MCAT: Motif Combining and Association Tool

Yanshen Yang

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Application

Lenwood S. Heath, Chair
Liqing Zhang
Silke Hauf

May 7, 2018
Blacksburg, Virginia

Keywords: Motif finding, Ensemble algorithms, DNA, Transcription Factor Binding Site

MCAT: Motif Combining and Association Tool

Yanshen Yang

(ABSTRACT)

Motivation: *De novo* motif discovery in biological sequences is always an important and computationally challenging problem. In the past 20 years, a myriad of algorithms have been proposed to solve this problem with varying success. Ensemble algorithms, which combine different individual algorithms, have been introduced in previous studies, and it has been proved that an ensemble strategy can improve the prediction accuracy. However, the performance of these tools has not yet met most people's expectation. One reason for the low performance is failure to adapt to complicated and large data sets. Another existing problem is that fewer motif finding tools are available, and many of them are not maintained.

Results: I present a novel and fast tool MCAT (Motif Combining and Association Tool) for *de novo* motif discovery by combining six state-of-the-art motif discovery tools (MEME, BioProspector, DECOD, XXmotif, Weeder, and CMF). In addition, I developed an innovative motif combining algorithm, VoteRank, which is a position based algorithm that votes, ranks, and combines candidate motifs. By testing against DNA sequences from budding yeast, fission yeast, human, fruit fly, and mouse, I showed that MCAT is able to identify exact match motifs in DNA sequences efficiently and achieves at least 30% improvement in prediction accuracy.

MCAT: Motif Combining and Association Tool

Yanshen Yang

(GENERAL AUDIENCE ABSTRACT)

Finding hidden motifs in DNA or protein sequences is an important and computationally challenging problem. A motif is a short patterned DNA/protein sequence that has biological functions. Motifs regulate the process of gene expression, which is the fundamental biological process in which DNA is transcribed into RNA which is then translated to protein. In the past 20 years, a myriad of algorithms have been developed to solve the motif finding problem with varying success, but it can be difficult for even a small number of these tools to reach a consensus. Because individual tools can be better suited for specific scenarios, an ensemble tool that combines the results of many algorithms can yield a more confident and complete result. I present a novel and fast tool MCAT (Motif Combining and Association Tool) for motif discovery by combining six state-of-the-art motif discovery tools (MEME, BioProspector, DECOD, XXmotif, Weeder, and CMF). I apply MCAT to data sets with DNA sequences that come from various species and compare our results with two well-established ensemble motif finding tools, EMD and DynaMIT. The experimental results show that MCAT is able to identify exact match motifs in DNA sequences efficiently, and it has an improved performance in practice.

Acknowledgments

I would like to thank my advisor, Prof. Lenwood S. Heath for his guidance and support during my studies at Virginia Tech. His patience, encouragement and insightful suggestions greatly aided my research. I would also like to thank my committee members, Prof. Silke Hauf and Prof. Liqing Zhang for their valuable time and helpful comments.

I am thankful to all of my group members and former colleagues, Jeff Robertson, Zhen Guo, Christy Coghlan, and Jake Martinez for helping with the MCAT project, Doaa Altarawy for her advice at the beginning of my research, Haitham Elmarakeby for the Beacon project, and Xiao Liang for her valuable ideas and encouragement during my research.

I also want to thank the Department of Computer Science at Virginia Tech for giving me an opportunity to work in the bioinformatics area. As always, I would like to thank my family for their unconditional love and support over the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Gene expression is the fundamental biological process in which DNA is transcribed into RNA (transcription) which is then translated to protein (translation). The process of gene expression is essential in all known life, including eukaryotes, prokaryotes, and viruses. All cellular functions are coordinated by only a small percentage of the genome sequence in the process of gene expression. Thus, transcription needs to be qualitatively and quantitatively accurate to ensure the proper function of cells. Gene expression is usually controlled by transcription factors (TFs) that activate or inhibit the transcription machinery. Specifically, transcription factors are proteins that are involved in the process of transcribing DNA into messenger RNA (mRNA). TFs have distinct DNA-binding domains to bind to specific DNA sequences in the promoter region. The actual DNA-binding domains interacting with a TF is called a motif [13]. In general, a motif is a contiguous sequence found in both strands of the DNA sequences. Sometimes, it can also be gapped or palindromic [11]. The motif finding problem is usually defined as finding conserved short sequences across a subset of sequences with a common biological significance. Sequences used for motif finding are extracted from promoter regions of genes. Identifying motifs is one of the keys for understanding the mechanisms that regulate gene expression.

Eukaryotic transcripts are more complex than prokaryotic transcripts as the gene sequences of eukaryotes contain non-coding regions called introns and coding regions called exons. The promoter regions of mammals and *Drosophila melanogaster* have been widely studied, while

