure 2.5: Comparison of MITRA, BioProspector (denoted BP) and SPACE averaged performance on 4 motif finding problems.

Figure 2.5 shows the averaged performance of SPACE, MITRA, and BioProspector on the synthetic datasets with Table 2.3 giving the detailed statistics

on one particular dataset of each test case. It also shows that SPACE performs better than the other tools over all four measures. The result is consistent with the one for real biological datasets.

Problem	Motif	RANK	nSn	nPPV	nPC	nCC
1. Actual	TGGGTAC					
SPACE	GGGTACC	3	0.83	0.72	0.75	0.82
MITRA	GGTACCCn	5	0.57	0.64	0.33	0.57
BP	GGGTACC	1	0.62	0.32	0.27	0.44
2. Actual	CCTGTnnnAGTTGTC					
SPACE	CCTGTnnTAGTTG	1	0.81	0.76	0.65	0.78
MITRA	CnTGTACTIONGTT	2	0.67	0.68	0.29	0.44
BP	CCTGTnnnACTTGTT	2	0.67	0.31	0.27	0.44
3. Actual	ATCGTnnnTGACCnnnCTTTC					
SPACE	TCGTnnnTGACnnnnnTTTC	1	0.76	0.66	0.55	0.69
MITRA	ATCCTnGnTGAC	1	0.49	0.38	0.27	0.39
BP	ATCGTnnnnnnnnnnCTTTC	1	0.71	0.38	0.33	0.50
4. Actual	CGGCnnnnnnTCTAA					
SPACE	TTCGGYnnnnTGTC	1	0.71	0.39	0.33	0.50
MITRA	CGGCnAAGnGTC	3	0.50	0.24	0.13	0.20
BP	CGTAnnnnnnTCTAA	1	0.33	0.18	0.13	0.22

Table 2.3: Performance of SPACE, MITRA and BioProspector (denoted BP) on 4 types of synthetic data (one dataset each).

The averaged evaluation statistics for both MITRA and SPACE can be in the table below.

Datasets	MITRA				SPACE			
	nSn	nPPV	nPC	nCC	nSn	nPPv	nPC	nCC
Test Case 1	0.58	0.61	0.24	0.41	0.77	0.69	0.74	0.78
Test Case 2	0.65	0.53	0.33	0.48	0.81	0.74	0.64	0.76
Test Case 3	0.48	0.34	0.25	0.37	0.73	0.73	0.57	0.71
Test Case 4	0.47	0.30	0.20	0.30	0.70	0.40	0.34	0.51

Table 2.4: Comparison of SPACE and MITRA averaged performance on 4 motif finding problems.

The averaged evaluation statistics for both BIOPROSPECTOR and SPACE can be in the table below.

Datasets	BIOPROSPECTOR				SPACE			
	nSn	nPPV	nPC	nCC	nSn	nPPv	nPC	nCC
Test Case 1	0.57	0.30	0.23	0.37	0.77	0.69	0.74	0.78
Test Case 2	0.64	0.33	0.26	0.38	0.81	0.74	0.64	0.76
Test Case 3	0.66	0.35	0.31	0.46	0.73	0.73	0.57	0.71
Test Case 4	0.52	0.26	0.21	0.35	0.70	0.40	0.34	0.51

Table 2.5: Comparison of SPACE and BIOPROSPECTOR averaged performance on 4 motif finding problems.

The detailed statistics for 3 datasets in each motif-finding problems can be found in the following tables.

Set	Known Motif	MITRA				SPACE				
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC
1	TGGGTAC	GGTACCn (5)	0.57	0.64	0.33	0.57	0.83	0.72	0.75	0.82
2	CCGAAAG	GAACGnn (7)	0.50	0.57	0.16	0.28	0.80	0.70	0.78	0.81
3	TGTTTTCC	TTTTCCA (4)	0.67	0.63	0.23	0.39	0.67	0.66	0.70	0.71
A		Average	0.58	0.61	0.24	0.41	0.77	0.69	0.74	0.78

Table 2.6: Detailed comparison of SPACE and MITRA performance on 3 motif finding Test Case 1.

Set	Known Motif	MITRA				SPACE				
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC
1	CCTGTnnnAGTTGTTC	CnTGTACTnGTTT (2)	0.67	0.68	0.29	0.44	0.81	0.76	0.65	0.78
2	GTTGTnnnAATACTC	GTTGTTGTTATA (1)	0.60	0.59	0.42	0.56	0.80	0.71	0.61	0.74
3	CCCCnnnACATTC	CCGcCnATAnAT (8)	0.67	0.33	0.28	0.44	0.83	0.74	0.65	0.77
B		Average	0.65	0.53	0.33	0.48	0.81	0.74	0.64	0.76

Table 2.7: Detailed comparison of SPACE and MITRA performance on 3 motif finding Test Case 2.

Set	Known Motif	MITRA				SPACE				
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC
1	ATCGTnnnTGACnnnCTTTC	ATCCnGnTGAC (1)	0.49	0.38	0.27	0.39	0.76	0.66	0.55	0.69
2	GAAAGCnnnCGAGGnnnGATCC	GAAnCTATCCnG (4)	0.47	0.32	0.24	0.35	0.62	0.73	0.50	0.65
3	GCCCAnnnGTTTAnnGATGA	ATTTAnnGGATG (1)	0.48	0.32	0.24	0.36	0.80	0.80	0.67	0.79
C		Average	0.48	0.34	0.25	0.37	0.73	0.73	0.57	0.71

Table 2.8: Detailed comparison of SPACE and MITRA performance on 3 motif finding Test Case 3.

Set	Known Motif	MITRA				SPACE				
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC
1	CGGC-6n-TCTAA	CGGCnAAGnGTC (3)	0.50	0.24	0.13	0.20	0.71	0.39	0.33	0.50
2	CAGT-6n-TAGAT	ACAGTCGAGCnT (6)	0.52	0.40	0.29	0.42	0.67	0.44	0.36	0.52
3	GGCA-6n-GGGTC	CAnTAGTTGGTT (5)	0.40	0.26	0.19	0.28	0.71	0.38	0.33	0.50
D		Average	0.47	0.30	0.20	0.30	0.70	0.40	0.34	0.51

Table 2.9: Detailed comparison of SPACE and MITRA performance on 3 motif finding Test Case 4.

Set	Known Motif	BIOPROSPECTOR					SPACE				
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC
1	TGGGTAC	GGGTACC (1)	0.62	0.32	0.27	0.44	GGGTACC (3)	0.83	0.72	0.75	0.82
2	CCGAACG	ACCAAAG (3)	0.60	0.30	0.24	0.37	mCGAAACGT (2)	0.80	0.70	0.78	0.81
3	TGTTTTCC	ATTTGTT (1)	0.50	0.29	0.19	0.29	GTTTCCA (2)	0.67	0.66	0.70	0.71
A		Average	0.57	0.30	0.23	0.37	Average	0.77	0.69	0.74	0.78

Table 2.10: Detailed comparison of SPACE and BIOPROSPECTOR performance on 3 motif finding Test Case 1.

Set	Known Motif	BIOPROSPECTOR					SPACE				
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC
1	CCTGTnnnAGTTGTTC	CCTGTnnnACTTGTTC (2)	0.67	0.31	0.27	0.44	CCTGTnnnTAGTTTC (1)	0.81	0.76	0.65	0.78
2	GTTGTnnnAATATCC	GTTGTnnnAATATCC (5)	0.51	0.30	0.24	0.37	TGTTTGT-5n-TACT (1)	0.80	0.71	0.61	0.74
3	CCCCnnnACAFTCC	CGCCnnnACAFTTC (1)	0.74	0.38	0.27	0.34	CCCCnnnACAFTTC (1)	0.83	0.74	0.65	0.77
B		Average	0.64	0.33	0.26	0.38	Average	0.81	0.74	0.64	0.76

Table 2.11: Detailed comparison of SPACE and BIOPROSPECTOR performance on 3 motif finding Test Case 2.

Set	Known Motif	BIOPROSPECTOR					SPACE				
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC
1	ATCGTnnnTGACGnnnCTTTC	ATCGT-11n-CTTTC (1)	0.71	0.38	0.33	0.50	TGGTnnnTGAC-5n-TTTC (1)	0.76	0.66	0.55	0.69
2	GAAGGnnnCGAGGnnnGATCC	GAACG-11n-CCATCC (1)	0.43	0.21	0.17	0.27	AwGC-4n-GAGGnnnGATC (3)	0.62	0.73	0.50	0.65
3	GCCCCAnnnGTTAnnnGATGA	GCCCCA-11n-GATGA (8)	0.83	0.45	0.42	0.60	GCCC-4n-GTTT-4n-GATG (1)	0.80	0.80	0.67	0.79
C		Average	0.66	0.35	0.31	0.46	Average	0.73	0.73	0.57	0.71

Table 2.12: Detailed comparison of SPACE and BIOPROSPECTOR performance on 3 motif finding Test Case 3.

Set	Known Motif	BIOPROSPECTOR					SPACE				
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC
1	CGGC-6n-TCTAA	CGTA-6n-TCTAA (7)	0.33	0.18	0.13	0.22	TTCGGy-4n-TGTC (1)	0.71	0.39	0.33	0.50
2	CAGT-6n-TAGAT	CAGT-6n-TCGTT (1)	0.62	0.36	0.29	0.45	CAGT-5n-GTAG (1)	0.67	0.44	0.36	0.52
3	GGCA-6n-GGGTC	GACA-6n-GGGCC (1)	0.60	0.25	0.21	0.37	GGCA-6n-GGGT (2)	0.71	0.38	0.33	0.50
D		Average	0.52	0.26	0.21	0.35	Average	0.70	0.40	0.34	0.51

Table 2.13: Detailed comparison of SPACE and BIOPROSPECTOR performance on 3 motif finding Test Case 4.

We also examine the performance comparison between SPACE and monad motif finders MEME and Weeder. The averaged evaluation statistics for both MEME and SPACE can be in the table below.

Datasets	MEME				SPACE			
	nSn	nPPV	nPC	nCC	nSn	nPPv	nPC	nCC
Test Case 1	0.10	0.54	0.10	0.21	0.77	0.69	0.74	0.78
Test Case 2	0.07	0.56	0.07	0.14	0.81	0.74	0.64	0.76
Test Case 3	0.07	0.68	0.07	0.12	0.73	0.73	0.57	0.71
Test Case 4	0.04	0.33	0.04	0.10	0.70	0.40	0.34	0.51

Table 2.14: Comparison of SPACE and MEME averaged performance on 4 motif finding problems.

The averaged evaluation statistics for both WEEDER and SPACE can be in the table below.

Datasets	WEEDER				SPACE			
	nSn	nPPV	nPC	nCC	nSn	nPPv	nPC	nCC
Test Case 1	0.37	0.36	0.25	0.35	0.77	0.69	0.74	0.78
Test Case 2	0.19	0.56	0.15	0.26	0.81	0.74	0.64	0.76
Test Case 3	0.15	0.65	0.13	0.25	0.73	0.73	0.57	0.71
Test Case 4	0.15	0.30	0.11	0.14	0.70	0.40	0.34	0.51

Table 2.15: Comparison of SPACE and WEEDER averaged performance on 4 motif finding problems.

The detailed statistics for 3 datasets in each motif-finding problems can be found in the following tables.

Set	Known Motif	MEME				SPACE					
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC
1	TGGGTAC	CACACC (19)	0.12	0.62	0.11	0.24	GGGTACC (3)	0.83	0.72	0.75	0.82
2	CCGAACG	CCGAAC (20)	0.18	1.00	0.18	0.40	mCGAACGT (2)	0.80	0.70	0.78	0.81
3	TGTTTTCC	GGCACGA (20)	0.00	0.00	0.00	0.00	GTTTTCCA (2)	0.67	0.66	0.70	0.71
A		Average	0.10	0.54	0.10	0.21	Average	0.77	0.69	0.74	0.78

Table 2.16: Detailed comparison of SPACE and MEME performance on 3 motif finding Test Case 1.

Set	Known Motif	MEME				SPACE					
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC
1	CCTGTnnAGTTGTTC	GCCTGGAGTG (13)	0.05	0.25	0.04	0.04	CCTGTnnTAGTTG (1)	0.81	0.76	0.65	0.78
2	GTGTnnnAATACTC	TACCAACGAC (1)	0.06	0.42	0.06	0.10	TGTTGT-6n-TACT (1)	0.80	0.71	0.61	0.74
3	CCCCnnnACATTC	CCCCCC (3)	0.10	1.00	0.10	0.29	CCCCnnnACATTC (1)	0.83	0.74	0.65	0.77
B		Average	0.07	0.56	0.07	0.14	Average	0.81	0.74	0.64	0.76

Table 2.17: Detailed comparison of SPACE and MEME performance on 3 motif finding Test Case 2.

Set	Known Motif	MEME				SPACE					
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC
1	ATCGTnnnTGACGnnnCTTTTC	GGCACC (9)	0.10	1.00	0.10	0.29	TCGTnnnTGAC-5n-TTTC (1)	0.76	0.66	0.55	0.69
2	GAAGCnnnCGAGGnnnGATCC	CCCCCA (2)	0.08	0.60	0.08	0.00	AwCC-4n-GAGGnnnGATC (3)	0.62	0.73	0.50	0.65
3	GCCCCnnnGTTTAnnnGATGA	CCCCGCC (1)	0.04	0.44	0.04	0.07	GCCC-4n-GTTT-4n-GATG (1)	0.80	0.80	0.67	0.79
C		Average	0.07	0.68	0.07	0.12	Average	0.73	0.73	0.57	0.71

Table 2.18: Detailed comparison of SPACE and MEME performance on 3 motif finding Test Case 3.

Set	Known Motif	MEME				SPACE					
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC
1	CGGC-6n-TCTAA	TAGCCGC (20)	0.00	0.00	0.00	0.00	TTCCGGy-4n-FGTC (1)	0.71	0.39	0.33	0.50
2	CAGT-6n-TAGAT	AAGCCC (20)	0.00	0.00	0.00	0.00	CAGT-5n-GTAG (1)	0.67	0.44	0.36	0.52
3	GGCA-6n-GGGTC	GCCCCC (15)	0.11	1.00	0.11	0.30	GGCA-6n-GGGT (2)	0.71	0.38	0.33	0.50
D		Average	0.04	0.33	0.04	0.20	Average	0.70	0.40	0.34	0.51

Table 2.19: Detailed comparison of SPACE and MEME performance on 3 motif finding Test Case 4.

Set	Known Motif	WEEDER						SPACE					
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC		
1	TGGGTAC	GTAGCC (1)	0.50	0.57	0.36	0.52	GGGTACC (3)	0.83	0.72	0.75	0.82		
2	CCGAACG	GCAGAGGCTACT (11)	0.61	0.52	0.39	0.54	mCGAACCT (2)	0.80	0.70	0.78	0.81		
3	TGTTTCC	TGGTGC (1)	0.00	0.00	0.00	0.00	GTTTCCA (2)	0.67	0.66	0.70	0.71		
A		Average	0.37	0.36	0.25	0.35	Average	0.77	0.69	0.74	0.78		

Table 2.20: Detailed comparison of SPACE and WEEDER performance on 3 motif finding Test Case 1.

Set	Known Motif	WEEDER						SPACE					
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC		
1	CCTGTnnnAGTTGTTC	AGAACTGTGCTA (2)	0.12	0.69	0.11	0.23	CCTGTnnTAGTTG (1)	0.81	0.76	0.65	0.78		
2	GTTGTnnnAAATACTC	AAATCTCA (2)	0.35	0.56	0.27	0.42	TGTTGT-5n-TACT (1)	0.80	0.71	0.61	0.74		
3	CCCCAnnnACATTC	ACAATCCT (2)	0.09	0.44	0.08	0.12	CCCCGnnnACAPT (1)	0.83	0.74	0.65	0.77		
B		Average	0.19	0.56	0.15	0.26	Average	0.81	0.74	0.64	0.76		

Table 2.21: Detailed comparison of SPACE and WEEDER performance on 3 motif finding Test Case 2.

Set	Known Motif	WEEDER						SPACE					
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC		
1	ATCGTnnnTGACGmnnGTTTC	CCCGGGTGAC (3)	0.15	1.00	0.15	0.36	TGGTnnnTGAC-5nTTTC (1)	0.76	0.66	0.55	0.69		
2	GAAGCnnnCGAGGnnnGATCC	TTTTGA (1)	0.14	0.55	0.12	0.23	AwGC-4n-GAGGnnnGATC (3)	0.62	0.73	0.50	0.65		
3	GCCCAnnnGTTTAnnnGATGA	GGTCAT (1)	0.17	0.39	0.13	0.17	GCCCC-4n-GTTT-4n-GATG (1)	0.80	0.80	0.67	0.79		
C		Average	0.15	0.65	0.13	0.25	Average	0.73	0.73	0.57	0.71		

Table 2.22: Detailed comparison of SPACE and WEEDER performance on 3 motif finding Test Case 3.

Set	Known Motif	WEEDER						SPACE					
		Predicted (Rank)	nSn	nPPV	nPC	nCC	Predicted (Rank)	nSn	nPPV	nPC	nCC		
1	CGGC-6n-TCTAA	GAAGTG (19)	0.10	0.18	0.07	0.07	TTCGGGy-4n-TGTC (1)	0.71	0.39	0.33	0.50		
2	CAGT-6n-TAGAT	GTACAC (4)	0.19	0.38	0.14	0.21	CAGT-5n-GTAG (1)	0.67	0.44	0.36	0.52		
3	GGCA-6n-GGGTC	GCCGGCGTGT (2)	0.15	0.35	0.12	0.15	GGCA-6n-GGGT (2)	0.71	0.38	0.33	0.50		
D		Average	0.15	0.30	0.11	0.14	Average	0.70	0.40	0.34	0.51		

Table 2.23: Detailed comparison of SPACE and WEEDER performance on 3 motif finding Test Case 4.

2.6.2 Results on Datasets with Monad Motifs

We are also interested in the performance of SPACE for motifs without spacers. We have performed two sets of experiments, one on Tompa’s benchmark datasets and the other on 10 real biological datasets.

Tompa’s Benchmark Data

Tompa’s benchmark dataset has been constructed based on real transcription factor binding sites drawn from four different organisms [135]. It consists of 56 datasets in total. The number of sequences ranges from 1-35 and the sequence lengths are up to 3000 bp. In this assessment, following Tompa’s approach, the motif ranked number 1 by the algorithm is used for comparison.

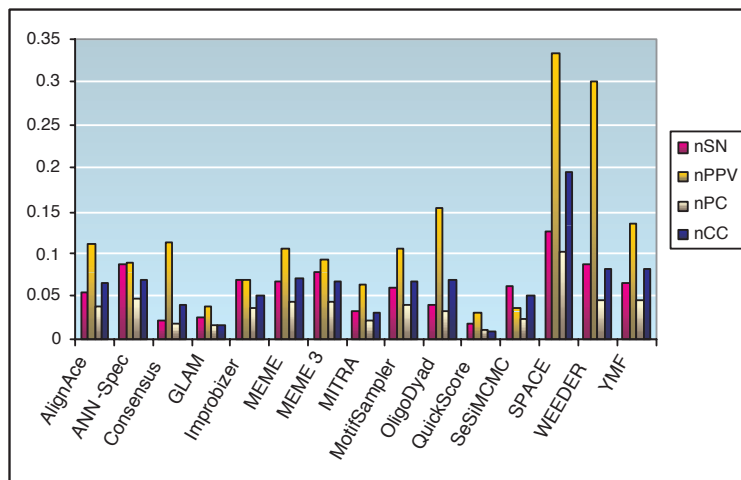


Figure 2.6: Comparison between SPACE with 13 other motif discovery tools.