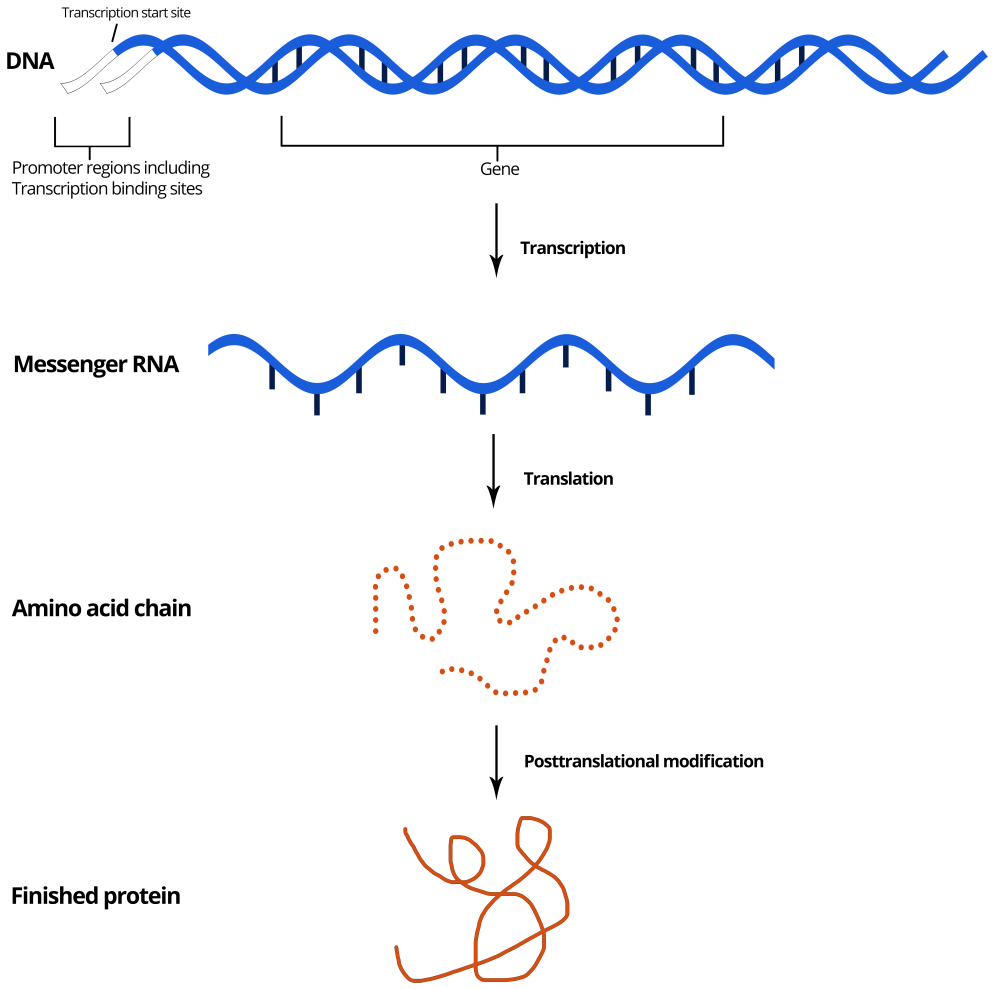

Figure 1.1: Gene expression at the molecular level. The process of gene expression has two major steps. Transcription is the the first step, in which the genetic information on DNA is copied to RNA. Translation is the second major step. In translation, Messenger RNA (mRNA) is decoded to produce amino acids. Then the amino acid chain is folded to form a protein.

for important unicellular eukaryotes, such as budding yeast (*Saccharomyces cerevisiae*) and fission yeast (*Schizosaccharomyces pombe*), the core promoter structures have not been com-

prehensively analyzed to date [26]. *S. pombe* is a unicellular eukaryote and has been subject to extensive experimental research in the study of basic principles of cell cycle regulation and cell division that can be used to understand more complex organisms like mammals and in particular humans. It contains one of the smallest numbers of genes yet recorded for a eukaryote [52].

In the past 20 years, numerous *de novo* motif discovery tools have been developed. These tools predict motifs using only a computational model without comparison to existing data. Motifs can be found computationally and then confirmed by experiment, making the process of motif identification significantly more efficient. For example, MEME [4] is the most commonly used tool in this area and uses multiple expectation maximizations to discover possible motifs within the input data. Weeder [37] exhaustively enumerates possible consensus sequences determined from precomputed frequency files and evaluates each one. However, a survey of DNA motif finding algorithms [11] revealed that most tools have been shown to have a better performance in yeast than that in higher organisms such as human and mouse. For the previous motif finding tools, early studies [47] indicated that the highest sensitivity and precision they can achieve are 13% and 35%, respectively. One reason for the low performance is that, when trying to find over-represented patterns from a set of sequences, patterns always contain mutations, insertions, or deletions of nucleotides.

There remain problems in the motif finding area. Besides the problem of low accuracy, another problem is that many motif finding tools for promoter analysis no longer exist as software that is maintained and can be downloaded. Before next-generation sequencing (NGS) was introduced, algorithms for motif finding were primarily designed for promoter analysis, which involved a small set of sequences with hundreds of base pairs (bps), while motif finding for ChIP-seq usually work on large genome-wide DNA data sets with short sequences. Even though we are now in the NGS era, focused promoter analysis is still

needed. However, I find many motif discovery tools for promoter analysis are no longer available. And most currently available motif finding tools are designed for ChIP sequences, such as DREME [3] and RSAT peak-motifs [46]. Finally, most ensemble pipelines take in FASTA for ChIP-Seq data only, and some limit the number of input sequences. In this case, an ensemble tool with greater sensitivity for a small number of longer sequences is desirable.

Since individual tools can be better suited for specific scenarios, ensemble tools that combine the results of many algorithms should be able to yield a more confident and complete result. According to the result of Hu's study [51], which tested against Tompa's data sets [47], ensemble motif finding tools show better performance compared to individual tools. The idea behind ensemble motif discovery is to run multiple existing tools, then process and combine their results to generate the final output. The idea of the ensemble approach has been widely applied in several biological areas, such as gene prediction [1] and protein domain prediction [40]. In the past 10 years, several ensemble tools for motif finding have been developed, combining different algorithmic methods, such as MotifVoter [54], GimmeMotifs [49], and EMD [20]. Some of these tools rely on a hidden Markov model [14], such as GimmeMotifs and MEME-ChIP [30]. Additionally, some compare the results against a database, such as JASPAR (Melina II [35]) or STAMP [31]. It has been demonstrated that the ensemble strategy for the motif discovery problem can effectively improve both the sensitivity and accuracy of the prediction [11].

After applying multiple algorithms, it is a challenge to combine the results from those different algorithms, which are based on different scoring and evaluation systems. For example, Melina II [35] does not perform any statistical analysis on the motif results and simply allows the user to run the results through an external tool. Ensembles like SCOPE [7] ranked the top motifs from component tools based on a scoring function. EMD and MotifVoter [51] proposed a clustering-based ensemble considering the binding sites when combining results

and ranking top motifs from several algorithms.

Here, we develop the Motif Combining and Association Tool (MCAT), a pipeline to find a single or multiple short motifs in the upstream (or promoter) regions of target genes believed to have common biological function. MCAT was initially motivated by the study of cell division in fission yeast. Cell division is a highly orchestrated process. Thousands of proteins are involved, with hundreds being important regulators. People want to understand how such a complex event can be reliably executed [15, 23]. Specifically, by using *Schizosaccharomyces pombe* (fission yeast) as a model organism, we want to discover those motifs, that are also hopefully transcription factor binding sites, existing in the promoter region of *S. pombe*, that regulate the mitotic checkpoint signaling procedure [16, 17]. The motifs do not necessarily appear exactly once in every sequence, one motif might appear one or multiple times in some sequences and not appear at all in others. MCAT searches for motifs by running six well-established motif discovery tools (MEME [4], BioProspector [28], CMF [32], DECOD[21], Weeder [37] and XXmotif [29]) on promoter sequences independently, then combines and reports the results using a novel algorithm we developed called VoteRank, see Section 4.1.2. We demonstrate that the MCAT approach provides an effective algorithm for improving sensitivity and accuracy of the prediction; see Chapter 5. The source code of MCAT is available at https://github.com/yanshen43/MCAT.

Chapter 2

Problem Definition

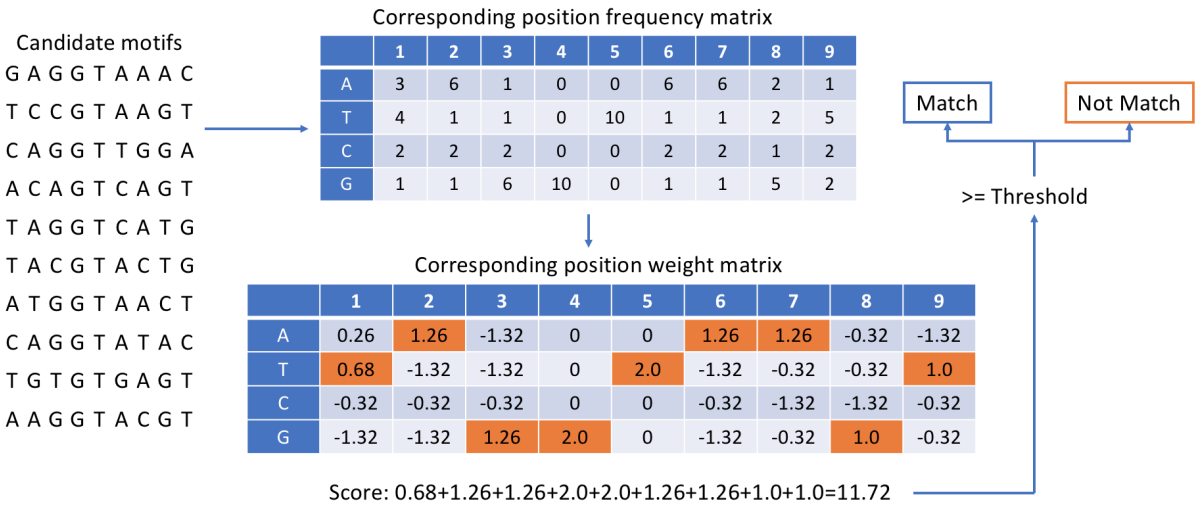

Figure 2.1: An example of Position Weight Matrix (PWM). To construct a PWM, the first step is to build position frequency matrix (PFM) by adding up the elements at each position. The next step is to convert PFM to PWM using log likelihoods. The highlighted scores are the highest scores at all positions. The sequence score is calculated by summing the highlighted scores. When given a threshold, the short sequence is identified as match if the sequences score is higher than the threshold.

The MCAT algorithm combines multiple motif finding algorithms. The motif finding problem is defined below. In detail, we consider DNA sequences and use a fixed alphabet $\sum = \{A, C, G, T\}$. We are given a set of DNA sequences $S = \{s_1, s_2, ..., s_m\}$ and a motif length L. We choose several motif discovery algorithms for the analysis. For each sequence s_i, the output is a set of candidate L-mer motifs c_1, c_2,...,c_n. MCAT retrieves all the motifs and then ranks and clusters motifs that maximize a consensus score. In particular, a motif

can be specified as a position weight matrix (PWM) [44] as shown in Figure 1. A PWM is a probabilistic description of nucleotide occurrences at each location of the motif, more specifically, it is a $4 \times L$ matrix with four rows for nucleotides and L columns for motif length. Often, the first step is to create a basic position frequency matrix (PFM) [44] by summing the occurrence of each nucleotide at each binding site. Then after having the PFM, a position probability matrix can be constructed by using the counts of each nucleotide by the total number of nucleotides at each position. Usually, the PWM employs the log likelihood ratios of observing a nucleotide in a binding site relative to genomic background [18]. The formulas used during the conversion is displayed below,

$$W_{i,j} = log_2 \frac{p(i,j)}{p(i)}, \tag{2.1}$$

where $p(i)$ is the background frequency of nucleotide i, and $p(i,j)$ is the frequency for nucleotide i at site j and can be calculated.

A particular site is evaluated by summing the elements from the scoring matrix at each position and comparing the sum to a match threshold. Sequence logo [10] can be applied to visualize PWMs.

Chapter 3

Review of Literature

This chapter reviews the literature related to the thesis. In Section 3.1, I discuss the related algorithms for the motif finding problem. Section 3.2 reviews the performance assessment for the motif finding problem.

3.1 Motif Finding Algorithms

In previous studies, various motif finding algorithms have been proposed for the motif finding problem, such as probabilistic algorithms, consensus-based methods, and ensemble algorithms. There are reviews focusing on individual motif finding tools [11], ensemble algorithms [22, 27] and Web applications for motif finding [48]. Although various tools are published in the past 20 years, we note that fewer tools are available now.

Probabilistic algorithms, which are profile-based algorithms, include methods such as expectation maximization (EM) [25] and Gibbs sampling [24]. The general procedure of a probabilistic algorithm is first to select some positions as the start positions for alignment, then align the related sequences, after that, it builds a PWM for each candidate motif and scores the motifs. This procedure will run iteratively and heuristically. The main difference between EM and Gibbs sampling is that Gibbs sampling will choose only one random initial starting position among all the input sequences, and it will give different results every time it runs. For instance, MEME [4] is EM based, and BioProspector [28] is based on Gibbs

sampling.

Another important algorithm for motif finding is consensus-based algorithm. The consensus-based method mostly relies on exhaustive enumeration by constructing multiple consensus strings and picking the ones with the highest scores. Therefore, the consensus-based method is a global optimization method and is guaranteed to find the best results. The disadvantage of consensus-based algorithm is that it usually reports similar results and has limitation on motif width. One example of a consensus-based tool is Weeder [37]. Consensus-based methods are usually more accurate but slower than probabilistic methods [55]. Consensus-based methods are more appropriate for motif finding in eukaryotes, which have shorter motifs in general. On the other hand, probabilistic methods are good choices for motif finding in prokaryotes [11].

Ensemble algorithms were introduced later. The general idea of an ensemble algorithm is to combine predications from multiple component algorithms in order to increase prediction accuracy. According to previous assessment experiments with Tompa [47, 51], ensemble algorithms can achieve up to 48% in precision, while individual algorithms can only achieve up to 30% in precision. Currently available ensemble algorithms can mainly be classified based on their combining methods [22].

Profile-based

Profile-based motif finding tools such as BEST [8], MotifVoter [51], and MProfiler [2] generally score each motif and report the top motifs based on their scores. The main idea of MotifVoter is to vote and cluster candidate motifs based on similarity measures, including motif similarity and cluster similarity. Melina II [35] employs a scoring function as well, however, it does not employ a combining function.