The performance of SPACE averaged over all datasets is shown in Figure 2.6. SPACE performs better than other tools based on the comparison measures ⁶.

⁶In our comparison, we did not include the new motif finder, MotifSeeker [102]. The experiments in their paper are based on a different subset of Tompa’s datasets, so a direct comparison is not appropriate.

As an example, we show the binding sites (see Figure 2.7) identified (in green) by our algorithm and Weeder on the dataset *hm17g* together with the actual binding sites (in blue). Weeder is reported to perform the best among other tools in this dataset [135]. Similar to Weeder, SPACE is able to identify almost all actual binding sites.

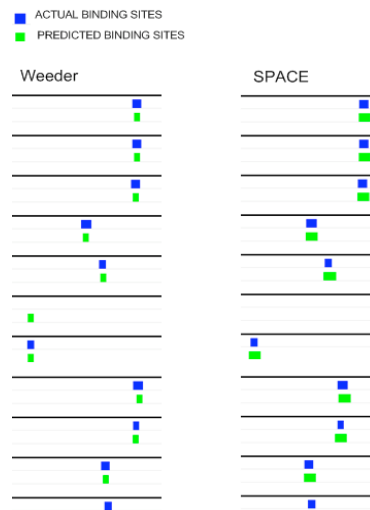


Figure 2.7: Binding sites without gaps reported by SPACE in *hm17g* (*human*), with $nSn = 0.90$, $nPPV = 0.72$, $nPC = 0.67$ and $nCC = 0.80$. Weeder with $nSn = 0.61$, $nPPV = 0.89$, $nPC = 0.57$ and $nCC = 0.73$.

We also analyzed the performance of SPACE across the four organisms. Figure 2.8 shows the average performance of SPACE for each organism, compared with the best algorithm among the other tools for the respective organism. The figure shows that the performance of SPACE is similar to that of the best performing algorithm for each organism. On the other hand, the averaged performance of SPACE for all four organisms is better than other tools (Figure 2.6), indicating that SPACE is more robust and organism independent.

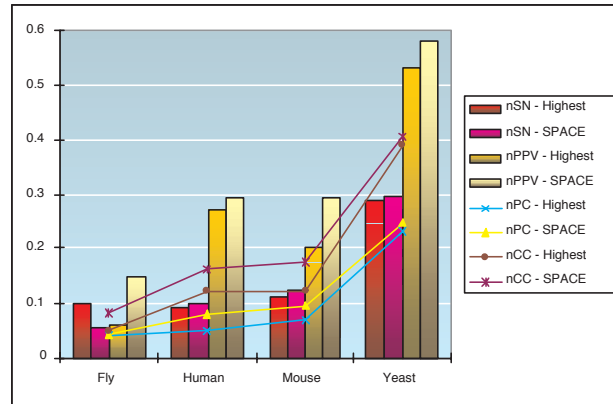


Figure 2.8: Comparison of SPACE and best performing algorithms on 4 types of organisms.

Real Biological Datasets

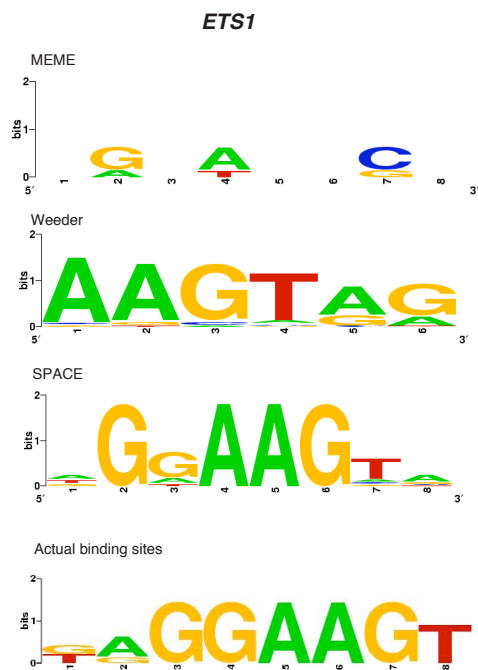
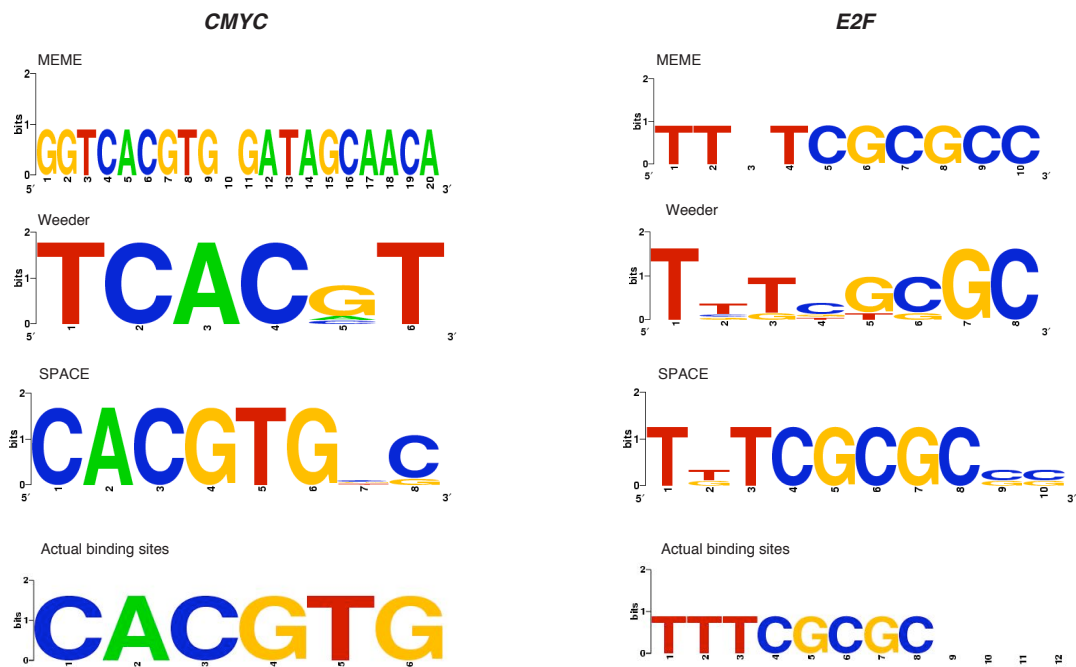
We also performed experiments on 10 real biological datasets whose binding sites are known to be monads from literature. Comparison is done with Weeder [100] and MEME [8], both of which are well known monad motif finding algorithms. We set MEME to use Two-Component mixture mode and find motifs of length ranging from 8 upto 20 bp. And for Weeder we use the *large* mode.

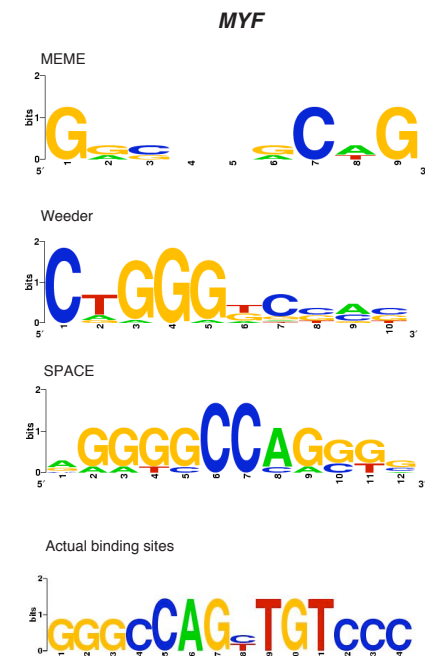
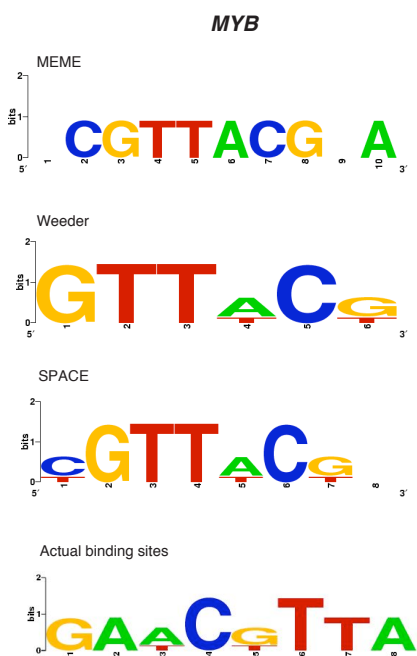
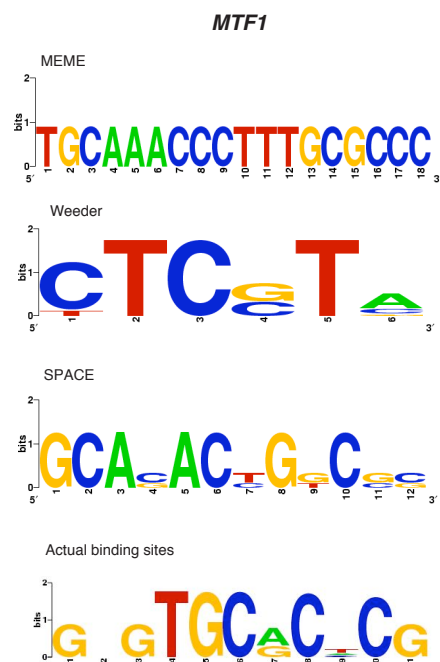
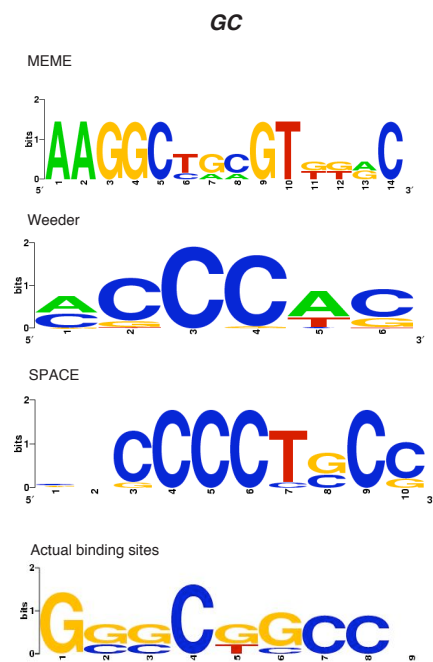
Table 2.24 summarizes the comparison results. From the table, for sensitivity, MEME shows a better performance than Weeder while SPACE is better than MEME. The reason for SPACE to have a higher sensitivity is due to the submotif modeling. Since the measure nSn focuses on predicting the binding sites, if there is a region in the motif that is not strongly conserved over all binding sites, the use of submotifs may still be able to identify most of these binding sites based on the regions that are strongly conserved, thus predicting more true binding sites. However, it does not mean that missing these binding sites, the software will predict a wrong motif pattern.

Unlike the case for spaced motifs, SPACE is not a clear winner for all the mea-

TF		Motif	RANK	nSn	nPPV	nCC	nPC
AP2A [33, 75]	Actual	GCCGGGGKSG					
	SPACE	CCAGGGAG	1	0.75	0.50	0.43	0.60
	MEME	GCCCCCCC	5	0.38	0.56	0.29	0.45
	Weeder	CCCCACCC	3	0.38	0.33	0.21	0.34
CAAT [141]	Actual	AAGCCAATTAGGCC					
	SPACE	GAAGCCAATTAG	1	0.51	0.60	0.38	0.54
	MEME	CGAAGCAA	2	0.08	0.67	0.08	0.20
	Weeder	GCCAAT	1	0.63	0.78	0.54	0.70
CJUN [52]	Actual	ATTATTCACHTCATC					
	SPACE	CATTWCCTCA	1	0.64	0.73	0.52	0.68
	MEME	CATTACCTCA	2	0.62	0.81	0.55	0.71
	Weeder	CATTACCTCA	3	0.62	0.81	0.55	0.71
CMYC [56]	Actual	CACGTG					
	SPACE	CACGTGCC	1	1.00	0.60	0.60	0.77
	MEME	GGTCACGTGGGATAGCAACA	1	1.00	0.27	0.27	0.71
	Weeder	TCACGT	1	0.71	0.71	0.56	0.71
E2F [149]	Actual	TTTCGCGC					
	SPACE	TTTCGCGCC	1	0.91	0.81	0.80	0.88
	MEME	TTGTCGCGCC	4	1.00	0.82	0.82	0.90
	Weeder	TTTCGCGC	2	0.64	0.91	0.60	0.75
ETS1 [33]	Actual	KAGGAAGT					
	SPACE	AGGAAGTA	1	0.76	0.65	0.54	0.70
	MEME	GGTATTCA	3	0.62	0.56	0.42	0.58
	Weeder	AAGTAG	1	0.40	0.48	0.28	0.43
GC [33]	Actual	GGGCGGCC					
	SPACE	GCCCCTGCC	1	0.56	0.60	0.41	0.57
	MEME	AAGGCTGCGTGGAC	1	0.57	0.27	0.22	0.38
	Weeder	ACCCAC	5	0.62	0.71	0.50	0.66
MTF1 [13]	Actual	GGGTGCACTCG					
	SPACE	GCACACTGGC	3	0.71	0.36	0.31	0.50
	MEME	TGCAAACCCTTTGGCGCCC	6	0.65	0.29	0.25	0.42
	Weeder	CTCGTA	9	0.38	0.43	0.25	0.39
MYB [75]	Actual	GAACGTTA					
	SPACE	CGTTACG	1	0.71	0.50	0.42	0.59
	MEME	ACGTTACGAA	9	1.00	0.55	0.55	0.73
	Weeder	GTTACG	1	0.57	0.57	0.57	0.40
MYF [75]	Actual	GGGCCAGTTGTCCC					
	SPACE	GGGGCCAGG	2	0.54	0.71	0.44	0.61
	MEME	GGCAAGCAG	5	0.39	1.00	0.39	0.62
	Weeder	CTGGGTGAC	1	0.47	0.64	0.37	0.53
AVERAGE	SPACE			0.71	0.61	0.49	0.64
	MEME			0.63	0.58	0.38	0.46
	Weeder			0.54	0.64	0.43	0.58

Table 2.24: Comparison of SPACE, MITRA and Weeder on monads in real biological datasets (the first motif among the top 20 that gives a better nSn and $nPPV$ than motif of Rank 1 is used for comparison).

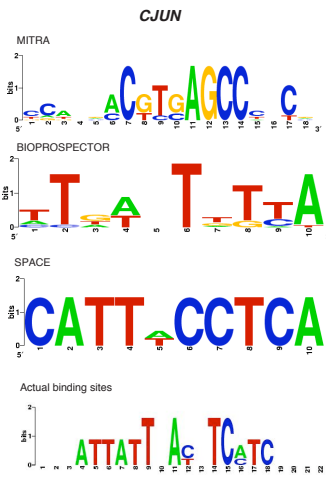
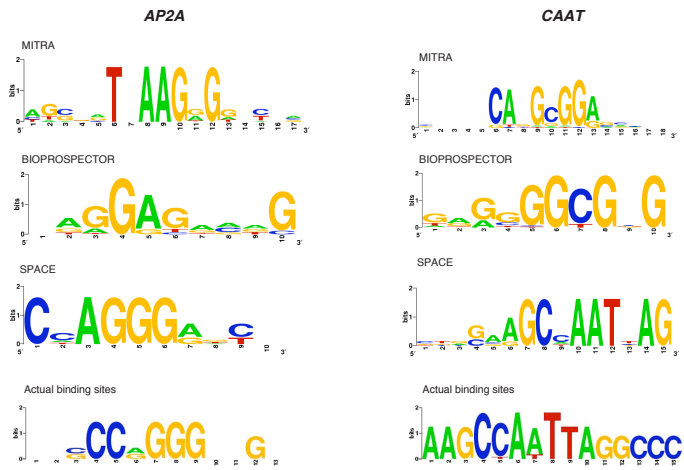