Clustering

Hu [20] developed an ensemble tool based clustering called EMD. EMD clusters and votes on motifs according to their similarities and positions in the corresponding sequences. MotifVoter and MProfiler are built based on ideas from EMD.

Machine Learning

With advancement of machine learning technology, many machine learning methods have been applied to solve motif finding problems. One earlier pipeline, MultiFinder [40], used hierarchical clustering and a statistic scoring function to combine results and merge motifs. Dassi [12] developed the toolkit DynaMIT for finding motifs in DNA and RNA sequences. They provided seven clustering strategies for integration, such as Jaccard similarity, and a PCA algorithm [38] as well as pairwise alignment.

Other Approaches

DMINDA [53] is an ensemble web application based on a phylogenetic footprinting framework consisting of five motif finding algorithms. As long as one single gene is conserved across input sequences, the phylogenetic footprinting method can identify it.

3.2 Performance Evaluation of Motif Finding Algorithms

Given the many motif finding tools, a robust assessment standard is needed for performance evaluation. Diversified data sets are expected to clarify the parameters that affect tools' performance. Biological data sets and synthetic data sets with planted motifs are often tested against some algorithms proposed before. Due to the limited knowledge of biological processes, some biological data sets may be too complicated to discriminate TF binding sites, while synthetic data sets may exhibit bias within a motif finding algorithm [41]. Further, tools may have a better performance on simpler organisms but have poor performance on

higher organisms [11].

Tompa [47] proposed a small-scale benchmark of data sets and evaluated thirteen available motif finding algorithms. The other goal of the benchmark is to assess future tools. The binding sites, locations and orientations were extracted from the TRANSFAC database [33]. The data sets come from four species: human, mouse, *Drosophila melanogaster*, and *Saccharomyces cerevisiae*. These consist of three types of sequence sets: (1) binding sites in their real promoter sequences, (2) binding sites planted in randomly chosen promoter sequences, and (3) binding sites planted in sequences generated according to a Markov model.

The results reported low performance of these tools including such tools as MEME [4] and Weeder [37]. Some limitations exists in Tompa's benchmarks. Tools have to return only one or zero motifs per sequence, and many binding sites are unusually long (31-71 bp). Furthermore, the results are still subject to human choices on parameters when running tools.

Later, another benchmark data set construction strategy [41] was presented based on Tompa's benchmarks from a machine learning prospective. They first described the discrimination algorithm suite to analyze data sets in the following steps: first the best possible discrimination scores of each data set are calculated and ranked according to the scores. Then it will help to choose subsets with specific properties so that it provides a good discrimination between positive and negative input sequences. They also proposed one model benchmark with data sets rely on common motif models. The three most common motif models include PWM, IUPAC codes, and mismatch strings. Exhaustive search algorithms have been developed to avoid any search bias.

Chapter 4

Methodology

4.1 The MCAT Algorithm

MCAT is based on the generation of a pool of short nucleotide sequences coupled with position votes and refinement. The MCAT pipeline consists of four stages: a masking stage, a tool extraction stage, a combining stage, and a scoring stage. MCAT executes each component motif finding tool independently and combines their results based on the VoteRank result with z-score and p-value.

DNA sequences usually contain regions with highly biased distributions of nucleotides, which is called low-complexity regions. Low complexity regions, such as simple tandem repeats and AT-rich regions, can lead to the high-score results that are biologically uninteresting. During the masking stage, the input sequences are filtered for low complexity regions by using the DUST module [34]. DUST module assigns a complexity score for each nucleotide. Any filtered nucleotides are then replaced with 'N' to indicate that they should not be considered as part of a motif.

In the tool extraction stage, once the low complexity regions are found and masked, the masked sequences are sent to each component motif finding tool and run multiple times independently. A set of candidate motifs and their respective sets of binding sites will be generated. The results from these tools are collected and scored against each other. The

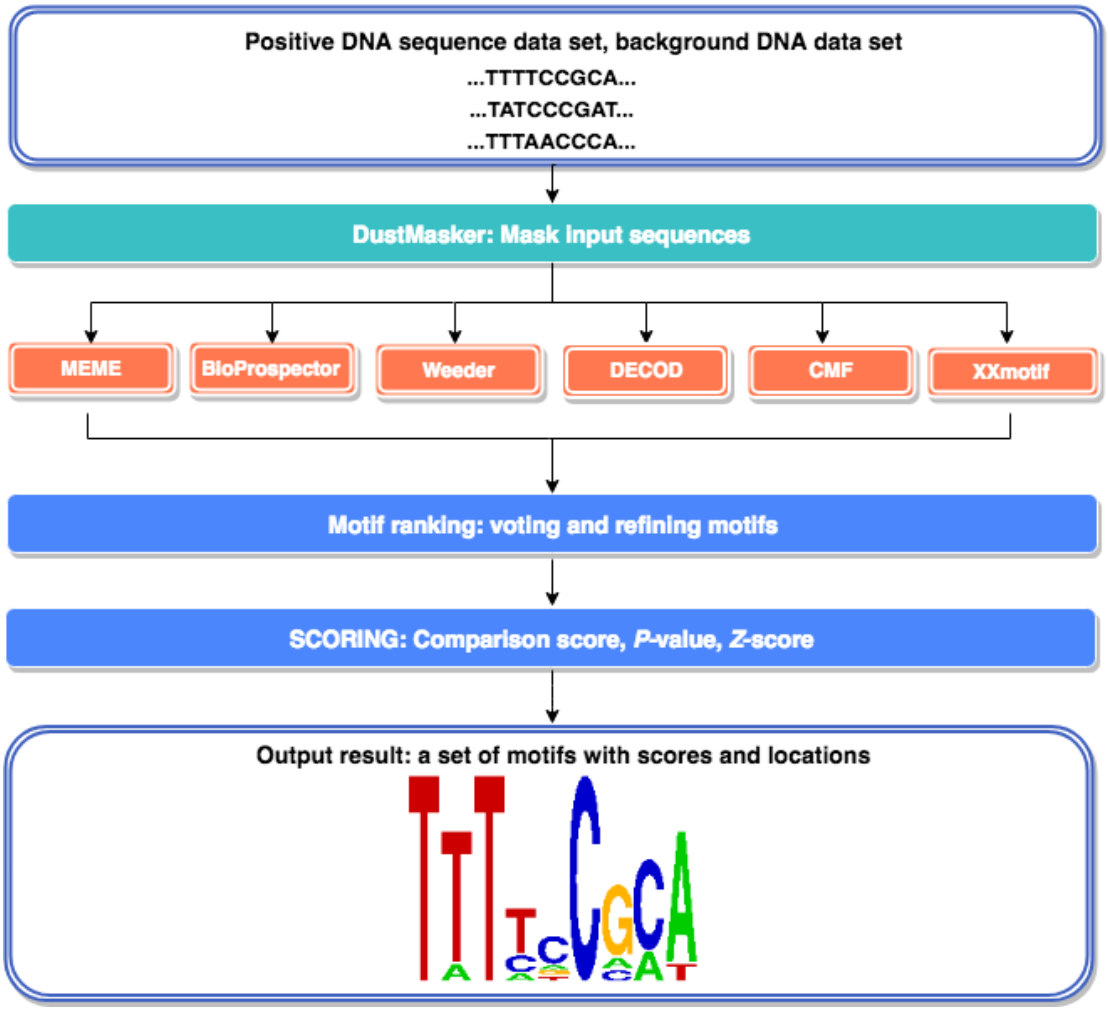

Figure 4.1: Architectural outline of the MCAT. MME, BioProspector, Weeder, DECOD, CMF and XXmotif are the motif-finding algorithms. Masked DNA sequences were sent to each algorithm and MotifRank will vote and refine motifs. The outputs of each motif will be shown with corresponding site locations.

original input files are searched for instances of the top scoring results. Based on these instances, the pipeline calculates statistical scores including z-score and p-value (see section 4.3) for each of the top results and visualizes them with Weblogo [10].

For the combining stage, MCAT combines independent results from each motif finding tool with novel VoteRank algorithm. VoteRank is described in Section 4.1.2.

4.1.1 Motif Finding Components in MCAT

The MCAT ensemble algorithm combines six tools. The details of the six tools are introduced below. These algorithms are based on different ideas, which makes combining results more reliable. Note that CMF, Weeder, and XXmotif do not allow the user to decide the specific length of the motif or the number of outputs.

MEME (Multiple EM for Motif Elicitation) [4] is one of the most widely used tools for motif finding. It uses multiple expectation maximizations (EM) algorithms to elicit motifs from the input data. It records each result, then masks it out, allowing MEME to find additional motifs without being distracted by motifs that have already been found.

BioProspector [28] is also a popular tool. It employs Gibbs sampling to find motifs. This is a randomized algorithm as opposed to the deterministic EM algorithm used in MEME, and the implementation in BioProspector allows for gapped motifs and motifs containing palindromic patterns. We use default values for all parameters.

CMF (Contrast Motif Finder) [32] is designed to discriminate between two sets of DNA sequences. It utilizes motif seeding to contrast a set of sequences believed to contain the desired motif and a negative set. The results are then derived from the motifs that are found to be over represented in the positive sample.

DECOD (DECOnvolved Discriminative motif discovery) [21] uses a discriminative approach to solve the problem of motif finding. Through deconvolution and heuristic hill climbing, DECOD attempts to search for a position weight matrix that maximizes a target function. The discriminative approach provides a significant difference in the methodology of this tool in comparison to the others, which allows the pipeline to be even more comprehensive.

Weeder [37] exhaustively enumerates possible consensus sequences determined from precom-

puted frequency files, evaluating each one against the sequences from the FASTA file. It does this multiple times with differing motif lengths and number of substitutions allowed, and then reports the top results.

XXmotif (eXhaustive, weight matriX-based motif discovery) [29] is a weight matriX-based motif finding web server that has three stages: masking stage, pattern stage, and position weight matrices (PWM) stage. It first masks out repeat regions, compositionally biased segments, and homologous segment pairs. Then it calculates enrichment p-values for 5-mer and 6-mer degenerate seed patterns, extends the best one, and merges similar PWMs until the p-value cannot be improved anymore.

4.1.2 VoteRank

VoteRank is the central new algorithm for MCAT. It votes, ranks and combines motifs from each motif finding tool. It is a position based algorithm that uses candidate motifs generated by any number of discovery tools and combines the identified regions to generate a smaller number of consensus motifs that fairly represent the significant results of all the tools. MCAT runs M motif finding algorithms on N input sequences and each algorithm reports a set of candidate motifs as well as the corresponding positions independently. Among all the candidate motifs, only some of them are real motifs. MCAT wants to retrieve and cluster the motifs that are given by as many motif finding tools as possible based on their votes.

Suppose the goal is to find the top K motifs. The overview of the VoteRank is presented in Figure 4.2. First, VoteRank conducts a poll, letting each tool vote iteratively for the positions where it found a motif, so as to count the score of results found at each position in each sequence. As each tool usually reports results with different numbers of motifs, instead