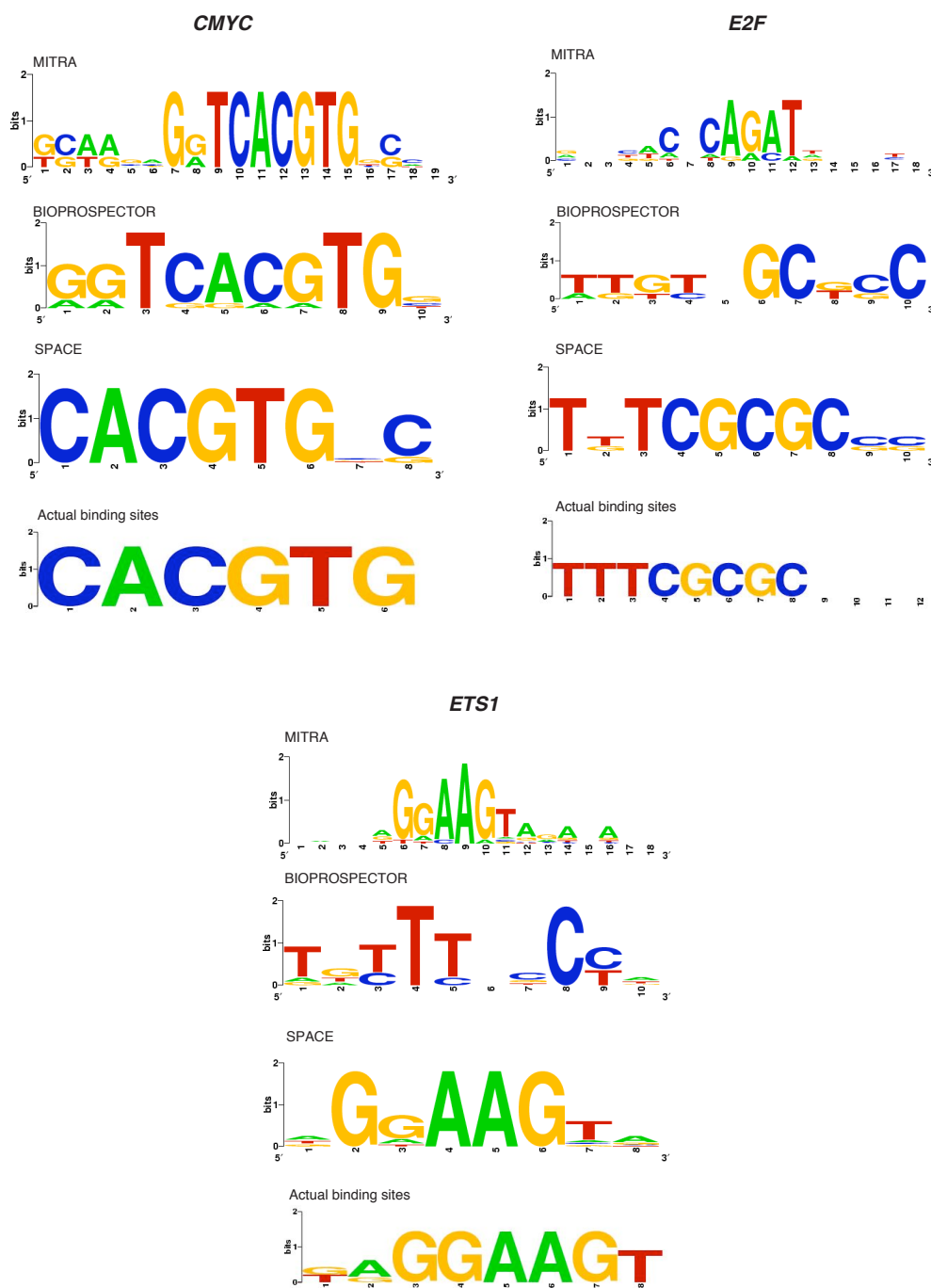


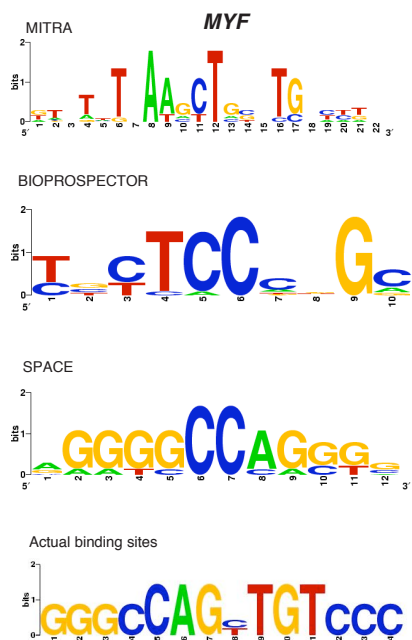
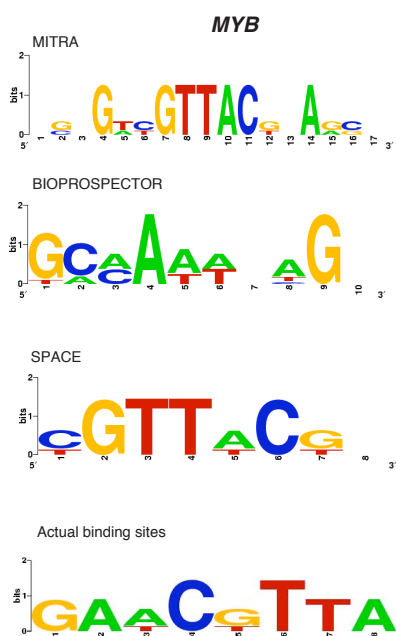
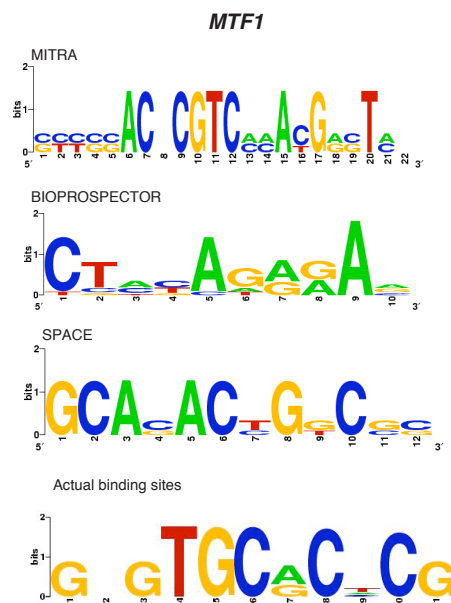
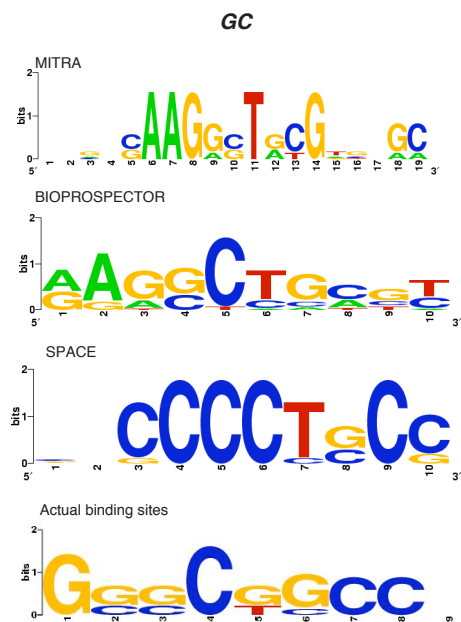
We also compare the performance of MITRA and BioProspector on these monads datasets.

TF		Motif	RANK	nSn	nPPV	nCC	nPC
AP2A [75] [33]	Actual	GCCGGGGKSG					
	SPACE	CCAGGGAG	1	0.75	0.50	0.43	0.60
	MITRA	TnAGnG	1	0.12	0.22	0.09	0.13
	BioProspector	AAGGAGACAG	1	0.50	0.20	0.17	0.30
CAAT [141]	Actual	AAGCCAATTAGGCC					
	SPACE	GAAGCCAATTAG	1	0.51	0.60	0.38	0.54
	MITRA	CAnGCGGA	1	0.36	0.21	0.15	0.26
	BioProspector	GAGGGGCGCG	1	0.75	0.16	0.15	0.33
CJUN [52]	Actual	ATTATTCACHCATC					
	SPACE	CATTWCCTCA	1	0.64	0.73	0.52	0.68
	MITRA	ACGTGAGC	1	0.11	0.16	0.07	0.09
	BioProspector	TAAAAATCA	1	0.38	0.38	0.23	0.35
CMYC [56]	Actual	CACGTG					
	SPACE	CACGTGCC	1	1.00	0.60	0.60	0.77
	MITRA	AGGTnACGT	1	1.00	0.29	0.29	0.53
	BioProspector	GGTCACGTGG	1	1.00	0.32	0.32	0.56
E2F [149]	Actual	TTCGCGC					
	SPACE	TTCGCGCC	1	0.91	0.81	0.80	0.88
	MITRA	CnCAGATn	2	0.22	0.13	0.09	0.15
	BioProspector	TTGTCGCGCC	2	1.00	0.47	0.47	0.68
ETS1 [33]	Actual	KAGGAAGT					
	SPACE	AGGAAGTA	1	0.76	0.65	0.54	0.70
	MITRA	GGAAGTAn	1	1.00	0.38	0.38	0.60
	BioProspector	TGGGAAAACA	1	0.67	0.24	0.21	0.38
GC [33]	Actual	GGGCGGCC					
	SPACE	GCCCCTGCC	1	0.56	0.60	0.41	0.57
	MITRA	AAGGnTACG	4	0.57	0.20	0.17	0.32
	BioProspector	AAGGCTCGCT	1	0.65	0.20	0.18	0.35
MTF1 [13]	Actual	GGGTGCACTCG					
	SPACE	GCACACTGGC	3	0.71	0.36	0.31	0.50
	MITRA	ACACGTCCCACG	1	0.44	0.17	0.14	0.26
	BioProspector	CTACAGAGAG	1	0.57	0.21	0.18	0.33
MYB [75]	Actual	GAACGTTA					
	SPACE	CGTTACG	1	0.71	0.50	0.42	0.59
	MITRA	nGTTACn	4	1.00	0.32	0.32	0.55
	BioProspector	GCCAATGAGG	2	0.43	0.16	0.13	0.24
MYF [75]	Actual	GGGCCAGTTGTCCC					
	SPACE	GGGGCCAGG	2	0.54	0.71	0.44	0.61
	MITRA	TnAAGCTGnATG	1	0.13	0.09	0.06	0.08
	BioProspector	TGTCGCCGGC	1	0.49	0.38	0.27	0.41
AVERAGE	SPACE			0.71	0.61	0.49	0.64
	MITRA			0.50	0.22	0.18	0.30
	BioProspector			0.64	0.27	0.23	0.39

Table 2.25: Comparison of SPACE, MITRA and BioProspector on real monad biological data (the first motif among the top 20 that gives a better nSn and $nPPV$ than motif of Rank 1 is used for comparison).







2.7 Conclusions

In this chapter we have proposed a new approach for finding spaced motifs based on the notion called submotifs. We developed a novel motif-finding algorithm SPACE that detects the target motif by first finding submotifs and then strategically compositing them using an efficient frequent submotif pattern mining approach. In finding motif with generic spacers, this framework provides the following novelties : the spacers could appear in more than two parts of the motif and their lengths need not be fixed. In experiments on real biological datasets, synthetic datasets and Tompa’s motif assessment benchmarks, we observed that our algorithm performs better than existing tools for spaced motifs with improvements in both sensitivity and specificity and for monads, SPACE performs as well as other tools.

However, based on the submotif notion we define, we implicitly assume that the mismatches are uniformly distributed in the motif *instances*. If that is not the case, SPACE may fail to capture these instances, and thus may miss the motif or the regions of the motif that contain these mismatches. On the other hand, in real biological datasets, it seems that mismatches are usually not clustered for most of the motif instances. Hence, SPACE can perform well in most cases.

Variance Based Ensemble Method for Integrating Generic Motif Finders

Although many tools have been developed, little knowledge is known about which motif finder should be used for a particular data set. This posits a real challenge for practitioners and biological researchers because of two reasons. Firstly, the performance of individual motif finders is unimpressive overall as shown in Tompa et.al.'s evaluation [135] where even the best performing algorithm has sensitivity < 0.13 and precision < 0.35 . The same study also shows that no motif finder performs consistently well for all datasets. It is not clear how to select the correct motif finder given a particular dataset. Secondly, even for a specific motif finder, it is not straightforward to decide how many motifs one should consider in the output list, since motifs of lower rank may be useful to reveal real binding sites. For these reasons, some works have hinted the possible improvement made by combining the results from different motif finders [54, 58, 83] and consequently several ensemble methods have been developed (e.g. SCOPE [27], EMD [59], BEST [36, 63], WebMotifs [49, 114]). Most of these methods (referred as classical

ensemble methods) use the principle of picking one motif based on some forms of scoring function from the collection of motifs returned by the individual motif finders. This however, suggests that their performance is bounded above by the performance of the best individual motif finder in the respective data set.

In fact, since different motif finders have different limits in their methods of finding statistically overrepresented motifs and different definitions of what constitute a motif [83, 135] it is unlikely that a single motif finder can predict all true binding sites correctly. Covering most of the true binding sites requires the predictions of multiple motif finders. Hence, selecting only the best motif may not be sufficient to get the best accuracy. A refined integration of motifs is crucial to significantly improve the predictive power in motif discovery.

Following this line of research, we have developed a novel ensemble method, called MotifVoter (Figure 3.1), that identifies the best motif by integrating the sites reported by several generic motif finders. The idea of MotifVoter is to select a subset of high confidence motifs given by multiple motif finders. And from these motifs we further refine their instances to form final motif. MotifVoter consists of two stages. Firstly, a novel discriminative clustering method is used to determine a cluster of motifs with the highest density. In other words, we identify the cluster which maximizes the density measure while at the same time minimizes this measure for its complement cluster. We expect the true sites will be identified by more than one motif finders. Thus, we also ensure the cluster of motifs should be contributed by as many motif finders as possible (called the constraint attribute). Secondly, among the motifs in the selected cluster, we select a subset of high confidence sites to form the final the motif.

MotifVoter improves sensitivity of the predictions since the final motif is generated from all the high confidence sites given by multiple motif finders. The

previous approaches only generate motif given by one motif finder only. It also improves specificity because the two clustering attributes (discriminative and constraint) effectively filters out all the false motifs. Furthermore, the second stage ensure that MotifVoter to retain high confidence sites by removing false sites.

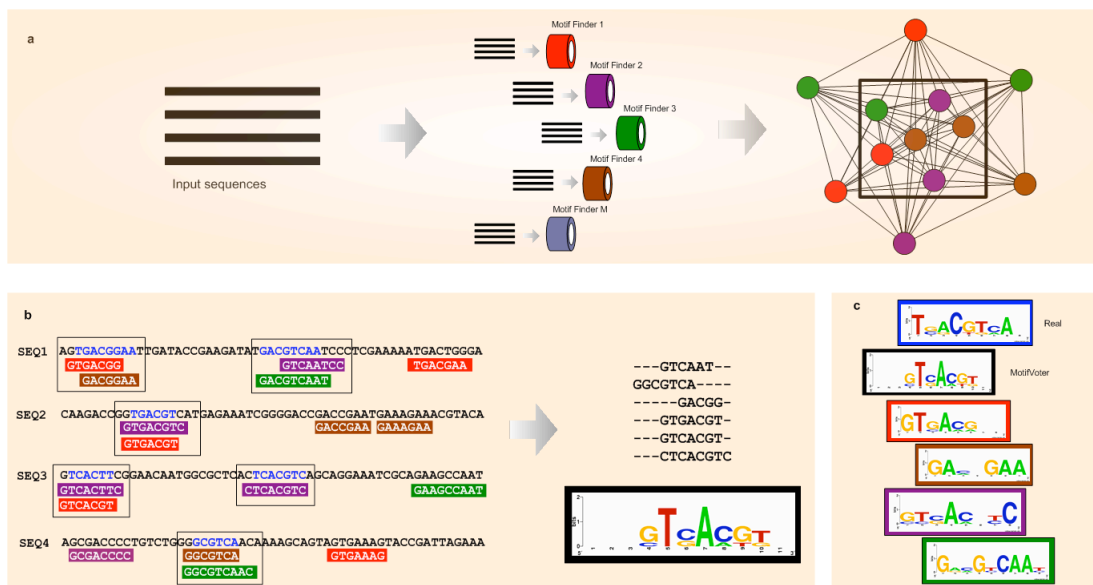


Figure 3.1: MotifVoter’s approach.

Figure 3.1 depicts the procedure involved in MotifVoter. Figure 3.1 (a) We apply M motif finders on the input sequences. Each motif finder reports a set of motifs and their respective sets of instances. The pairwise similarities among predicted motifs can be visualized in a graph where each node represents a motif and the similarity between two motifs is represented by a weighted edge (we use a shorter edge to represent a pair of motifs that are more similar). We expect the clustered motifs to approximate the real motif while the rest of the motifs are spurious. Hence, Stage 1 aims to find a cluster such that (1) motifs predicted by multiple motif finders and (2) the motifs are close (similar) to one another, but far away (different) from other. We employ a variance-based statistical approach

to achieve this effect. (b) The diagram shows the ideas behind the second stage of MotifVoter. Binding sites in blue are real binding sites. The remaining colors are used to illustrate the binding sites predicted by 4 other motif finders. As shown by the framed binding sites, MotifVoter can discover more true binding sites compare to individual motif finders. Furthermore, using a confidence measure, MotifVoter is also able to detect the true binding site which can only be discovered by one motif finder. (c) In this figure we exhibit the weblogos of real bindings sites, binding sites predicted by MotifVoter and other 4 stand alone motif finders. Although the weblogos of the 4 stand alone motif finders are similar to the real ones, the binding sites predicted by those 4 stand alone motif finders still contain false binding sites (see Figure 3.1b). However, MotifVoter can effectively filter the spurious binding sites and give a better approximation of the true motif.

We have evaluated MotifVoter and compared it with other 17 motif finders and four most recent ensemble methods. The results show that MotifVoter significantly outperforms all of them in term of both sensitivity and precision. For example, on Tompa’s benchmark datasets, MotifVoter improves the sensitivity by 215% and the precision by 45.5%. More importantly, MotifVoter can locate almost all binding sites that are found by its basic motif finders. It can distinguish the real binding sites from the false positives in the aggregation of outputs from the multiple motif finders. We also show that MotifVoter works well across different species and different types of background sequences. In particular, MotifVoter gives the biggest improvement in real background sequences (see description on Tompa’s benchmark dataset in the next section) and higher organisms (*H. Sapiens* and *M. Musculus*). Finally, we show that as long as some good motif finders are included in MotifVoter, then even if there are a few motif finders with poor performance, the performance of MotifVoter is still substantially better.

In practice we might not always be able to run a lot of motif finders. Hence, we have studied the performance of MotifVoter by only including the fastest N ($N = 3, 4, 5$) motif finders. The results show that MotifVoter is stable. The results show that the performance of MotifVoter is still significantly better than the best motif finder in terms of sensitivity and precision when we run only the 5 fastest motif finders.

3.1 Performance of Individual Motif Finders with the Inclusion of Lower Rank Motifs

Tompa et al.'s study [135] assessed rank 1 motifs predict by various motif finders. However, this assessment did not address whether using motifs of lower rank will improve the overall performance of individual motif finders. This section shows that even by including motifs of lower ranks, the performance of individual motif finders cannot be improved substantially. Figure 3.2a shows the sensitivity of the predicted binding sites by the top- n motifs of each motif finder. The best individual motif finder has sensitivity 0.130 if we just consider the predicted motifs of rank 1. When we consider the sites predicted by top-30 motifs of the best individual motif finder, the sensitivity is improved to 0.175. This suggests that, even if we consider motifs of rank 2 or above, the sensitivity of individual motif finder is improved by at most 25%. Moreover, the precision decreases significantly since a lot of noise exists in the motif list of rank 2 or above (Figure 3.2b).

The black curve in Figure 3.2a shows the sensitivity of the predicted sites by all 10 motif finders. If we just consider the predicted rank 1 motifs of the

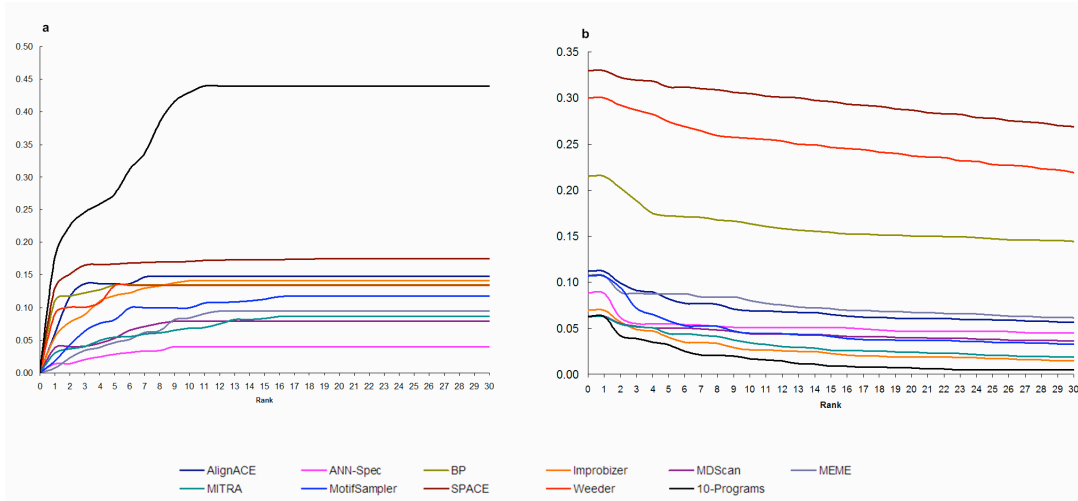


Figure 3.2: This figure shows the performance of 10 individual motif finders (color curves) and the combined result of all 10 motif finders (black curve). Figures (a) and (b) show the cumulative sensitivity (nSN) and precision ($nPPV$), respectively, of these 11 motif finders when we include more motifs with lower rank. The figure shows that the combined result of all 10 motif finders has a much higher sensitivity than any individual motif finder. However, it also reduces the precision a lot.

10 motif finders, the sensitivity is 0.177. The sensitivity is improved to 0.439 when we consider the top-30 motifs of all 10 motif finders. This suggests an improvement of 148% in sensitivity. This observation suggests that rank 2 or above binding sites predicted by all 10 motif finders are useful.