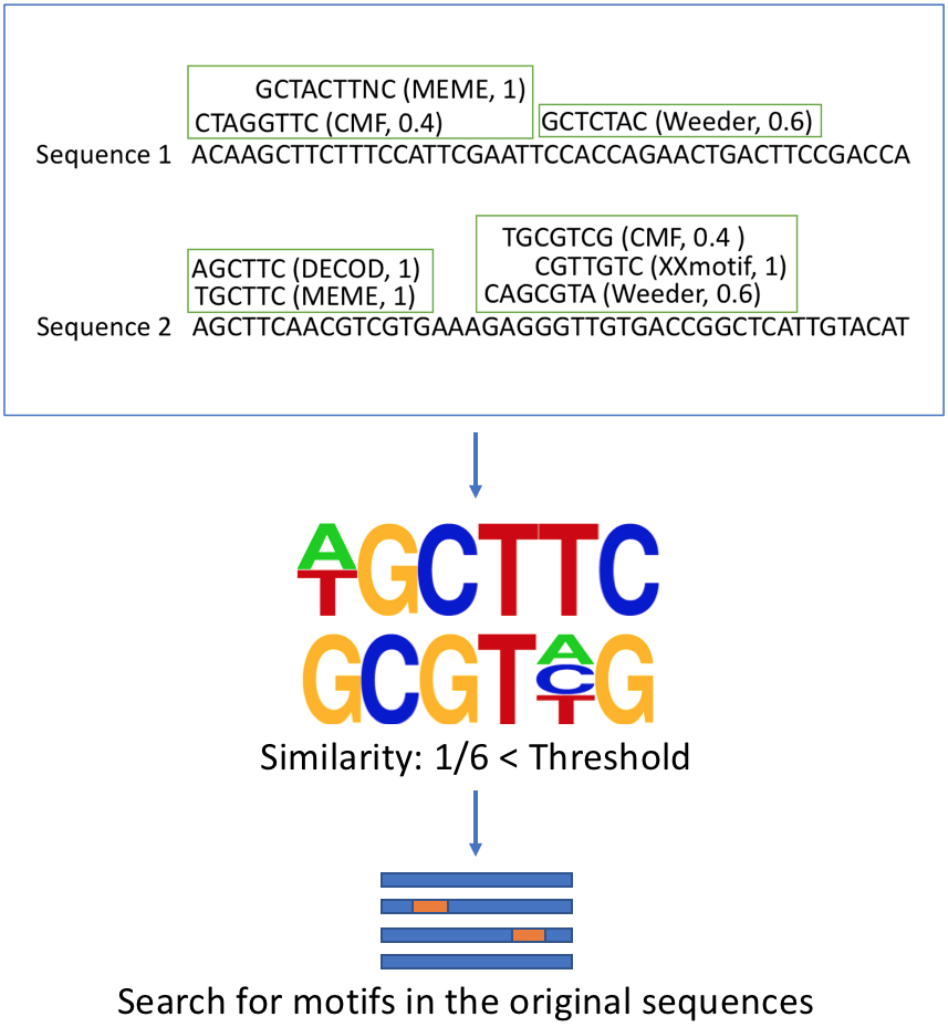

Figure 4.2: The figure shows the procedure of VoteRank. Each candidate motif votes iteratively for its corresponding position. The numbers following each motif represent their weights. For the heavily voted motifs, shown by the framed motifs, PWMs will be generated and a threshold is set to refine the poll. If the similarity is smaller than the threshold, VoteRank will verify the motifs by going back to the original sequences and searching for the motifs.

of giving all candidates motifs equal weights, VoteRank can bias selection based on tool performance and the number of outputs.

VoteRank goes through the previously conducted poll to compile the best PWMs of the

motifs with the most votes, and refines the motifs based on the PWM scores. For each sequence, supposed there are two motifs x and y that comes from two different tools. $R(x)$ refers to the region covered by motif x and $R(y)$ refers to the region covered by motif y. Let $R(x) \cup R(y)$ denotes to the region covered by both motifs. For example, in sequences 1, MCAT generates PWMs for the union of the motif GCTACTTNC and the motif CTAGGTTC, and picks the motif with highest PWM score. At the same time, MCAT calculates similarities of motifs, and it will combine motifs with high similarity. A similarity threshold is needed to make sure that the top K motifs are not too similar to each other.

The last step is to take the K top results and search for them in the original FASTA sequence files. It does this by looking at every position in each DNA sequence. Comparison scores are generated during this process by comparing each motif (and associated list of positions) with each motif found by a different tool and sums the results of each comparison. VoteRank will return any instances it finds that have one of the top K comparison scores to the motif. For motifs that are well conserved, this can return a surprisingly large number of results.

4.2 Performance Measurement

For performance evaluation, we use nucleotide level accuracy to measure the prediction accuracy, as defined in a previous study [19].

According to the scoring schema [55], index n represents that the assessment is at the nucleotide level. nTP is the number of true nucleotide binding site positions shown predicted as binding site; nFN is the number of target nucleotide binding site positions in known sites but not in predicted sites; nFP is the number of nucleotide binding site positions not in target binding sites but in predicted sites; nTN is the number of nucleotide binding site positions in neither target binding sites nor predicted sites. The confusion matrix is shown

in Table 4.1.

We can define sensitivity as,

$$nSn = \frac{nTP}{nTP + nFN} \tag{4.1}$$

We also use performance coefficient (nPC) for evaluation according to [19, 39], which is defined as,

$$nPC = \frac{nTP}{nTP + nFN + nFP} \tag{4.2}$$

The range of nPC is (0,1), and it considers both specificity and sensitivity in one single measurement. Then we have nucleotide level correlation coefficient (nCC) defined as below[6]:

$$nCC = \frac{nTP \cdot nTN - nFN \cdot nFP}{\sqrt{(nTP + nFN)(nTN + nFP)(nTP + nFP)(nTN + nFN)}} \tag{4.3}$$

Inspired by the motif level success rate score [19], we introduced the motif level prediction accuracy (mPa) to evaluate performance of MCAT. mPa is defined as the number of target motifs Nt, which match the predicted binding sites, divided by the total number of target

Table 4.1: The confusion matrix

Actual	Predicted	
	+	-
+	nTP	nFN
-	nFP	nTN

motifs Mt.

$$mPa = \frac{Nt}{Mt}$$

A prediction is considered to be a match when the predicted motif overlaps with the target motif by at least 75% of the length of the target motif.

4.3 Motif Scoring

A comparison score, a z-score, and a P-value of a matching motif to a set of candidate DNA sequences are computed for each of the candidate motifs. These scores indicate how strongly the various tools were in agreement about this result. MCAT also reports the log likelihood for the position weight matrix of each motif based on the background nucleotide ratios in the FASTA file [44].

4.3.1 Comparison Score

A comparison score is calculated by VoteRank when comparing each motif (and associated list of positions) with each motif found by a different tool and sum the results of each comparison. These individual comparisons are performed by finding a weighted edit distance between the two motifs, adding a constant multiplied by the reciprocal of the minimum distance between instance positions, and dividing by the square root of the length of the shorter of the sequences. A comparison score by itself only reflects on the motif quality compared to other comparison scores from within the same run of the pipeline but, taken together, can indicate how strongly the various tools were in agreement about this result.

4.3.2 A z-score

In general, to calculate the z-score for a sequence, one collects sequence alignment scores of one sequence to another random sequence. Then one computes the average score and standard deviation. In MCAT, motifs are sorted by the z-score. z-score is defined as,

$$z - score(m) = \frac{score - mean(score)}{std(score)}, \tag{4.4}$$

Here, score refers to the alignment score of this sequences to other random sequences. The z-score reveals the difference between observed count and expected count [42]. Higher z-scores are better, because the further the real score is from the mean, the more significant it is.

4.3.3 A p-value

To support the result of VoteRank, we evaluate each output motif against all the input sequences by p-value with QFAST [5]. A p-value is defined as the probability that an event occurs by chance. The range of p-value is (0,1), and the smaller the p-value is, the more significant the motif is. The original paper combines p-values from independent sources in the motif-finding problem. They computed the product of position p-values and sequence p-value, then computed the confidence of this product using the QFAST algorithm as the combined p-value.