Though rank 2 or above motifs predicted by various motif finders may help to improve sensitivity, majority of them may be noise. For instance, in Tompa’s dataset, among all sites predicted by the rank 2-30 motifs of the 10 motif finders, only 0.47% of them are real binding sites. On the other hand, 6.27% of the sites predicted by the rank 1 motifs of the 10 motif finders are real binding sites (see Figure 3.2b). Hence, there is more noise in rank 2 or above motifs. This suggests that inclusion of motifs from lower rank can only be effective if we consider ensemble methods.

3.2 Different Motif Finders Discover Different Binding Sites

In general, motif finders can be divided into two major types, namely PWM model (profile based) and (l, d) model (consensus based). There is no general agreement on which model is better. Figure 3.3a gives a comparison of the binding sites predicted by the two types of motif finders. We divide the motif finders into two groups depending on the model they are based on. The first group consists of 3 motif finders based on (l, d) model, which are MITRA [40], Weeder [100], and SPACE [144]. The second group consists of 7 motif finders based on PWM model, which include AlignACE [61], ANN-Spec [147], BioProspector [79], Improbizer [5], MDScan [80], MEME [8], and MotifSampler [133]. It shows the number of sites correctly predicted by (i) both groups, (ii) (l, d) model group only, and (iii) PWM model group only. The figure showed that 45.3% (243 out of 536) of the correctly predicted sites are predicted by either (l, d) model or PWM model. This implies that (l, d) model and PWM model may be suitable for discovering motifs for different types of datasets.

Even for motif finders of the same type, the individual motif finders may be based on different heuristics and use a different set of parameters, and so may be suitable for discovering motifs from different types of datasets. For instance, consider the three motif finders SPACE, Weeder, and MITRA which are based on (l, d) model. Figure 3.3b shows the correctly predicted sites by them. We observe that, even by using the same (l, d) model, different motif finders are suitable for finding different types of motifs. And it also provides evidence that combining results from motif finders of the same model may still provide a better motif.

3.3 MotifVoter - A Method That Utilizes the Sites Predicted by Multiple Motif Finders

Predicted by Multiple Motif Finders

Combining results from multiple motif finders and considering motifs of lower rank will obviously include more binding sites, but it will also include more false positives. To develop a robust ensemble method, we need an effective way to distinguish real binding sites from noise based on the outputs from the various motif finders.

Most existing methods (e.g. SCOPE [27], BEST [36, 63] and WebMotifs [49, 114]) are based on integration at the motif level rather at the binding site level. The issue of how to distinguish a real binding site from false binding sites is not

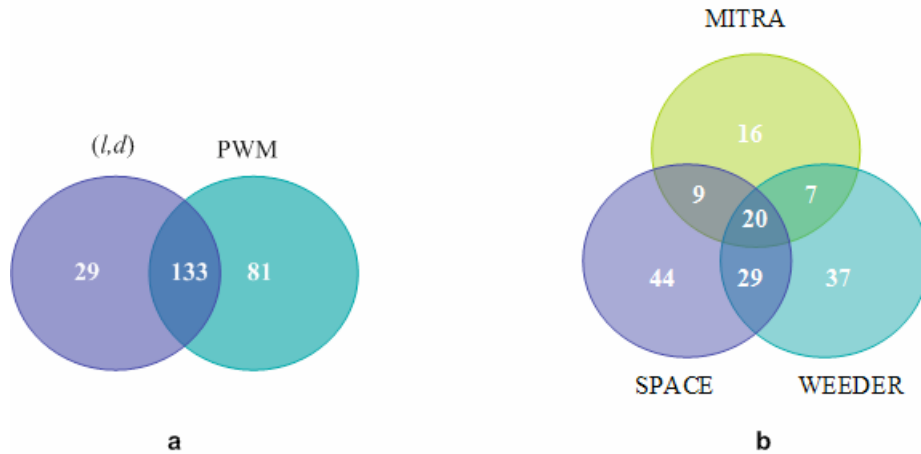


Figure 3.3: Our study has 3 motif finders based on (l, d) model and 7 motif finders based on PWM model. Using their top-30 motifs, the 10 motif finders can discover 243 binding sites in Tompa’s benchmark dataset. (a) shows the numbers of sites that can be found by (i) both groups, (ii) (l, d) model group only, and (iii) PWM model group only. (b) focuses on the three (l, d) motif finders and shows the number of sites that can be found by various combination of 3 (l, d) motif finders.

adequately addressed in the previous ensemble methods.

A naive approach is to report the binding sites that are covered by more than 2 motifs. However, our experiments show that the improvement is only limited. (For instance, though this naive approach improves sensitivity (nSN) by 68% over the current best motif finder (SPACE), this method loses in precision ($nPPV$) as much as 17.3% over SPACE in Tompa’s benchmark dataset). More importantly, it is not trivial to define whether a binding site reported by multiple finders is real or noise.

We developed a novel ensemble method MotifVoter, which integrates the results of 10 motif finders that:

1. performed reasonably well on Tompa’s benchmark and,
2. were easily obtainable from public domain.

These component motif finders are: AlignACE [61], ANN-Spec [147], BioProspector [79], Improbizer [5], MDScan [80], MEME [8], MITRA [40], MotifSampler [133], SPACE [144], and Weeder [100]. It may be noted they are also some of the widely used motif finders in the community of biologists. Appendix A describes the characteristics and parameters used in each of these motif finders. In the evaluation, we have used three datasets (Tompa’s benchmark dataset, the metazoan dataset, and the *E. coli* dataset). Below we describe the method detail of MotifVoter.

3.4 Pairwise Similarity Between Motifs

We measure the similarity of two motifs x and y based on their instances. Let $I(x)$ be the set of instances (or the regions covered by the instances) of x . Let

$I(x) \cap I(y)$ be the set of regions covered by at least one instance in x and one instance in y . Let $I(x) \cup I(y)$ be the set of regions covered by any instance of x or y . We denote the total number of nucleotides of all the regions in $I(x) \cap I(y)$ and $I(x) \cup I(y)$, by $|I(x) \cap I(y)|$ and $|I(x) \cup I(y)|$ respectively. The similarity of x and y , denoted $sim(x, y)$, is expressed as $|I(x) \cap I(y)| / |I(x) \cup I(y)|$. Note that $0 \leq sim(x, y) \leq 1$ and $sim(x, x) = 1$.

Consider m basic motif finders, each reporting n motifs. Each motif corresponds to its list of predicted binding sites. MotifVoter aims to integrate the information and to give an accurate prediction of the binding sites. The main assumption behind the method is that the true binding sites have a higher chance to be predicted by more than one motif finders.

There are three stages in MotifVoter: (1) Motif filtering: this stage filters away the spurious motifs from all the candidate motifs predicted by the m motif finders (see Figure 3.1a). (2) Instance refinement: based on the candidate motifs retained in Stage 1, we identify a set of instances with high confidence that they are real binding sites (see Figure 3.1b). (3) PWM generation: from the instances computed in Stage 2, we generate the PWM of the motif (see Figure 3.1c).

3.5 Motif Filtering

MotifVoter uses a variance-based statistical measure [16, 120] to identify cluster of highly similar motifs based on similarity function as described above. Given m motif finders and each motif finder reports its top- n candidate motifs, there will be a set P of mn candidate motifs. Among all the candidate motifs in P , some of them will approximate the real motif while the other will not. We would like to identify the subset X of P such that the candidate motifs in X are likely to

approximate the real motif. Our basic idea is that if the candidate motifs in X can model the real motif, they should have high similarity. Below, we define a score function which allows us to identify X .

Let X be some subset of candidate motifs of P . The mean similarity among the candidate motifs in X , denoted as $sim(X)$, is defined as:

$$sim(X) = \frac{\sum_{x,y \in X} sim(x,y)}{|X|^2} \quad (3.1)$$

The w score of X , denoted by $w(X)$, is defined as:

$$w(X) = \frac{sim(X)}{\sqrt{\sum_{x,y \in X} ((sim(x,y) - sim(X))^2)}} \quad (3.2)$$

Note that $w(X)$ measures the similarity among the candidate motifs in X . If many of the candidate motifs in X approximate the significant motif, we should expect to have a high $w(X)$. On the other hand, we expect the complement of X , that is $P - X$, should have a low $w(P - X)$. Thus $w(P - X)$ constitute a discriminative attributes in the clustering procedure. In other word, if X is the set of candidate motifs which approximate the significant motif, we expect to have a high $A(X)$ score, where:

$$A(X) = \frac{w(X)}{w(P - X)} \quad (3.3)$$

In addition, we also assume that most of the motif finders are effective. In other word, for each motif finder, if we select its top n candidate motifs for some n , we expect at least one of these top n candidate motifs approximates the real motif. Based on this assumption, we have an additional criterion that X must

contain candidate motifs predicted by at least t motif finders for some pre-defined threshold t . In our experiments, we set $n = 30$ and $t = m$.

In summary, this stage aims to find $X \subseteq P$ which (1) maximizes $A(X)$ and (2) X contains the candidate motifs predicted by at least t motif finders. The naive method to identify X is to consider all possible X as a subset of P that satisfies the above two criterion. However, this approach is computationally infeasible. In the next section we describe our proposed heuristics to identify X to overcome this difficulty.

3.6 Heuristics Used in MotifVoter

Here we describe the heuristic method used in the MotifVoter algorithm to identify X , the set of similar candidate motifs in the first stage. Let P be all the motifs found by m motif finders, where each motif finders return n motifs.

Let P be all motifs found by m motif finders, where each motif finder returns n motifs. Steps 1-3 compute the pairwise similarity scores for all pairs of motifs. Based on these similarity scores, we apply the following heuristics approach to find X (Steps 4-9).

The time complexity of heuristics can be shown to be $O(m^2n^2)$ as follows. There are $|P| = mn$ motifs. For each motif z , we consider mn different $X_{z,j}$ subsets. For each subset, we need to compute the value of $A(X_{z,j})$ for all j . Note that we can obtain the value of $A(X_{z,j})$ from the value of $A(X_{z,j-1})$ in constant time. So, for each motif z , to compute all values of $A(X_{z,j})$, it takes $O(mn)$ time. Therefore, the overall time complexity is $O(m^2n^2)$.

Algorithm 2 MotifVoter**Require:** $P; |P| = k$ **Ensure:** PWM of the aligned sites

```

1: for each  $z, y \in P$  do
2:   compute  $sim(z, y)$ 
3: end for
4: for each  $z \in P$  do
5:   sort  $p_1, p_2, \dots, p_k$  such that  $sim(z, p_i) \geq sim(z, p_{i+1}); \forall i = 1..k$ 
6:   consider sets  $X_{z,j} = \{z, p_1, \dots, p_j\}; \forall j = 1, \dots, k$ 
7:   compute  $A(X_{z,j})$  for all such  $X_{z,j}$ 
8:   set  $Q(X_{z,j}) =$  number of motif finders contributing to  $X_{z,j}$ 
9: end for
10: set  $X = X_{z,j}$  with the maximum  $A(X_{z,j})$  score, if there are two such  $X_{z,j}$ 's, pick the one
    with the largest  $Q(X_{z,j})$ 
11: extract and align sites of motifs in  $X$ 
12: construct PWM

```

3.7 Instance Refinement

Given X , we obtain the list of instances using two criteria. First, we accept all regions which are covered by instances of at least two motifs x and y in X where x and y are predicted by two different motif finders. The reason behind is that it is unlikely that several motif finders predict the same spurious binding sites.

Second, we accept all the instances of the motif in X that have the highest confidence to approximate the real motif the best. To rank the candidate motifs x in X , we use a confidence score defined as follows. Let $B(x)$ be the total number of nucleotides covered by the instance of x . Let $O(x)$ be the total number of nucleotides covered by the instances of x and also the instances of the motif y where y is a motif in X predicted by some other motif finder. The confidence score of x is defined as $O(x)/B(x)$.

For the selected instances that are covered by more than one motif finder, we further apply a post-processing procedure to refine each instance by removing the nucleotides that are only covered by a single finder to increase the precision of our prediction as these nucleotides are likely to be noise.

There are two cases in this post-processing procedure for instances in X .

1. If there exist only two instances overlap with each other, we perform intersections for these two instances. Example below shows that the final instances given by MotifVoter is the intersection of instances given by MEME and AlignACE:

```
2, -309, CAGGGTAGGGACA, MEME
2, -305, GTAGGGACAGAGC, ALIGNACE
2, -305, GTAGGGACA, MOTIFVOTER
```

2. If there are more than two instances overlap with each other. In this case we first take intersection of two adjacent instances, then perform a join/union of the previous intersection. Example below shows that we would first take intersection of instances from BP and MITRA, followed by MITRA and MEME. Union of these two intersection gives the final instances of MotifVoter.

```
0, -286, AGGAAAATTT, BP
0, -283, AAAATTTGTTTCATACAGAAGG, MITRA
0, -283, AAAATTTGTTTCA, MEME
0, -283, AAAATTTGTTTCA, MOTIFVOTER
```

Intersection of the first two adjacent instances (BP and MITRA) gives:

```
0, -283, AAAATT
```

Intersection of the second two adjacent instances (MITRA and MEME) gives:

```
0,-283,AAAATTTGTTTCA
```

Finally the union of the two above intersection gives:

```
0,-283,AAAATTTGTTTCA,MOTIFVOTER
```

Note that the join/union operation is sensitive to the order of the binding site position. It assumes that their positions are sorted ascendingly.

3.8 Position Weight Matrix (PWM) Generation

Given all the instances predicted by MotifVoter, Stage 3 generates a PWM motif to model the instances. This stage has two steps: First, a multiple sequence alignment of those instances are computed using MUSCLE [38]. Second, a PWM is generated from the alignment to model the motif. Figure 3.1c provides an illustration of Stage 3.

3.9 Experimental Results

3.9.1 The performance of MotifVoter versus individual motif finders

We compare the performance of MotifVoter with individual and ensemble motif finders on Tompa's benchmark datasets. Figure 3.4 shows the results. MotifVoter improves the sensitivity (nSN) by 215% (from 0.13 to 0.41) when compared with the best performing stand-alone motif finder while the precision ($nPPV$) is improved by 45.5%.

More importantly, MotifVoter can locate almost all binding sites that are found by any existing finders (see Figure 3.5). As MotifVoter uses 10 basic motif

finders as its components, if the basic motif finders cannot find a particular real binding site, MotifVoter cannot find it too. Thus the highest possible sensitivity that can be achieved by MotifVoter (or any ensemble method) is the fraction of real binding sites that can be found by at least one basic motif finder. Evaluation in Tompa's benchmark datasets shows that the highest possible sensitivity that can be achieved is 0.44. MotifVoter, on the other hand, can achieve a sensitivity of 0.419.

We also evaluate the performance of classical ensemble methods that uses the principle of picking one motif based on some forms of scoring function from the collection of motifs returned by the individual motif finders (e.g. WebMotifs, SCOPE, cBEST). We observe that even if they can pick the most sensitive motif per data set, their sensitivity is at most 0.282 (refer to the performance of the "Best Motif Finder" in Figure 3.4.), which is 48.6% lower than the sensitivity of MotifVoter. This implies that the principle of integrating motifs at the sites level gives significant improvement in performance.

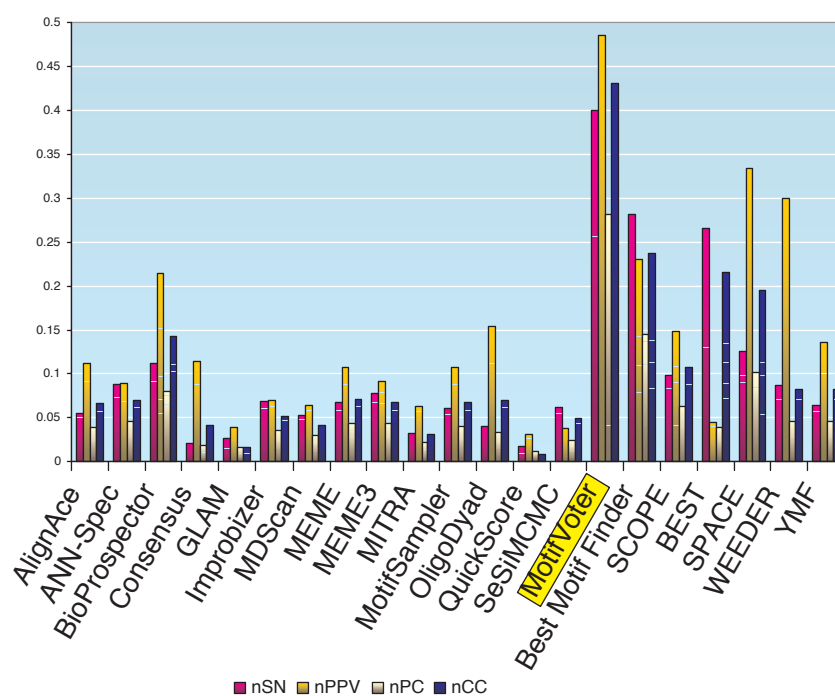


Figure 3.4: Comparison of MotifVoter with individual and ensemble motif finders on Tompa's Benchmark dataset.

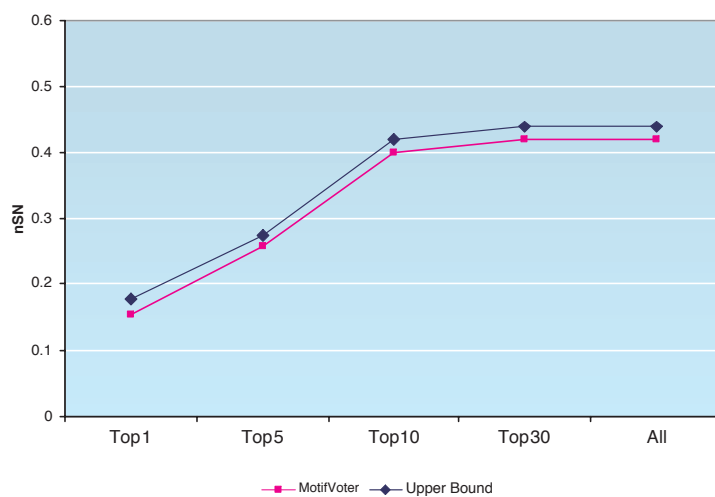


Figure 3.5: The sensitivity of MotifVoter versus the maximum possible sensitivity (using 10 selected motif finders). The blue curve shows the fraction of nucleotides that are found by at least 1 motif finder. The pink curve shows the corresponding nucleotide sensitivity of MotifVoter. Note that the x -axis refers to the top- N number of motifs we use from each basic motif finder in MotifVoter. For example, top-10 means we use the top 10 motifs from each finder. It is not the number of motifs returned by MotifVoter per se. MotifVoter only returns rank-1 result.

3.9.2 Performance of MotifVoter on Different Background Sequences and Species.

This section discusses the performance of MotifVoter on different species and background sequences. Figure 3.6 shows the performance of MotifVoter on various background sequences in Tompa’s benchmark datasets. In this evaluation, the major improvement is on real datasets (275%), followed by generic dataset (128%). Since modeling the background sequences of real type is more difficult, individual motif finders usually perform worse in real datasets when compared with markov and generic datasets. On the other hand, MotifVoter combines both PWM and (l, d) models from different motif finders, and hence it is able to recover more binding sites in real datasets.

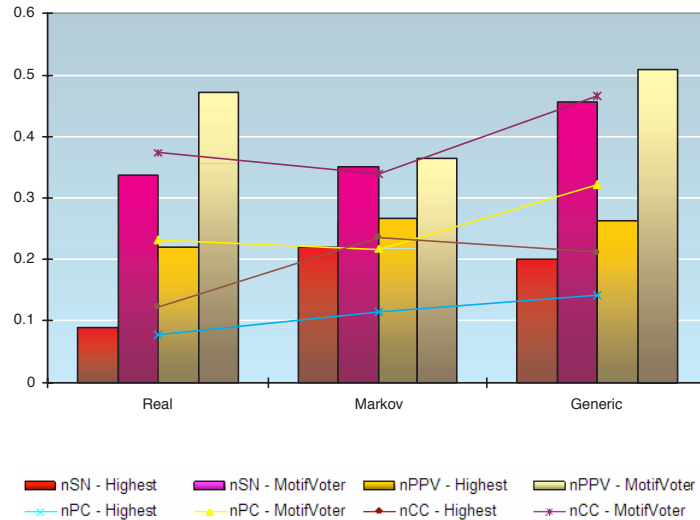


Figure 3.6: Performance of MotifVoter on various types of background sequences when compared with the best individual motif finder on Tompa’s Benchmark dataset.

We obtain consistent results in the evaluation based on species also (Figure 3.7). MotifVoter achieves the highest nSN and $nPPV$ in datasets on all four

species namely human, mouse, fruitfly and yeast. But MotifVoter made major improvement on human dataset (314%) followed by fruitfly (263%) while the least improvement is made on yeast dataset (84%). One possible explanation is that the binding sites in human, mouse, and fruitfly are much less conserved than yeast. By making use of various modeling capability of different basic motif finders, MotifVoter has a higher chance of capturing more diverse binding sites model on human, mouse, and fruitfly.

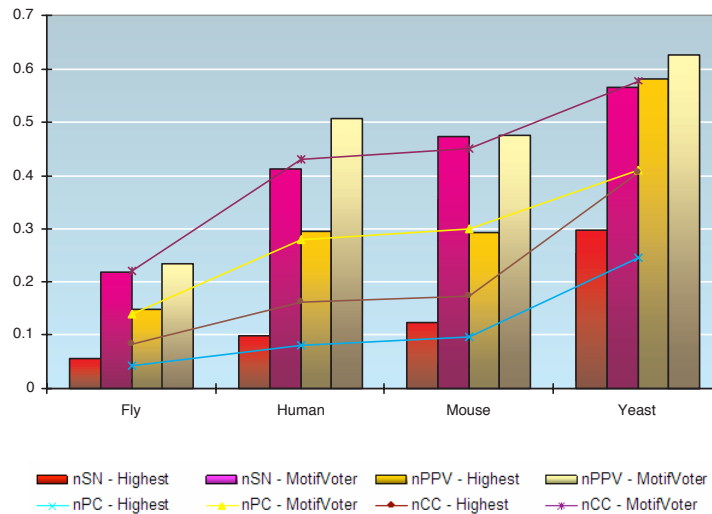


Figure 3.7: The performance of MotifVoter on various species when compared with the best individual motif finder on Tompa’s Benchmark dataset.

3.9.3 Time Complexity of MotifVoter

The time complexity is an important issue for MotifVoter. Running all 10 motif finders for MotifVoter is not always practical. We investigated whether MotifVoter can improve the sensitivity and precision compared to the best individual motif finder, if we only execute the fastest N ($N=3, 4, 5$) motif finders in MotifVoter. Note that the total running time to execute the fastest 5 motif finders is still smaller than the running time of MEME. Figure 3.8 shows the detailed

running time of 10 programs on 1.5K bp datasets based on both Tompa and Metazoan dataset. It is measured on a 3.6Ghz Xeon Linux workstation with 4 processors and 8GB RAM. The mean running time is 111.59 secs, and the standard deviation is 110.89 secs.

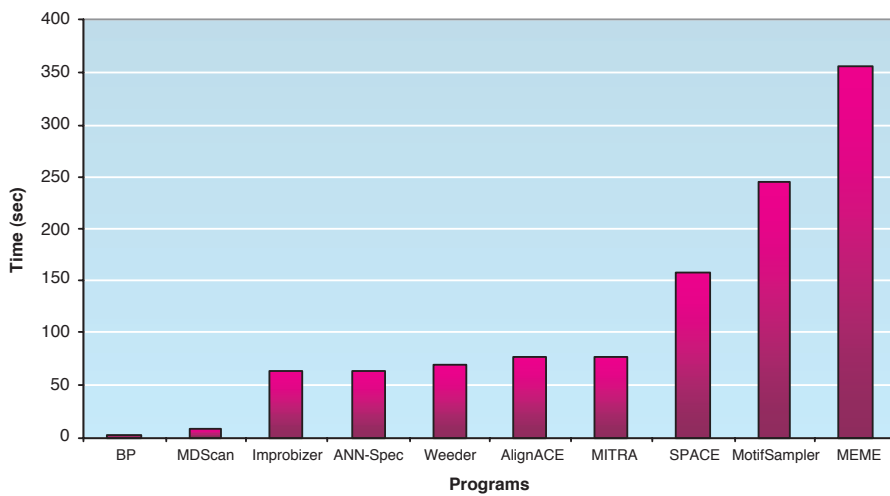


Figure 3.8: Running time of 10 motif finders on 1.5KB dataset

The actual running time of the heuristics used by MotifVoter can be seen in the Figure 3.4 below.

Figure 3.10 shows the performance of MotifVoter if we only run the fastest N finders (where $N = 3, 4, 5$). The results show that the performance of MotifVoter is still significantly better than the best motif finder in terms of sensitivity and precision.

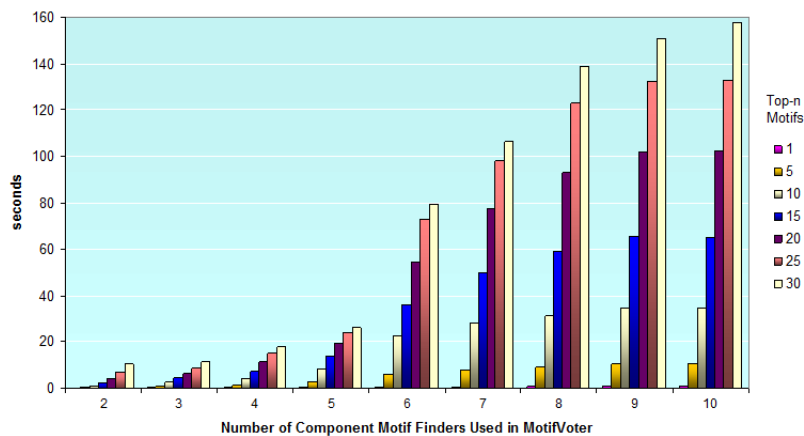


Figure 3.9: Running time of heuristic with respect to changes in m and n .

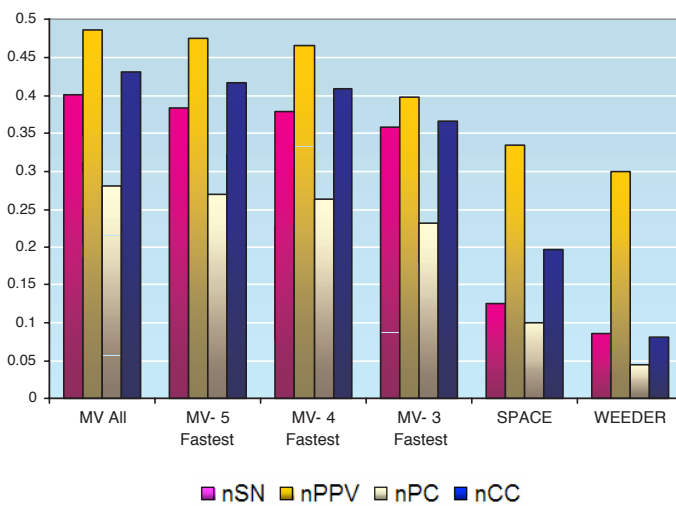


Figure 3.10: The performance of MotifVoter when we use 10 motif finders together with 1-5 random motif finders on Tompa's Benchmark dataset.

3.9.4 Robustness of MotifVoter

MotifVoter relies on individual motif finders. So, a natural question is whether the performance of MotifVoter will degrade a lot if we include some motif finders that do not perform very well. To study this aspect, we included 1-5 motif finders that predict motifs randomly (to represent motif finders with poor performance) in addition to the 10 motif finders. Each random motif finder picks a random length- l string in the input sequences as a motif. The corresponding motif instances are generated using the (l, d) motif model (that is, length- l substring with at most d mutations from the motif), where the parameter used for (l, d) are: $(8,1)$, $(10,2)$, $(10,3)$, $(15,2)$, $(15,3)$.

Figure 3.11 shows the evaluation results on this experiment. The performance of MotifVoter does degrade as more random motif finders (representing motif finders with poor performance) are included. However, even if we include 5 random motif finders (that is half of the real motif finders we used), the sensitivity (0.357) of MotifVoter is still significantly greater than that of the best individual motif finder (0.126). A similar observation is obtained for precision. In other words, MotifVoter is robust even if some of the component motif finders perform unsatisfactorily. We also observe that MotifVoter does not give any preference to any of its component motif finders (see Figure 3.12).

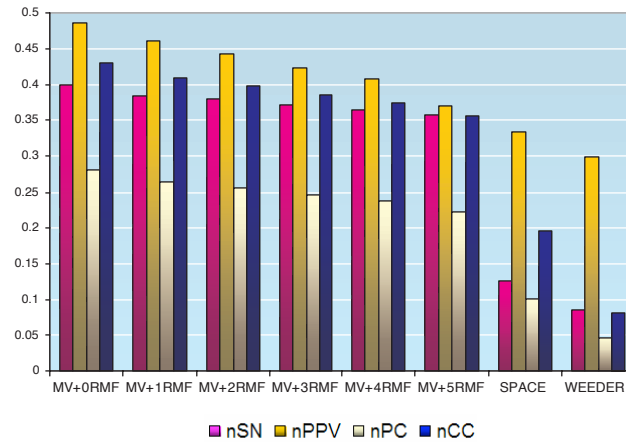


Figure 3.11: Performance of MotifVoter based on all 10 motif finders (MV), the fastest 5 motif finders (MV-5), the fastest 4 motif finders (MV-4), and the fastest 3 motif finders (MV-3). The fastest 5 motif finders we considered are BP, MDScan, Weeder, ANN-Spec, and Improbizer. (Note that the total running time of these 5 motif finders is faster than MEME.) on Tompa’s Benchmark dataset.

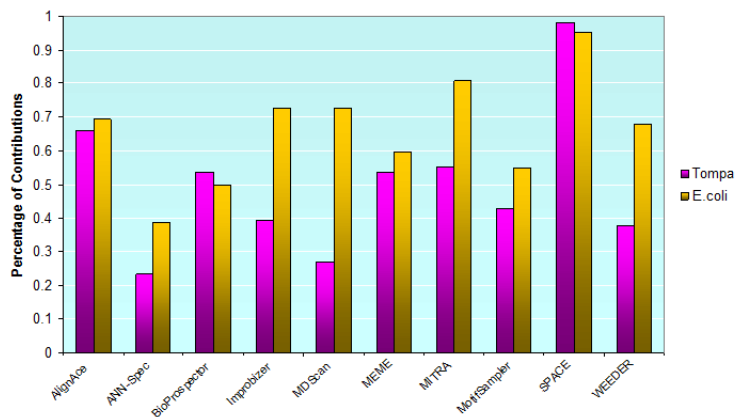


Figure 3.12: Contribution of component motif finder to the output given my MotifVoter.

3.9.5 Validation on Metazoan Datasets

We also examine the performance of MotifVoter on the metazoan datasets that have been drawn from real genomic sequences. The metazoan datasets are taken from ABS database [14] (<http://genome.imim.es/datasets/abs2005/index.html>), and consist of 68 datasets. The number of sequences ranges from 3-39 and the sequence lengths are up to 500 bp. The binding sites are gathered from the literature where they have been experimentally verified. The sites and the promoter sequences have been manually curated to ensure data consistency. They come from three different organisms: human, rat and mouse.

When we repeated the same experiments on metazoan datasets, we observed similar results. MotifVoter outperforms the best motif finder in this dataset by 103% and 35% in nSN and $nPPV$ respectively (Figure 3.14). We also validate the performance of MotifVoter on individual species of the metazoan dataset. MotifVoter also performs better in each case (Figure 3.15). The highest possible sensitivity for this dataset is 0.650, and the sensitivity of MotifVoter is 0.632 which is again close to the upper bound. Please refer to Figure 3.13 below for the detailed evaluation of MotifVoter on the upper bound analysis.

For Metazoan dataset, the maximum possible sensitivity is 0.668 while MotifVoter has a sensitivity of 0.650 by missing only a few binding sites.

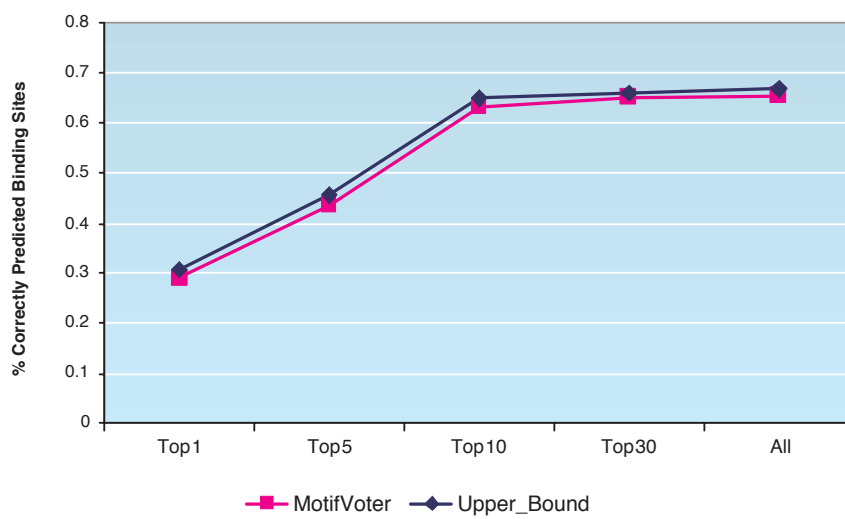


Figure 3.13: Upper bound analysis on Metazoan datasets

Figure 3.16 shows several example binding sites from metazoan datasets. It illustrates that MotifVoter finds more binding sites than stand-alone motif finders. Also, in general the predicted motif models are similar to the actual motifs.

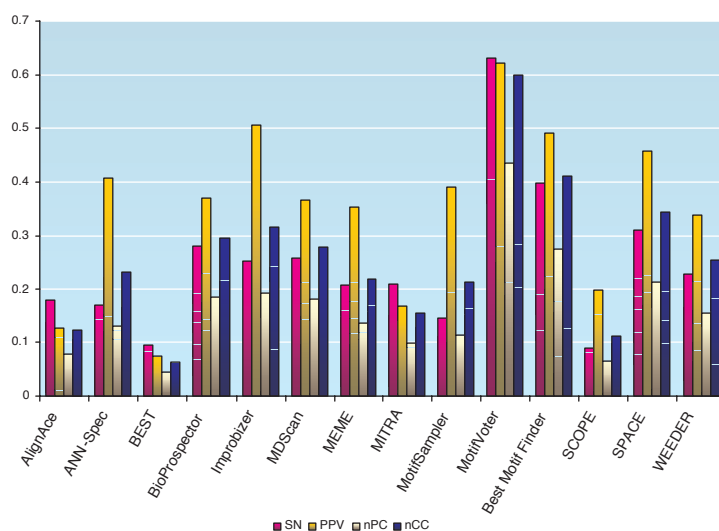


Figure 3.14: Comparison of MotifVoter and individual motif finders on metazoan dataset.

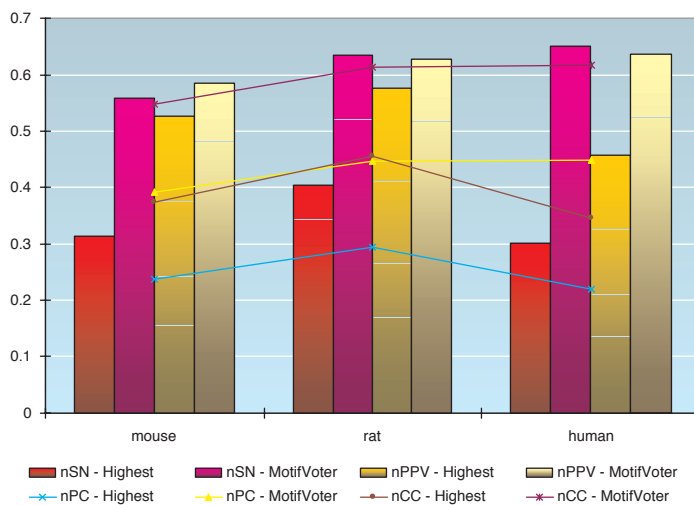


Figure 3.15: Performance of MotifVoter on various species compared to the best performing individual motif finders.

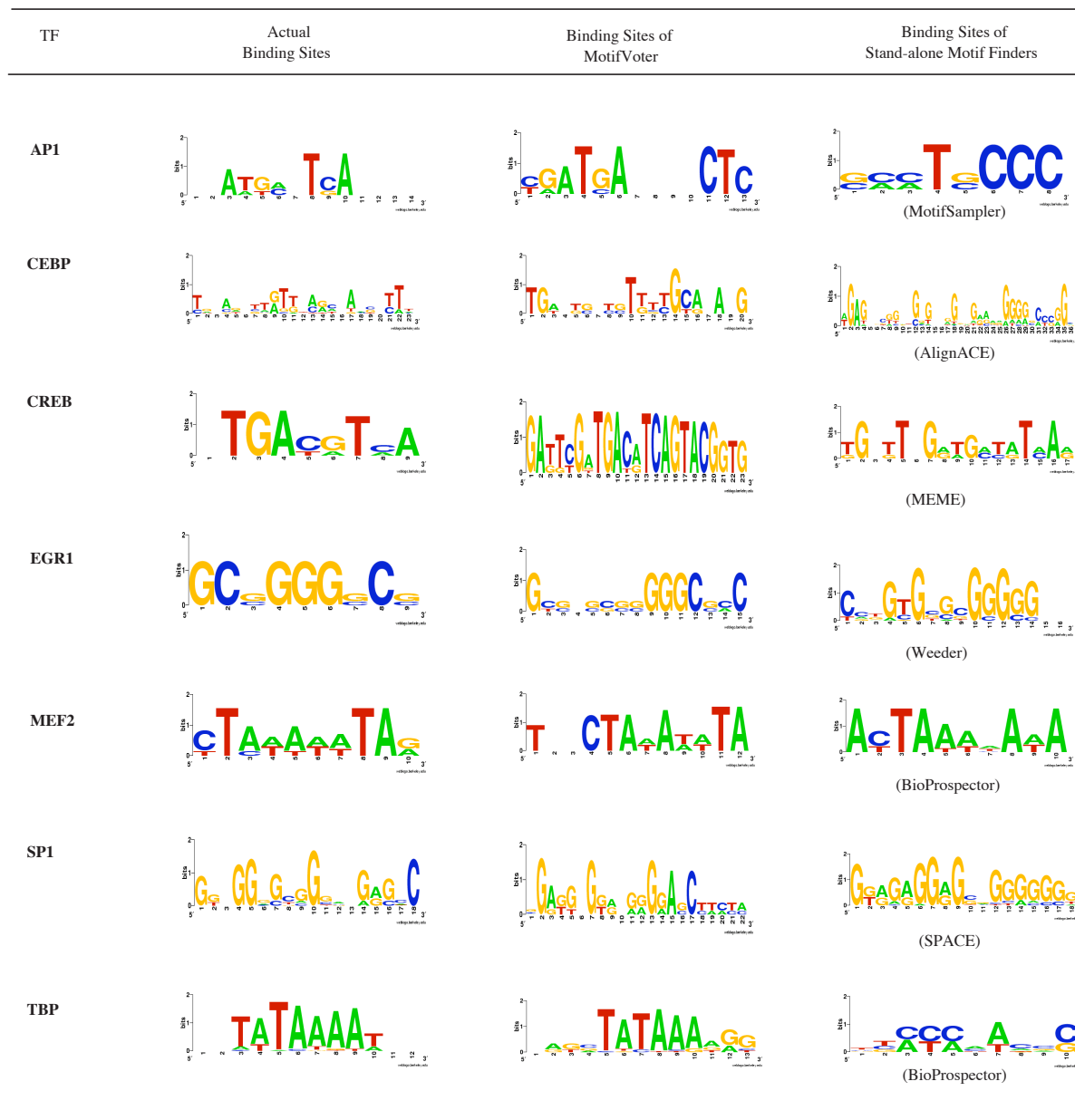


Figure 3.16: Examples of the binding sites found by MotifVoter and stand-alone motif finders on real metazoan datasets. For each of these datasets, we report the result from the best performing stand-alone motif finder.