Here, MCAT takes one query motif and compares it to each sequence of the target sequences. MCAT first finds the position in each of the target sequence that best matches the motif and calculate the position p-value, which is the probability of observing a match score at least as good from a random sequence. Then the position p-value is normalized for the length of

the sequence as the sequence p-value. All the independent p-values are multiplied together as the product and the combined p-value is the probability that a motif has a score better than the current product based on the distribution of the product.

4.4 Comparison with Existing Tools

I compared MCAT with two other ensemble tools, EMD [20] and DynaMIT [12], by running data sets including *Saccharomyces cerevisiae* data sets, *Schizosaccharomyces pombe* data sets, and the Tompa data sets.

One ensemble tool, EMD, works to make it simple to incorporate new component algorithms into the ensemble and is made to run on distributed computer systems. The sites that each algorithm predicts are grouped by the score that each algorithm gives them. Motifs that are in a top group as part of most of the algorithms are identified (all of the predicted sites are given equal weights). The number of times a position in the sequence is identified is counted and a sliding window is used to smooth the score and decide the final site prediction. It works by running the algorithms multiple times, often with different parameters, in order to achieve diverse predictions.

Another ensemble tool, DynaMIT, is completely customizable. It allows users to use whichever motif finding algorithms, combinations, and results printers that they choose. In the paper, DynaMIT was often tested by running MEME and GLAM2 or Weeder on the sequences, then running the results from each algorithm through novel combining methods, known as integration strategies, such as Jaccard similarity and PCA (Principal Component Analysis) algorithm.

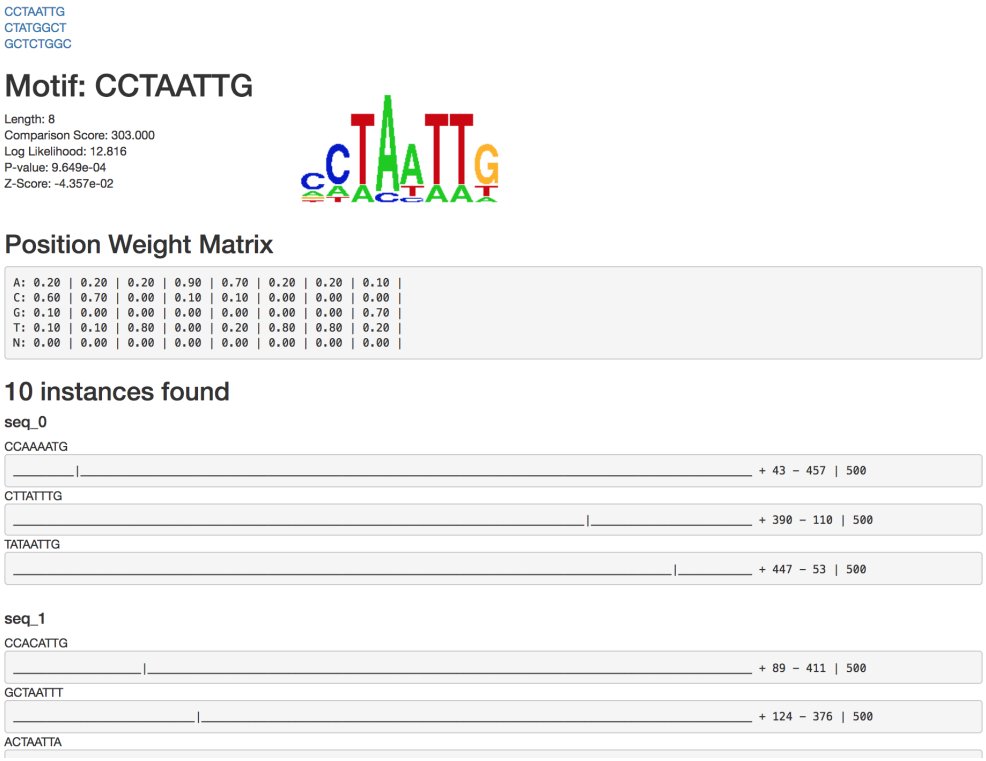

Figure 4.3: Visualization of one result. The HTML file first shows the top n motifs reported by MCAT. Then, it provides the information about each motif, including motif sequence, WebLogo, statistic scores, PWM, and positions at input sequences.

4.5 Visualization

WebLogo [10] provides the functionality in MCAT for creating sequence logos from the resulting position weight matrices generated by the pipeline. Additionally, a simplified motif location diagram is included along with the sequence logo in a generated HTML document, which displays the sequence name, the specific motif match, and the location of the motif match within the original sequence. One example of visualization is shown in Figure 4.3.

Chapter 5

Results

As MCAT was first inspired by the study of the cell cycle in fission yeast, we performed an extensive evaluation of MCAT over yeast data sets including *Saccharomyces cerevisiae* and *Schizosaccharomyces pombe*. Since *Saccharomyces cerevisiae* has been more fully studied compared to *Schizosaccharomyces pombe*, we extracted larger data sets from *Saccharomyces cerevisiae*. After that, we assessed the performance of MCAT against a well-known comprehensive data set from [47], which covered DNA sequences from human, fruit fly, mouse, and yeast.

When running MCAT, I used default settings when executing most of the algorithms except Weeder. The default width of motif is 8 and the top 5 motifs were chosen, ranked by comparison scores. For MEME and XXmotif, the zero or one occurrence per sequence (ZOOPs) model was used. As Weeder requires the specification of the organisms, I applied each data set with the corresponding organism. The number of iterations in DECOD is set to 35. I took the top 5 motifs when running the stand-alone tools and ensemble tools for comparison. For DynaMIT, I employed three of the motif search tools in it, MEME, Weeder, and GLAM2. The PCA Strategy was chosen as integration strategy in DynaMIT.

5.1 Comparison using *Saccharomyces cerevisiae* data sets

First, I evaluated MCAT using real *Saccharomyces cerevisiae* data sets. That data sets were obtained from Yeastract [45] (www.yeastract.com). The database contains promoter sequences, documented regulations, TF names with motifs and corresponding locations. Promoter sequences has gene name and sequences. I mapped and joined information into a table, which contains information such as for each sequence which motifs it contains. These will come up with a list of biological upstream sequences (motif is a consensus for a given TF which is known to regulate a given gene which has this upstream sequence) and biological negative sequences (motif is a consensus for a given TF which is not known to regulate a given gene which has this upstream sequence). Then picked smaller subsets among those sequences. I used hashing and suffix arrays to manually check if each biological upstream actually contained the motif, and marked the upstream as a strong sequence if it actually contained the motif, and as weak sequence otherwise. Then three sub-sets were generated with different numbers of combination of strong sequences and weak sequences. Specifically, the data set Budding Yeast 1 contains 58 test files, Budding Yeast 2 contains 107 test files, and Budding Yeast 3 contains 108 test files. Each test file contains 20 sequences. Sequence is retrieved from up to the 1000 bp upstream in the $5'$ UTR.