3.9.6 Comparison of MotifVoter with Other Ensemble Methods

In the literature, there are two existing directions for performing ensembles, they are: motif-based and site-based methods. The motif-based methods includes SCOPE [27], BEST [36,63], WebMotifs [49,114] ensemble methods. Their principle is to identify the highest confident motif (under certain criteria) out of all predicted motifs. Although they can improve the accuracy of motif finding, they will fail to report a good motif when none of the reported motifs is approximates the true motif.

The second group tries to identify good sites to generate the final motif. EMD [59] belongs to this group. EMD assumes the sites reported by higher ranking motifs have higher accuracy. Based on the motif score, the predicted sites are grouped. Then, a motif is generated from each group of sites. This method will give good result if majority of the high ranking sites are similar to the true motif. However, when the good sites are distributed in motifs of different ranks, the performance of EMD will be reduced.

We compare MotifVoter with these four most recent ensemble methods SCOPE, EMD, BEST and WebMotifs. For comparison with SCOPE, EMD, and BEST we perform experiments on E.Coli datasets. For comparison with Webmotifs we use yeast ChiP dataset, since WebMotifs only take probe names as input.

E.Coli datasets are taken from RegulonDB [119] (<http://regulondb.ccg.unam.mx/>). They are generated from the intergenic regions of E.Coli genome. In total they contain 62 datasets. The average number of sequences is 12 and the average sequence length is 300bp. We are unable to perform the evaluation

on Tompa's benchmark and the metazoan datasets since EMD is not available for public use. Hence we make the comparison using E.Coli datasets alone, the results for which are obtained from EMD's publication.

SCOPE is a motif finder which integrates the motifs predicted by BEAM, PRISM and SPACER while EMD is a motif finder which uses the motifs predicted by AlignACE, BioProsPector, and MDScan. To make a fair comparison, we run a version of the MotifVoter that uses the same three motif finders used by EMD¹. Figure 3.17 shows the evaluation results.

In this dataset, SCOPE is better than EMD in terms of $nPPV$ but has a slightly lower nSN . We believe that this is because SCOPE only reports instances from 1 motif, unlike EMD which also considers instances from other motifs of the same rank. Nevertheless, even with 3 motif finders, MotifVoter can improve the nSN to 0.448 and $nPPV$ to 0.509. For further analysis of SCOPE and EMD, please refer to the discussion section. In Figure 3.17, we also include the performance of the best two individual motif finders (SPACE and Weeder) for reference.

Please refer to Figure 3.16 for the detailed evaluation of MotifVoter and other stand-alone motif finders on E.Coli dataset.

¹We cannot create a MotifVoter which uses BEAM, PRISM, and SPACER since these three stand-alone motif finders are not available. Also, note that the motif finder SPACER used by SCOPE is different from SPACE used by MotifVoter.

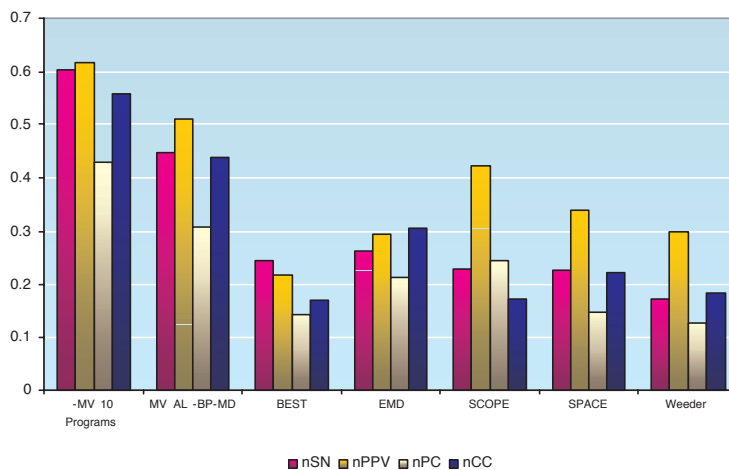


Figure 3.17: Comparison of MotifVoter with SCOPE and EMD. MotifVoter performs consistently better in both nSN and $nPPV$. We also include the performance of the best two individual motif finders (SPACE and Weeder) for reference. It shows that both SCOPE and EMD improve the performance. However, the improvement is not as significant as MotifVoter. In particular, SCOPE is better than SPACE in terms of $nPPV$ only. EMD, on the other hand, can only improve the nSN of SPACE and Weeder marginally.

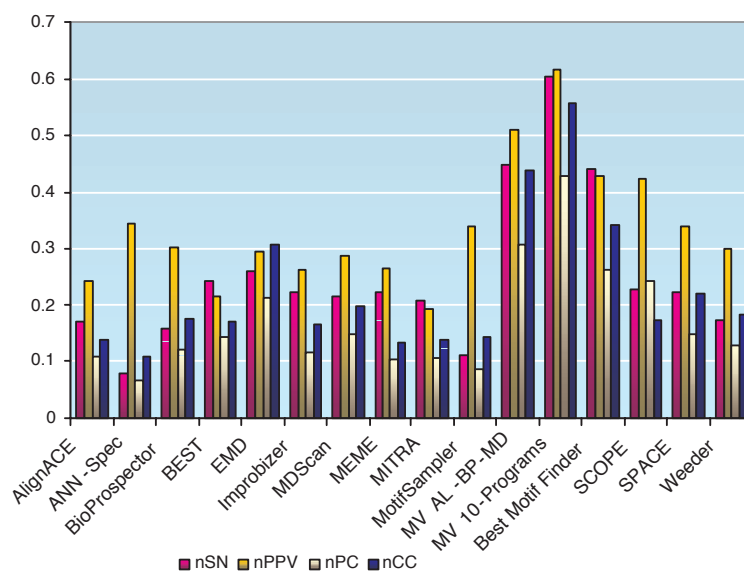


Figure 3.18: Evaluation of MotifVoter with other stand-alone motif finders in *E. Coli* dataset.

Table 3.1 lists the average sensitivity (nSN) and precision (nPPV) of each motif finder for Tompa’s Benchmark and *E. coli* datasets. The nSN and nPPV for stand-alone motif finders in Tompa’s benchmark are taken directly from [135]. For *E. coli* dataset, we are unable to obtain the result of a few stand-alone motif finders (marked with dash), because either these programs are not available or the output does not give binding sites for us to evaluate the results (e.g YMF), during the preparation of this manuscript. Motif finders marked with asterisks (*) are ensemble methods. Note that for some ensemble methods, we are not able to execute on the given datasets, we only estimate their upper bound by selecting the most sensitive and precise motif for each dataset.

	Tompa		<i>E. coli</i>	
	nSN	nPPV	nSN	nPPV
QuickScore	0.017	0.030	-	-
Consensus	0.021	0.113	-	-
GLAM	0.026	0.038	-	-
MITRA	0.031	0.062	0.206	0.193
OligoDyad	0.040	0.154	-	-
MDScan	0.053	0.063	0.217	0.289
AlignAce	0.055	0.112	0.171	0.243
MotifSampler	0.060	0.107	0.110	0.339
SeSiMCMC	0.061	0.037	-	-
YMF	0.064	0.137	-	-
MEME	0.067	0.107	0.224	0.267
Improbizer	0.069	0.070	0.225	0.264
MEME3	0.078	0.091	-	-
WEEDER	0.086	0.300	0.173	0.3
ANN-Spec	0.087	0.088	0.079	0.346
BioProspector	0.111	0.215	0.158	0.304
SPACE	0.126	0.334	0.225	0.341
SCOPE*	0.098	0.149	0.229	0.423
BEST*	0.266	0.044	0.244	0.217
RGSMiner*	≤ 0.282	≤ 0.347	≤ 0.441	≤ 0.446
WebMotifs*	≤ 0.282	≤ 0.347	≤ 0.441	≤ 0.446
EMD*	≤ 0.338	≤ 0.071	0.262	0.296
MotifVoter*	0.410	0.486	0.603	0.617

Table 3.1: Average sensitivity and precision (nPPV) of each motif finder in *E. Coli* and Tompa dataset.

We further perform comparison with BEST [36], WebMotifs [114] and SCOPE [27], the using *yeast* ChIP [54] experiments dataset. They are obtained from (<http://fraenkel.mit.edu/Harbison/>). The performance comparison can be found at Figure 3.19. MotifVoter identified 56 out of 65 motifs previously found (86.2%).

BEST uses AlignACE, BioProsPector, CONSENSUS, and MEME as its component motif finders, SCOPE uses BEAM, PRISM and SPACER, Webmotifs uses AlignACE, MDScan, MEME. Comparing with the best performing ensemble method, the improvement is 40% over YPD media, and 10% on overall datasets.

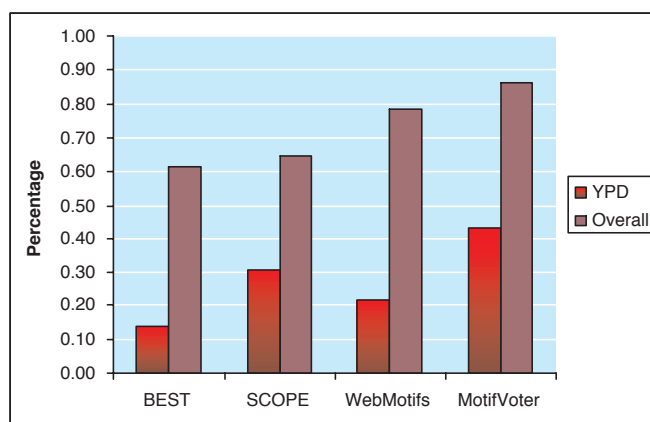


Figure 3.19: Comparison on yeast ChIP [54] experiments, with BEST, Webmotifs and SCOPE in terms of predicting percentage of correct motifs

The detail of the motif found by MotifVoter on these datasets and the comparison with the actual sites, can be found in the following Figure 3.18.

TF name	Literature	MotifVoter
ABF1		
ACE2		N/A
AFT2		
AZF1		
BAS1		
CAD1		
CBF1		
CIN5		
DAL82		
DIG1		
FHL1		












Figure 3.20: Binding sites comparison of MotifVoter on *yeast* ChIP experiments.

TF name	Literature	MotifVoter
FKH1		
FKH2		
GAL4		
GAT1		
GCN4		
GLN3		
HAP1		
HAP4		
HSF1		
IME1		N/A
INO2		
INO4		

TF name	Literature	MotifVoter
LEU3		
MBP1		
MCM1		
MET4		
MSN2		
NDD1		
NRG1		
PDR1		N/A
PHD1		
PHO2		N/A
PHO4		
RAP1		

TF name	Literature	MotifVoter
RCS1		
RDS1		
REB1		
RFX1		
RLR1		N/A
RPN4		
SFP1		
SIG1		N/A
SIP4		
SKN7		
SNT2		
SOK2		

TF name	Literature	MotifVoter
SPT2		N/A
SPT23		N/A
STB1		
STB4		
STB5		
STE12		
SUM1		
SUT1		
SW14		
SW16		
TEC1		
THI2		

TF name	Literature	MotifVoter
TYE7		
UME6		
YAP1		
YAP7		
YDR026c		
ZAP1		N/A

and 250bp downstream sequences of the annotated TSS were repeat and exon masked.

E2F: We retrieved 700bp upstream and 200bp downstream sequence from the annotated genes start site were repeat and exon masked. The sample sequences correspond to promoter region of genes in their Table 3 of [112] and the background sequences correspond to the promoter region of genes in their supplemental data: (<http://www.genesdev.org/cgi/content/full/16/2/245/DC1>).

MYOD/MYOG: The 750bp upstream and 250bp downstream sequences of the annotated transcription start site (TSS) [23] are extracted from genes targeted by either MYOD or MYOG in both MDER and C2C12 cells found in this url: (<http://www.nature.com/emboj/journal/v25/n3/extref>).

HNF4/HNF6: the dataset are downloaded from (http://jura.wi.mit.edu/young_public/autoregulation/downloaddata.html). The 750bp upstream and 250bp downstream sequences of the annotated TSS [96] were repeat and exon masked.

SOX: First the exact loci where the transcription factor has been bound were extracted from MacIsaac et.al [84]. Secondly, entire promoter regions (10kb) that include the bound loci are used using the data from [17].

NOTCH: The target genes were taken from the supporting Table 1 of [99] and the 3 kb upstream of human sequences were retrieved from EnsEMBL 41 (repeat masked sequences).

3.10 Effect of Discriminative and Constraint Attributes

In this section we assess the importance of discriminative measure and constraint that the motif in the cluster should be contributed by at least t motif finders. We perform experiments on *E. Coli* dataset and drop each of these two attributes at a time. MotifVoter uses same component motif finders as EMD, namely AlignACE, BioProspector, and MDScan.

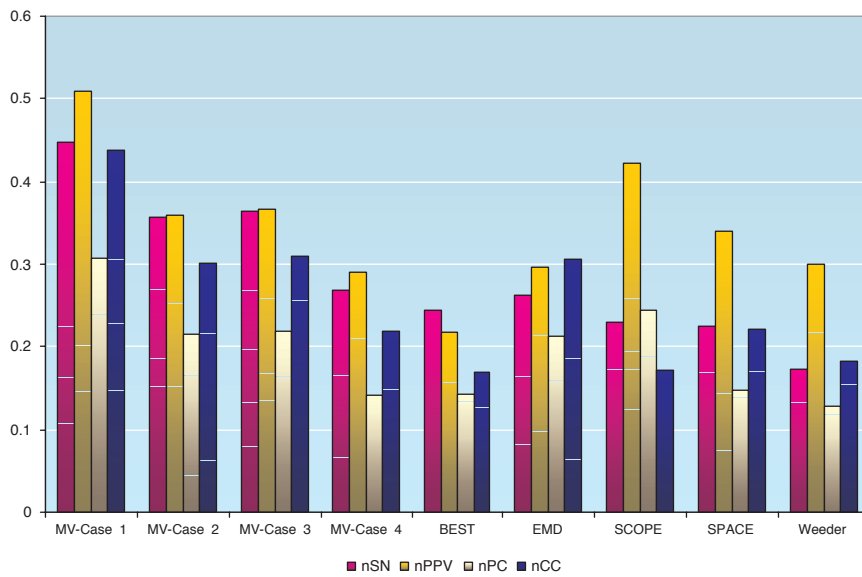


Figure 3.22: Importance of discriminative measure and constraint.

Hence, overall we evaluate four possible scheme:

1. MotifVoter contain both discriminative and constraint attributes.
2. MotifVoter contain only discriminative attributes.
3. MotifVoter contain only constraint attributes.

4. MotifVoter does not contain both discriminative and constraint attributes.

We observe that both of these two attributes are equally important in their own right. As can be seen in Case 2 and Case 3, by dropping any of them reduce the precision of MotifVoter. And without both of them the performance of MotifVoter is reduced greatly in terms of sensitivity and precision.

3.11 Observations on the Binding Sites Missed by MotifVoter

In Tompa's benchmark dataset, out of 56 datasets there are 22 datasets in which less than 50% binding sites can be found. Having analyzed those 22 datasets, we suspect that most of these binding sites are highly unconserved. Precisely, out of 22 datasets, 15 datasets are unconserved (70%). For the remaining 7 datasets out of 22 datasets (30%), the density of binding sites (that is the ratio of total length of binding sites over the total size of dataset) is relatively low. Under the low signal to noise ratio, it is harder to discover the binding sites.

Below we show three examples of the actual binding sites in Tompa's benchmark dataset. Observe that in each of these datasets, we only find a short conserved region in the binding sites. The majority of the binding sites are highly unconserved. Since none of the current motif finders can capture such motif model, without any extra information, MotifVoter is bound to miss the binding sites.

The highlighted regions are the ones found by MotifVoter. For example in *hm03r* dataset, out of 15 true binding sites MotifVoter could identify all the conserved parts in 6 binding sites. The remaining 60% of the binding sites missed

by MotifVoter are highly unconserved.

```

hm03r, percentage of missed binding sites 0.60
CCATTTCTTTATG-----ATTGATAGTCTGAG----
ACTGAAAAGCTTAGGAAATGGTA-----TTGAGAAATCTGGGC--
A-----A-----TTACGAAAT----GGA--
T-----CTCCTGCAGTAAGGTAGGT-
-----TTG-GAAGTCAATATTTG
-----TTGGAAAAGTCAAGGTTTG
TATTGCAGTG-----ATGTAATCAGC-----
GAC-----CTTTGCAATCCTGG-----
CACACTTGGAAAT----AGCAATAG-----ATGCAATTTGGGACTTA
-----CCTTTTATC-----TGTTTTGACAGTCTGGG-----
----AAGTGTG-----AAGCAAGA-----
----CGGGTGTATTCAAGCAAAAAAATAAATAAATACCTATGCAATAC-----
----GGATGTTACACAAGCAAACA----AAATAAATATCTGTGCAATAT-----
----TGGGTGTTATATGAGCAAACA----AAATAAATACCTGTGCAACAT-----
-----GGGCGATTGGGCAACCCGG-----C

hm05r, percentage of missed binding sites 0.64
-----T-----AAC-----
-----T-----AAC-----
-----ATT-----GAA-----
-----TGGTGAGTGGAGAAGG-----
-----AGCCAAGCTGTCAACTTCCAGTT
--ACCG-----GCAGTTAGGATACTCCTAAG-----
--CA--AAAAAGGGCGTGAACTTGG-----A-----
--CG--GAAAAAGCG-----TTTCG-----C-----
GGGGCG-----GGGCGCGCGGCAGGGTCGTTACGAAG-----
G-AGCGATATAAACGGGCGC-----
----CTTTCCAAC TGCCCGCTAATTCCG-----

mus12m, percentage of missed binding sites 0.57
GGAAAA--CAAAGG-----TAATG
AAAGAAATTCAGAGAGTCAT-----CAGAA
TGAAATATGTG-----TAATA
GCACTGGAAACCCTGAGTTTC-----AGGAC
-----CTCATTTCCTTGGTTTCAGCAACTTAACT
-----TT-ATTTT-----C-----CA--
-----ATTTTC-----CAATGTAA--

```

We also observe that the density of binding sites (ratio of total binding sites length over the total size of dataset) does increase the difficulty of the dataset. In Tables 3.11 and 3.11, we show some examples in which we compare the performance of MotifVoter on the basis of the binding site density.

Dataset	Total Binding Sites Length (a)	Total Sequence Length (b)	Ratio (a/b)	%Missed Sites
dm02r	36	2000	0.024	0.80
dm03m	105	6000	0.017	0.78
dm06r	36	3000	0.012	0.86
hm01g	157	36000	0.004	0.56
hm06g	81	4500	0.018	0.56
hm10m	48	3000	0.016	0.55
yst03m	72	4000	0.018	0.57

Table 3.2: Low binding sites density will have higher percentage of missed binding sites

Dataset	Total Binding Sites Length (a)	Total Sequence Length (b)	Ratio (a/b)	%Missed Sites
hm23r	143	2000	0.071	0.20
mus09r	41	1000	0.041	0
yst05r	74	1500	0.049	0.25
yst06g	160	3500	0.050	0.29

Table 3.3: Higher binding sites density will have lower percentage of missed binding sites

3.12 Conclusion

This chapter argues that all current motif models can only approximate the correct motif. To maximize the sensitivity, we should integrate the outputs discovered by multiple motif finders. We proposed Motifvoter, which can effectively retain almost all the correct binding sites discovered by the given individual motif finders while removing significant amount of false binding sites. It also works well across different species and different types of background sequences. We hope Mo-

tifVoter can offer a practical alternative for biologist to study novel transcription factors.

Despite of its effectiveness, our ensemble method MotifVoter is still unable to fully model the true binding sites. Since the underlying biology of regulatory mechanism is very incompletely misunderstood, exploitation of additional information such as microarray data [15] or phylogenetic footprinting [128] may help us to recover more binding sites which cannot be found with *de novo* method.

In order to improve the performance of MotifVoter, we plan to implement it as a parallel system. We also plan to extend MotifVoter for finding protein motifs. And also currently the MotifVoter assumes that all its component motif finder are equally good, we plan to develop a weighted version on this aspect.

Conclusion and Future Directions

4.1 Conclusion

Discovery of transcription factor binding sites plays an important role in gene regulation. There is a need from biologist to have a method that can help them in identifying novel transcription factors as automatic as possible. There are challenges given by the nature of real biological data and also from current practice from biologist.

In real biological data many motifs are known to be composite patterns which are groups of monad patterns (short contiguous patterns with some mismatches) that occur relatively near each other with one or more gaps [11].

For example, the binding site for *ArcA-P*, a transcription factor for regulating gene related to the respiratory metabolism in *E.coli* contains two conserved segments, separated by a gaps of length approximately 6 [88]. Another example is *Mcm1* [68, 132] or often called as the early cell cycle box (ECB) which has 3 segments and two gaps.

In the current practice many motif finding tools have been developed. Little knowledge is known on which motif finder should be used for a particular dataset. Individually, these motif finders perform unimpressively overall based on Tompa's benchmark datasets [135]. Moreover, these motif finders vary in their definitions of what constitute a motif, and in their methods for finding statistically overrepresented motifs. This makes different motif finders perform well for identifying binding sites of certain types of datasets only. There is no clear ways for biologists to choose the motif finder that is most suitable for their task. Hence, we can see that there is an immediate need for a more effective and efficient methods that allows the biologist to make use these diverse motif finders for finding novel transcription factors accurately.

In my thesis we have presented three contributions in the area of *de novo* identification of regulatory sites in response to the challenge above, they include:

1. We have addressed the problem of motif finding for generic spaced motifs by proposing a new method called SPACE. The key idea is to obtain the motif as an integration of the submotifs as defined by the frequent pattern. Submotif model provide a better modeling power compare to monad model. First, since the union of overlapping set of submotifs can represent an arbitrary length segment, submotif model can find the longest motif which fit the dataset. Hence it yield better sensitivity compare to fixed-length monad model. Additionally, it also gives better *specificity*. It can fit conserved region better when compare to monads model. Because spaced motif uses multiple segments to model the conserved functional part and spacer to model the non-functional parts.
2. For evaluating the biological significance of generic spaced motifs, we have

proposed a method to overcome the difficulty in handling biased samples by incorporating background sequence from various *species*. It is based on the idea that a motif is significant if: 1) the total number of its occurrences in all input sequences is a lot more than expected with respect to background sequence, and 2) the pattern is either very conserved or occurs in quite a number of the input sequences. Based on experiments on real biological datasets and Tompa's benchmark datasets, we show that SPACE outperforms the existing tools for spaced motifs in both sensitivity by 20.3% and specificity by 76%. And for monads, it performs as good as other tools.

3. We have proposed a novel ensemble method to identify regulatory motifs by integrating the results found by motif finders of different models. It applies a variance based statistical measure to remove the spurious motifs and then refines the prediction by filtering the noisy binding sites from using a novel voting scheme. Validation of our method on the 186 datasets (Tompa's benchmark), *metazoan*, and *E.Coli*) shows that we can improve the sensitivity to 0.487 and precision to 0.542. This is 120% improvement in sensitivity and 77% in precision over stand alone motif finders.

We conclude that our integrative approach towards motif finding offers a practical alternative for biologists to study novel regulatory sites.

4.2 Future Directions

In this section, we discuss the possible improvements in the performances and usability of the methods in the previous chapters.

Firstly, for our SPACE algorithm, we are enhancing our algorithm to handle motifs for which the same gap may have different lengths across different

instances. Our idea is to allow tolerances in the gap lengths across different instances during the mining process. Other directions include applying our approach for motif-finding on *fruitfly* regulatory regions where the sites maybe overlapping and with fluctuating positions [86], discovery of motif modules (cooperating binding factors) [51].

Secondly, in order to improve the speed performance of MotifVoter, we plan to implement it as a parallel system. Furthermore, we plan to extend MotifVoter into larger framework by allowing joint learning with multiple types of genomic data. Especially since the underlying biology of regulatory mechanism is very incompletely misunderstood, exploitation of additional information such as microarray data [15] or phylogenetic footprinting [128] may help us to recover more binding sites which cannot be found with *de novo* method. From such joint exploration of systems we hope to obtain a comprehensive knowledge on the functioning of life.

References

- [1] ADAMS, M. *et al.* The genome sequence of *Drosophila melanogaster*. *Science* 287(5461) (2000), 2185–2195.
- [2] AGGARWAL, G., WORTHEY, E. A., MCDONAGH, P., AND MYLER, P. J. Importing statistical measures into artemis enhances gene identification in the leishmania genome project. *BMC Bioinformatics*, 4 (2003), 23.
- [3] ALBRECHT, M., TOSATTO, S. C., LENGAUER, T., AND VALLE, G. Simple consensus procedures are effective and sufficient in secondary structure predictions. *Protein Eng.* 16 (2003), 459–462.
- [4] ANDERSSON, S., AND LAGERGREN, J. Motif Yggdrasil: Sampling from a tree mixture model. In *Proc. of the 10th Annual International Conf. on Research in Computational Molecular Biology (RECOMB)* (2006).
- [5] AO, W., GAUDET, J., KENT, W., MUTTUMU, S., AND MANGO, S. Environmentally induced foregut remodeling by PHA-4/FoxA and DAF-12/NHR. *Science* 305 (2004), 1743–1746.
- [6] AOKI-KINOSHITA, K. *et.al.* ProfilePSTMM: capturing tree-structure motifs in carbohydrate sugar chains. In *Proc. of the 14th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2006).
- [7] APOSTOLICO, A., COMIN, M., AND PARIDA, L. Conservative extraction of over-represented extensible motifs. In *Proc. of the 13th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2005).
- [8] BAILEY, T., AND ELKAN, C. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning* 21 (1995), 51–80.
- [9] BAILEY, T., AND ELKAN, C. The value of prior knowledge in discovering motifs with MEME. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology* (Menlo Park CA, 1995), AAAI Press, pp. 21–29.

-
- [10] BECKER, B. *et al.* A nonameric core sequence is required upstream of the *lys* genes of *saccharomyces cerevisiae* for *lys14p*-mediated activation and apparent repression by lysine. *Molecular Microbiology* 29 (1998), 151–63.
- [11] BERGER, M. *et al.* Compact, universal DNA microarrays to comprehensively determine transcription factor binding sites specificities. *Nature Biotechnology* 24, 11 (2006), 1429–1435.
- [12] BERNARDI, G. Compositional constraints and genome evolution. *Journal of Molecular Evolution* 24, 1-2 (1986), 1–11.
- [13] BLANCHETTE, M., AND TOMPA, M. Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Research* 12 (2002), 739–748.
- [14] BLANCO, E. *et al.* ABS: a database of annotated regulatory binding sites from orthologous promoters. *Nucleic Acid Research* 34 (2006), D63–D67.
- [15] BOCKHORST, J. *et al.* Predicting bacterial transcription units using sequence and expression data. *Bioinformatics* 19 (2003), S34–S43.
- [16] BORGONOVO, E. Measuring uncertainty importance: investigation and comparison of alternative approaches. *Risk Analysis* 26 (2006), 1349–1361.
- [17] BOYER, L. *et al.* Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell* 122 (2005), 947–956.
- [18] BRĀZMA, A., JONASSEN, I., UKKONEN, E., AND VILO, J. Predicting gene regulatory elements in silico on a genomic scale. *Genome Research* 8, 11 (1998), 1202–1215.
- [19] BROWN, T. *Genomes*. Wiley-Liss, 1999.
- [20] BUCHER, P. Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. *J. Mol Biol* 212(4) (1990), 563–578.
- [21] BUHLER, J., AND TOMPA, M. Finding motifs using random projections. In *Proceedings of the Fifth Annual International Conference on Research in Computational Molecular Biology* (Montreal, Canada, April 2001), RECOMB-01, pp. 69–76.
- [22] BURDICK, D., CALIMLIM, M., FLANNICK, J., GEHRKE, J., AND YIU, T. MAFLIA: A maximal frequent itemset algorithm. vol. 17, IEEE Computer Society, pp. 1490–1504.
- [23] CAO, Y., *et al.* Global and gene-specific analyses show distinct roles for *myod* and *myog* at a common set of promoters. *EMBO Journal* 25 (2006), 502–511.

-
- [24] CARLSON, J. *et.al.* BEAM: A beam search algorithm for the identification of cis-regulatory elements in groups of genes. *J. Comp. Biol.* 13 (2006), 686–701.
- [25] CARLSON, J. *et.al.* Bounded search for de novo identification of degenerate cis-regulatory elements. *BMC Bioinformatics* 7 (2006), 254.
- [26] CARVALHO, A. *et al.*. Highly scalable algorithm for the extraction of cis-regulatory regions. In *Proceedings of the Third Asia-Pacific Bioinformatics Conference (APBC)* (2005), pp. 273–282.
- [27] CHAKRAVARTY, A. *et.al.* A parameter-free algorithm for improved *de novo* identification of transcription factor binding sites. *BMC Bioinformatics* 8 (2007), 29.
- [28] CHAKRAVARTY, A. *et.al.* SPACER: identification of cis-regulatory elements with non-contiguous critical residues. *Bioinformatics* 23, 8 (2007), 1029–1031.
- [29] CHEN, C., HUGHES, T., AND MORRIS, Q. RankMotif++: a motif-search algorithm that accounts for relative ranks of k-mers in binding transcription factors. In *Proc. of the 15th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2007).
- [30] COLLADO-VIDES, J., AND HOFESTEADT, R. *Gene Regulation and Metabolism: Postgenomic Computational Approaches*. MIT Press, 2002.
- [31] DAS, D., BANERJEE, N., AND ZHANG, M. Q. Interacting models of cooperative gene regulation. *Proc Natl Acad Sci U S A* 101, 46 (November 2004), 16234–16239.
- [32] DAVIDSON, E. *Genomic Regulatory Systems: Development and Evolution*. Academic Press, 2001.
- [33] DERMITZAKIS, E., AND CLARK, A. Evolution of transcription factor binding sites in mammalian gene regulatory regions: conservation and turnover. *Mol. Biol. and Evol.* 19 (2002), 1114–1121.
- [34] DIETTERICH, T. G. Ensemble methods in machine learning. In *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems* (London, UK, 2000), Springer-Verlag, pp. 1–15.
- [35] DONG, X., SUNG, S., SUNG, W., AND TAN, C. Constrained based method for finding motif in DNA sequences. In *BIBE* (2004), pp. 483–492.
- [36] DONGSHENG, C., JENSEN, S., AND LIU, J. BEST: Binding-site estimation suite tools. *Bioinformatics* 21 (2005), 2909–2911.
- [37] DURBIN, R., EDDY, S., KROGH, A., AND MITCHISON, G. *Biological sequence analysis*. Cambridge University Press, 1998.

-
- [38] EDGAR, R. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research* 32 (2004), 1792–1797.
- [39] EISEN, M. All motifs are not created equal: structural properties of transcription factor - DNA interaction and the inference of sequence specificity. *Genome Biology* 6 (2005), 7.
- [40] ESKIN, E., AND PEVZNER, P. Finding composite regulatory patterns in dna sequences. *Bioinformatics (Supplement 1)* 18 (2002), S354–S363.
- [41] ETWILLER, L. *et.al.* Trawler: de novo regulatory motif discovery pipeline for chromatin immunoprecipitation. *Nature Methods* 4, 7 (2007), 1–3.
- [42] FAVOROV, A., GELFAND, M., GERASIMOVA, A., MIRONOV, A., AND MAKEEV, V. Gibbs ssampler for identification of symmetrically structured, spaced dna motifs with improved estimation of the signal length and its validation on the arca binding sites, 2004.
- [43] FAVOROV, A. *et al.* A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length. *Bioinformatics* 21 (2005), 2240–5.
- [44] FISCHER, D. 3d-shotgun: A novel, cooperative, fold-recognition meta-predictor. *Proteins* 51 (2003), 434–441.
- [45] FRANCES, M., AND LITMAN, A. On covering problems of codes. *Theory of Computing Systems* 30, 2 (Mar./Apr. 1997), 113–119.
- [46] FRATKIN, E., NAUGHTON, B., BRUTLAG, B., AND BOTZOGLOU, S. Finding regulatory motifs with maximum density subgraph. In *Proc. of the 14th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2006).
- [47] FRITH, M., HANSEN, U., SPOUGE, J., AND WANG, Z. Finding functional sequence elements by multiple local alignment. *Nucleic Acid Research* 32 (2004), 189–200.
- [48] GINALSKI, K., ELOFSSON, A., FISCHER, D., AND RYCHLEWSKI, L. 3d-jury: a simple approach to improve protein structure prediction. *Bioinformatics* 19 (2003), 1015–1018.
- [49] GORDON, D. *et.al.* TAMO: a flexible, object oriented framework for analyzing transcriptional regulation using DNA-sequences motifs. *Bioinformatics* 21 (2005), 3164–3165.
- [50] GROCHOW, J., AND KELLIS, M. Network motif discovery using subgraph enumeration and symmetry-breaking. In *Proc. of the 11th Annual International Conf. on Research in Computational Molecular Biology (RECOMB)* (2007).

-
- [51] GUHATHAKURTA, D., AND STORMO, G. Identifying target sites for cooperatively binding factors. *Bioinformatics* 17 (2001), 608–621.
- [52] HAN, T.H., L. W., AND PRYWES, R. Mapping of epidermal growth factor-, serum-, and phorbol ester-responsive sequence elements in the c-jun promoter. *Mol. Cell. Biol.* 12 (1992), 4472–4477.
- [53] HANNENHALLI, S., AND WANG, L. Enhanced position weight matrices using mixture models. In *Proc. of the 13th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2005).
- [54] HARBISON, C. *et al.* Transcription regulatory code of a eukaryotic genome. *Nature* 431 (2004), 99–104.
- [55] HARTEMINK, A., GORDAN, R., AND NARLIKAR, L. Nucleosome occupancy information improves de novo motif discovery. In *Proc. of the 11th Annual International Conf. on Research in Computational Molecular Biology (RECOMB)* (2007).
- [56] HERMEKING, H. *et al.* Identification of CDK4 as a target of c-MYC. *Proc. Natl. Acad. Sci.* 97 (2000), 2229–2234.
- [57] HERTZ, G., AND STORMO, G. Identifying dna and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* 15 (1999), 563–577.
- [58] HU, J., AND KIHARA, D. Limitations and potentials of current motif discovery algorithms. *Nucleic Acids Research* 33 (2005), 4899–4913.
- [59] HU, J. *et al.* EMD: an ensemble algorithm for discovering regulatory motifs in DNA sequences. *BMC Bioinformatics* 7 (2006), 342.
- [60] HUANG, E., YANG, L., CHOWDHARY, R., KASSIM, A., AND BAJIC, V. An algorithm for *Ab Initio* DNA motif detection. In *Information Processing and Living Systems* (London, 2005), Imperial College Press, pp. 611–614.
- [61] HUGHES, J., ESTEP, P., TAVAZOIE, S., AND CHURCH, G. Computational identification of cis-regulatory elements associated with functionally coherent groups of genes in *saccharomyces cerevisiae*. *Journal of Molecular Biology* 296 (2000), 1205–1214.
- [62] HUGHEY, R., AND KROGH, A. Hidden markov models for sequence analysis: extension and analysis of basic method. *Comp. Appl. BioSci* 12, 2 (Apr. 1996), 95–108.
- [63] JENSEN, S., AND LIU, J. BioOptimizer: a Bayesian scoring function approach to motif discovery. *Bioinformatics* 20 (2006), 1557–1564.
- [64] JIAWEI, H., AND KAMBER, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

- [65] JOHNSTON, M., AND CARLSON, M. *Molecular and Cellular Biology of the Yeast Saccharomyces: Gene Expression*. CSHL Press.
- [66] JONASSEN, I., COLLINS, J., AND HIGGINS, D. Finding flexible patterns in unaligned protein sequences. *Protein Science* 4 (1995), 1587–1595.
- [67] KAPLAN, T., FRIEDMAN, F., AND MARGALIT, H. Predicting transcription factor binding sites using structural knowledge. In *Proc. of the 9th Annual International Conf. on Research in Computational Molecular Biology (RECOMB)* (2005).
- [68] KATO, M., AND *et al.* Identifying combinatorial regulation of transcription factors and binding motifs. *Genome Biology* 5 (2004), R56.
- [69] KIM, T., ABDULLAEV, Z., SMITH, A., CHING, K., LOUKINOV, D., GREEN, R., ZHANG, M., LOBANENKOV, V., AND REN, B. Analysis of the vertebrate insulator protein ctf-binding sites in the human genome. *Cell* 128, 6 (2007), 1231–45.
- [70] KUTACH, A., AND KADONAGA, J. T. The downstream promoter element DPE appears to be as widely used as TATA box in *Drosophila* core promoters. *Mol. Cell Biology* 20(13) (2000), 4754–4764.
- [71] LATCHMAN, D. *Gene Regulation: a Eukaryotic Perspective*, 4 ed. Nelson Thornes Ltd, 2002.
- [72] LAWRENCE, C., ALTSCHUL, S., BOGUSKI, M., LIU, J., NEUWALD, A., AND WOOTTON, J. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* 262 (1993), 208–214.
- [73] LAWRENCE, C. E., AND REILLY, A. A. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function and Genetics* 7 (1990), 41–51.
- [74] LEE, H., LAN, T., AND ZHANG, L. Structural environment dictates the biological significance of heme-responsive motifs and the role of hsp90 in the activation of the heme activator protein hap1. *Molecular and Cellular Biology* 23(16) (2003), 5857–5866.
- [75] LENHARD, B. *et al.* Identification of conserved regulatory elements by comparative genome analysis. *Journal of Biology* 2 (2003), 13.
- [76] LEWIN, B. *Genes VII*. Oxford University Press, 2000.
- [77] LI, M., MA, B., AND WANG, L. Finding Similar Regions in Many Sequences. *Journal of Computer and System Sciences* 65, 1 (2002), 73–96. Early version appeared in STOC 99.

- [78] LIANG, S., SAMANTA, M. P., AND BIEGEL, B. A. C-Winnower: Algorithm for finding fuzzy DNA motifs. *J. Bioinformatics and Computational Biology* 2, 1 (2004), 47–60.
- [79] LIU, X., BRUTLAG, D., AND LIU, J. BioProspector: discovering DNA motifs in upstream regulatory regions of co-expressed genes. In *Proceedings of the Seventh Pacific Symposium of Biocomputing (PSB)* (2001), pp. 127–138.
- [80] LIU, X., BRUTLAG, D., AND LIU, J. An algorithm for finding protein-DNA binding sites with application to chromatin-immunoprecipitation microarray experiments. *Nature Biotechnology* 20 (2002), 835–839.
- [81] LIU, X., AND WULF, P. Probing arca-p modulon of escherichia coli by whole genome transcriptional analysis and sequence recognition profiling. *J. Biological Chemistry* 279 (2004), 12588–12597.
- [82] LUNDSTRÖM, J., RYCHLEWSKI, L., BUJNICKI, J., AND ELOFSSON, A. Pcons: a neural-network-based consensus predictor that improves fold recognition. *Protein* 10, 2354–2362 (2001).
- [83] MACISAAC, K., AND FRAENKEL, E. Practical strategies for discovering regulatory DNA sequence motifs. *PloS Computational Biology* 2, 4 (2006), 201–210.
- [84] MACISAAC, K. *et.al.* A hypothesis based approach for identifying the binding specificity of regulatory proteins from chromatin immunoprecipitation data. *Bioinformatics* 22 (2006), 423–429.
- [85] MAHONY, S. *et.al.* Improved detection of DNA motifs using a self-organized clustering of familial binding profiles. In *Proc. of the 13th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2005).
- [86] MAKEEV, V. *et al.* Distance preferences in the arrangement of binding motifs and hierarchical levels in organization of transcription regulatory. *Nucleic Acids Research* 31 (2003), 6016–6026.
- [87] MARSAN, L., AND SAGOT, M.-F. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *Journal of Comp. Biol* 7 (2000), 345–360.
- [88] MCGUIRE, A. A weight matrix for binding recognition by the redox-response regulator arca-p of *Escherichia coli*. *Molecular Microbiology* 32 (1999), 219–221.
- [89] MIDDENDORF, M., KUNDAJE, A., SHAH, M., FREUND, F., WIGGINS, C., AND LESLIE, C. Motif discovery through predictive modeling of gene regulation. In *Proc. of the 9th Annual International Conf. on Research in Computational Molecular Biology (RECOMB)* (2005).

-
- [90] MITCHELL, T. *Machine Learning*. McGraw Hill, New York, US, 1996.
- [91] MOSES, A., CHIANG, D., AND EISEN, M. Phylogenetic motif detection by expectation-maximization on evolutionary mixtures. In *Proceedings of Pacific Symposium on Biocomputing* (2004), pp. 324–335.
- [92] NARLIKAR, L., GODAN, R., OHLER, U., AND HARTEMINK, A. Informative priors based on transcription factor structural class improve de novo motif discovery. In *Proc. of the 14th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2006).
- [93] NG, P., NIRANJAN, N., JONES, N., AND KEICH, U. Apples to apples: improving the performance of motif finders and their significance analysis in the twilight zone. In *Proc. of the 14th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2006).
- [94] NIMWEGEN, E. Finding regulatory elements and regulatory motifs a general probabilistic framework. *BMC Bioinformatics* 8, Suppl 6 (2007), S4.
- [95] NISHIKAWA, K. Prediction of protein secondary structure by a new joint method. *Seikagaku*, 62 (1990), 1490–1496.
- [96] ODOM, D. *et.al.* Control of pancreas and liver gene expression by HNF transcription factors. *Science* 303 (2004), 1378–81.
- [97] OHLER, U., AND FRITH, M. Models for complex eukaryotic regulatory DNA sequences. In *Information Processing and Living Systems* (London, 2005), Imperial College Press, pp. 575–610.
- [98] OWEN, G., AND ZELENT, A. Origins and evolutionary diversification of nuclear receptor superfamily. *Cell Mol. Life. Sci.* 57 (2000), 809–827.
- [99] PALOMERO, T. *et.al.* NOTCH1 directly regulates c-MYC and activates a feed-forward-loop transcriptional network promoting leukemic cell growth. *PNAS* 103 (2006), 18261–18266.
- [100] PAVESI, G., MAURI, G., AND PESOLE, G. An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics* 17, 90001 (2001), S207–S214.
- [101] PAVESI, G., MEREGHETTI, P., MAURI, G., AND PESOLE, G. Weeder web: discovering transcription factor binding sites in a set of sequences of co-regulated genes. *Nucleic Acids Research* 32 (2004), W199–W203.
- [102] PENG, C.-H. *et al.* Identification of degenerate motifs using position restricted selection and hybrid ranking combination. *Nucleic Acids Research* 34 (2006), 6379–6391.
- [103] PEVZNER, P., AND SZE, S. H. Combinatorial approaches to finding subtle signals in DNA sequences. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology* (2000), pp. 269–278.

-
- [104] PEVZNER, P. A. *Computational Molecular Biology: An Algorithmic Approach*. MIT Press, 2000.
- [105] PHUONG, T., LEE, D., AND LEE, K. H. Regression trees for regulatory element identification. *Bioinformatics* 20, 5 (2004), 750–757.
- [106] PRAKASH, A., BLANCHETTE, M., SINHA, S., AND TOMPA, M. Motif discovery in heterogeneous sequence data. In *Pacific Symposium on Biocomputing* (2004), pp. 348–359.
- [107] PRZYTYCKA, T. An important connection between network motifs and parsimony models. In *Proc. of the 10th Annual International Conf. on Research in Computational Molecular Biology (RECOMB)* (2006).
- [108] PTASHNE, M., AND GANN, A. *Genes and Signals*. Cold Spring Harbor Laboratory Press, 2001.
- [109] RECORD, M. *et al.* *Escherichia coli* RNA polymerase σ^{70} promoters, and the kinetics of the stepstranscription initiation. *Escherichia Coli and Salmonella* 1 (1996), 792–820.
- [110] REEDER, J., AND REEDER, J. ROBERT GIEGERICH, R. Locomotif: from graphical motif description to RNA motif search. In *Proc. of the 15th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2007).
- [111] REGNIER, M., AND DENISE, A. Rare events and conditional events on random strings. *Discrete Math. Theor. Comput. Sci* 6 (2004), 191–214.
- [112] REN, B. *et.al.* CE2F integrates cell cycle progression with DNA repair, replication and G2/M checkpoints. *Genes and Development* 16 (2002), 245–256.
- [113] RIGOUTSOS, I., AND FLORATOS, A. Combinatorial pattern discovery in biological sequences. *Bioinformatics* 14 (1998), 55–67.
- [114] ROMER, K., KAYOMBYA, G.-R., AND FRAENKEL, E. WebMOTIFS: automated discovery, filtering, and scoring of DNA sequence motifs using multiple programs and bayesian approaches. *Nucleic Acids Research* 35 (2007), W217–W220.
- [115] ROTH, F., HUGHES, J., ESTEP, P., AND CHURCH, G. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole genome mRNA quantitation. *Nature Biotechnology* 16 (1998), 939–945.
- [116] ROVEN, C., AND BUSSEMAKER, H. J. REDUCE: an online tool for inferring cis-regulatory elements and transcriptional module activities from microarray data. *Nucleic Acids Research* 31 (2003), 3487–3490.
- [117] RUVKUN, G. Glimpses of a tiny RNA world. *Science* 5543 (2001), 797–9.

- [118] SAINI, H. K., AND FISCHER, D. Meta-dp: domain prediction meta-server. *Bioinformatics*, 21 (2005), 2917–2920.
- [119] SALGADO, H. *et.al.* RegulonDB (version 4.0): transcriptional regulation, operon organization and growth condition in escherechia coli k-12. *Nucleic Acids Research* 32 (2003), D303–306.
- [120] SAVAGE, L. *The Foundations of Statistics*, 2 ed. Dover Publications, 1972.
- [121] SCHJERLING, P., AND HOLMBERG, S. Comparative amino acid sequence analysis of the c6 zinc cluster family of transcriptional regulators. *Nucleic Acid Research* 24 (1996), 4599–607.
- [122] SCHRIEBER, J. *et.al.* Coordinated binding of NFkB family members in the response of human cells to lipopolysaccharide. *PNAS* 103 (2006), 5899–5904.
- [123] SIDDHARTHAN, R., VAN NIMWEGEN, E., AND SIGGIA, E. PhyloGibbs: Incorporating phylogeny and tracking-based significance assessment in a gibbs sampler. In *Proc RECOMB Satellite Workshop on Regulatory Genomics* (2004).
- [124] SINHA, S. On counting position weight matrix matches in a sequence, with application to discriminative motif finding. In *Proc. of the 14th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2006).
- [125] SINHA, S., AND TOMPA, M. A statistical method for finding transcription factor binding sites. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular (ISMB-00)* (Menlo Park, CA, Aug. 16–23 2000), R. Altman, L. Bailey, Timothy, P. Bourne, M. Gribskov, T. Lengauer, and I. N. Shindyalov, Eds., AAAI Press, pp. 344–354.
- [126] SINHA, S., AND TOMPA, M. Performance comparison of algorithms for finding transcription factor binding sites. In *Third IEEE Symposium on Bioinformatics and Bioengineering* (2003), pp. 214 – 220.
- [127] SINHA, S., AND TOMPA, M. YMF: a program for discovery of novel transcription factors and their dna binding sites. *Nucleic Acids Research* 31 (2003), 3586–3588.
- [128] SINHA, S. *et.al.* PhyME: A probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics* 4 (2004), 170.
- [129] SMITH, A., SUMAZIN, P., DAS, D., AND ZHANG, M. Mining ChIP-chip data for transcription factor and cofactor binding sites. In *Proc. of the 13th Annual International Conf. on Intelligent Systems for Molecular Biology (ISMB)* (2005).

- [130] SVETLOV, V., AND COOPER, T. Compilation and characteristics of dedicated transcription factors in *saccharomyces cerevisiae*. *Yeast* 11 (1995), 1439–84.
- [131] TANNER, M., AND WONG, W. The calculation of posterior distributions by data augmentation. with discussion and with a reply by the authors. *Journal of the American Statistical Association* 82, 398 (1987), 528–550.
- [132] TAVAZOIE, S. *et al.*. Systematic determination of genetic network architecture. *Nature Genetics* 22 (1999), 281–285.
- [133] THIJS, G. *et al.* A higher-order background model improves the detection of promoter regulatory elements by Gibbs Sampling. *Bioinformatics* 17 (2001), 1113–1122.
- [134] TOMPA, M. An exact method for finding short motifs in sequences with application to the ribosome binding site problem. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology* (Heidelberg, Germany, 1999), pp. 262–271.
- [135] TOMPA, M., LI, N., BAILEY, T., AND CHURCH, G. *et al.* Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology* 23 (2005), 137–144.
- [136] UNO, T., KIYOMI, M., AND ARIMURA, H. Lcm ver.3: collaboration of array, bitmap and prefix tree for frequent itemset mining. In *OSDM '05: Proceedings of the 1st international workshop on open source data mining* (New York, NY, USA, 2005), ACM, pp. 77–86.
- [137] VAN HELDEN, J. Regulatory sequence analysis tools. *Nucleic Acids Res* 31(13) (2003), 3593–6.
- [138] VAN HELDEN, J., ANDRE, B., AND COLLADO-VIDES, J. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology* 281, 5 (1998), 827–842.
- [139] VAN HELDEN, J., RIOS, A., AND VIDES, J. C. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Research* 281 (1998), 827–842.
- [140] VENTER, J. C. Sequencing the human genome. In *Proceedings of the Sixth Annual International Conference on Computational Biology (RECOMB-02)* (New York, Apr. 18–21 2002), G. Myers, S. Hannenhalli, S. Istrail, P. Pevzner, and M. Waterman, Eds., ACM Press, pp. 309–309.
- [141] WASSERMAN, W., AND FICKET, J. Identification of regulatory regions which confer muscle-specific gene expression. *Journ of Mol. Biol* 278 (1998), 167–181.

-
- [142] WEINZIERL, R. *Mechanism of Gene Expression*. Imperial College Press, 1999.
- [143] WERNER, T. Models for prediction and recognition of eukaryotic promoters. *Mammalian Genome 10* (1999), 168–175.
- [144] WIJAYA, E., RAJARAMAN, K., YIU, S., AND SUNG, W. Detection of generic spaced motifs using submotif pattern mining. *Bioinformatics 23* (2007), 1476–1485.
- [145] WIJAYA, E., AND RAJARAMAN, K. *et.al.* A hybrid algorithm for motif discovery from DNA sequences. *3rd Asia-Pacific Bioinformatics Conference - Satellite Symposium* (2005).
- [146] WINGENDER, E., DIETZE, P., KARAS, H., AND KNÜPPEL, R. TRANSFAC: a database on transcription factors and their DNA binding sites. *Nucleic Acid Rresearch 24*, 1 (1996), 238–241.
- [147] WORKMAN, C., AND STORMO, G. ANN-Spec: a method for discovering transcription factor binding sites with improved specificity. In *Proceedings of Pacific Symposium of Biocomputing (PSB)* (2000), pp. 467–478.
- [148] YADA, T., TOTOKI, Y., ISHIKAWA, M., ASAI, K., AND NAKAI, K. Automatic extraction of motifs represented in hidden Markov model from a number of DNA sequences. *Bioinformatics 14* (1998), 317–325.
- [149] YAGI, H. *et al.* Regulation of the mouse histone H2A.X gene promoter by the transcription factor E2F and CCAAT binding protein. *J. Biol. Chem 270* (1995), 18759–18765.
- [150] ZHANG, X. *et.al.* Genome-wide analysis of camp-response element binding protein occupancy, phosphorylation, and target gene activation in human tissues. *PNAS 102* (2005), 4459–4464.

Appendix: Basic motif finders with their parameters used by MotifVoter

Below we describe the characteristics of the component motif finders used by MotifVoter.

- **Motif Finder:** AlignACE

Description and Parameters: AlignACE is a profile based motif discovery algorithm based on Gibbs Sampling method. Running parameters for AlignACE we set as the default, except the expected motif width was set to 15 upper bound. The major statistical score in AlignACE is maximum a posterior (MAP) score, being the larger the better.

URL: <http://atlas.med.harvard.edu/>

- **Motif Finder:** ANN-Spec

Description and Parameters: ANN-Spec is a profile based method. It uses Gibbs sampling for training positive examples. The scoring function is based on log likelihood that a binding sites binds at least once in the each sequence of positive training data versus the background sequence. Running parameter for ANN-Spec is set as default.

URL: <http://www.cbs.dtu.dk/~workman/ann-spec/>

- **Motif Finder:** BioProspector

Description and Parameters: BioProspector is another variant of Gibbs Sampling algorithm. We used the default values for the running parameters, except for the motif width, which was set to 15 upper bound. The background frequency model was generated using the whole genome of the species and the third order Markov model was used. BioProspector also uses maximum a posterior (MAP) to score the motifs.

URL: <http://robotics.stanford.edu/~xsliu/BioProspector/>

- **Motif Finder:** Improbizer

Description and Parameters: Improbizer uses expectation maximization to determine the profile of binding sites that occur improbably often in the input sequence. Running parameter for Improbizer is set to default.

URL: <http://www.soe.ucsc.edu/~kent/improbizer>

- **Motif Finder:** MDScan

Description and Parameters: MDScan is an enumerative deterministic greedy algorithm. Among its ten parameters, we only specified the following parameters. The motif width is set to maximum 15. The background frequency model was generated using the whole genome of the species and the third order Markov model was used. MDScan uses maximum a posterior (MAP) to score the motifs.

URL: <http://ai.stanford.edu/~xsliu/MDscan/>

- **Motif Finder:** MEME

Description and Parameters: MEME is an algorithm based on expectation maximization (EM) technique. MEME does not require user input like motif widths, because MEME can estimate by itself. And we set it to use *two component mixture* mode, in which it assume that the binding sites may appear more than once in a sequence. MEME uses *p*-value to score the motifs.

URL: <http://meme.sdsc.edu/>

- **Motif Finder:** MotifSampler

Description and Parameters: MotifSampler is another algorithm that uses Gibbs Sampling. It has seven major parameters. We use default values for all of them except motif widths is set to maximum 15. The background frequency model was generated using intergenic region sequences of the respective species genome and the third order Markov model was used. We use the information content score as the statistical measure to rank the motifs.

URL: <http://homes.esat.kuleuven.be/~thijs/Work/MotifSampler.html>

- **Motif Finder:** MotifSampler

Description and Parameters: MotifSampler is another algorithm that uses Gibbs Sampling. It has seven major parameters. We use default values for all of them except motif widths is set to maximum 15. The background frequency model was generated using intergenic region sequences of the respective species genome and the third order Markov model was used. We use the information content score as the statistical measure to rank the

motifs.

URL: <http://homes.esat.kuleuven.be/~thijs/Work/MotifSampler.html>

- **Motif Finder:** MITRA

Description and Parameters: MITRA is a consensus based motif-finder which is designed to find highly degenerate binding sites (weak signals). It uses specially designed data structure called mismatch tree. We let MITRA to search for maximum possible motif length which is 12. For the rest of two other parameters we use default values. MITRA uses information content score as the statistical measure to rank the motifs.

URL: <http://www.calit2.net/compbio/mitra>

- **Motif Finder:** SPACE

Description and Parameters: SPACE is also a consensus based motif finders. As a novel motif finding algorithm SPACE is based on a notion called submotifs. It aims to find a generic spaced motif by first finding submotif and then strategically compositing them using an efficient frequent submotif pattern mining approach. This framework provides the following novelties: the spacers could appear in more than two parts of the motif and their lengths need not be fixed. From the three running modes, we have chosen the *large* as the default parameter setting. The background frequency model uses seventh order Markov chain for the respective species intergenic sequence. For scoring it uses sequence specific and background score to rank the final motifs.

URL: <http://www.comp.nus.edu.sg/~bioinfo/SPACE>

- **Motif Finder:** Weeder

Description and Parameters: Weeder is a consensus based motif finders that uses exhaustive search. To speed-up the process it uses suffix tree as their data structure. From the three running modes, we have chosen the large as the default parameter setting. The background frequency model uses seventh order Markov chain for the respective species intergenic sequence. For scoring it uses sequence specific and background score to rank the final motifs.

URL: <http://159.149.109.16:8080/weederWeb/>