Figure 5.1 gives the results of prediction accuracy of running the six tools with three *Saccharomyces cerevisiae* data sets. The MCAT -2 refers to MCAT in which the components CMF and Weeder are given 0.4 and 0.6 weights, while others have a weight of 1. For MCAT, I selected the top five motifs, ranked by comparison scores. While for EMD and DynaMIT, I chose the top five motifs as well. Then I calculated the predication accuracy of the top five motifs from each ensemble tool. The prediction accuracy refers to motif level prediction

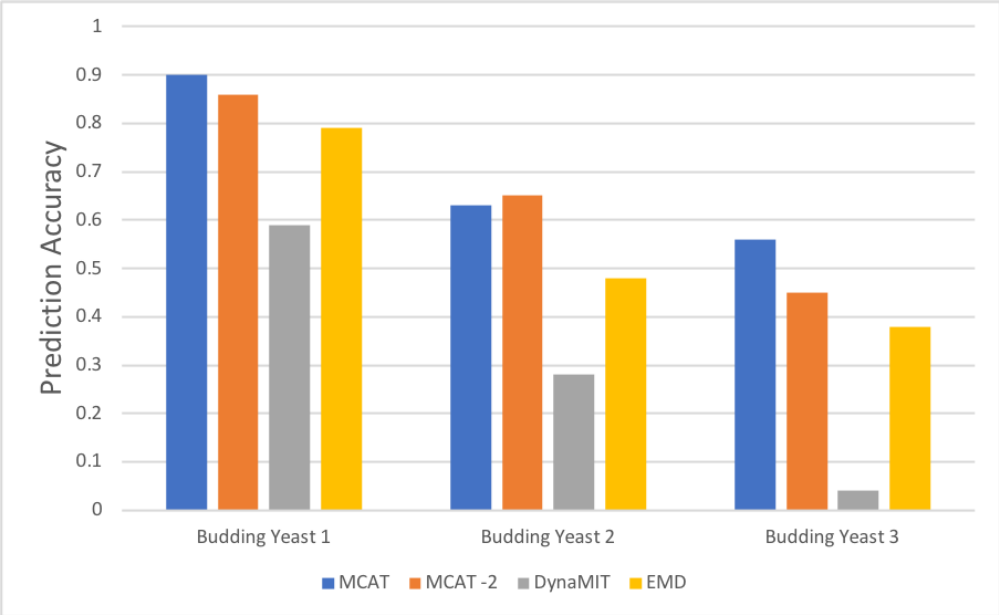

Figure 5.1: Comparison of *Saccharomyces cerevisiae* data sets. It displays the prediction accuracy on each of the three DNA data sets. MCAT -2 uses weights for CMF and Weeder that are 0.4 and 0.6, while the weights of other tools are 1.

accuracy. From the figure, we can tell that MCAT improved the accuracy of prediction by about 30% on average compared to EMD. The reason for the low predication accuracy of DynaMIT might be because it does not provide the option to set the width of the target motifs.

Figure 5.2 shows the performance of each motif finding component in MCAT. Compared to Figure 5.1, the performance of MCAT is better compared to each stand-alone tool. XXmotif gives the best performance compared to other algorithms. However, as users are unable to set the exact number of outputs, XXmotif has a number of false negatives.

Table 5.1 shows the average running time of each tool with each test file in *Saccharomyces cerevisiae* data sets. Figure 5.3 shows the test on the possible combinations of component algorithms with budding yeast *Saccharomyces cerevisiae* data sets and the result of prediction accuracy. Note that the combination of all six tools BioProspector (BP), MEME (ME),

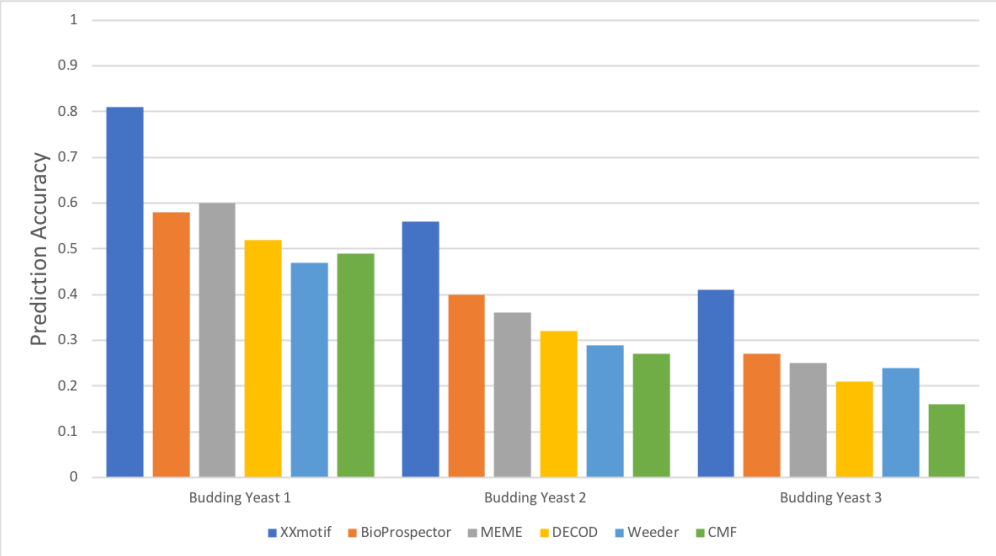

Figure 5.2: Prediction accuracy of each motif finding algorithm in MCAT. The result shows the performance assessment on three budding yeast *S.cerevisiae* data sets. The y-axis represents the prediction accuracy.

Weeder (WD), DECOD (DE), CMF, and XXmotif (XX) is slightly better than the result of five component algorithms without DECOD which only improved the result by 5% on average. It was noted that the execution time increases by at least two times in combinations with DECOD as the number of iterations increases.

Table 5.1: The table list the average running time for finding one motif in *Schizosaccharomyces cerevisiae* data sets. The values of running time is in seconds. MCAT-3 refers to MCAT with all component tools except DECOD, and all the tools are equal weight.

Tools	Range of running time (seconds)	Average running time (seconds)
MCAT	443 - 1216	879
MCAT-3	137 - 496	335
EMD	149 - 471	319
DynaMIT	106 - 369	293

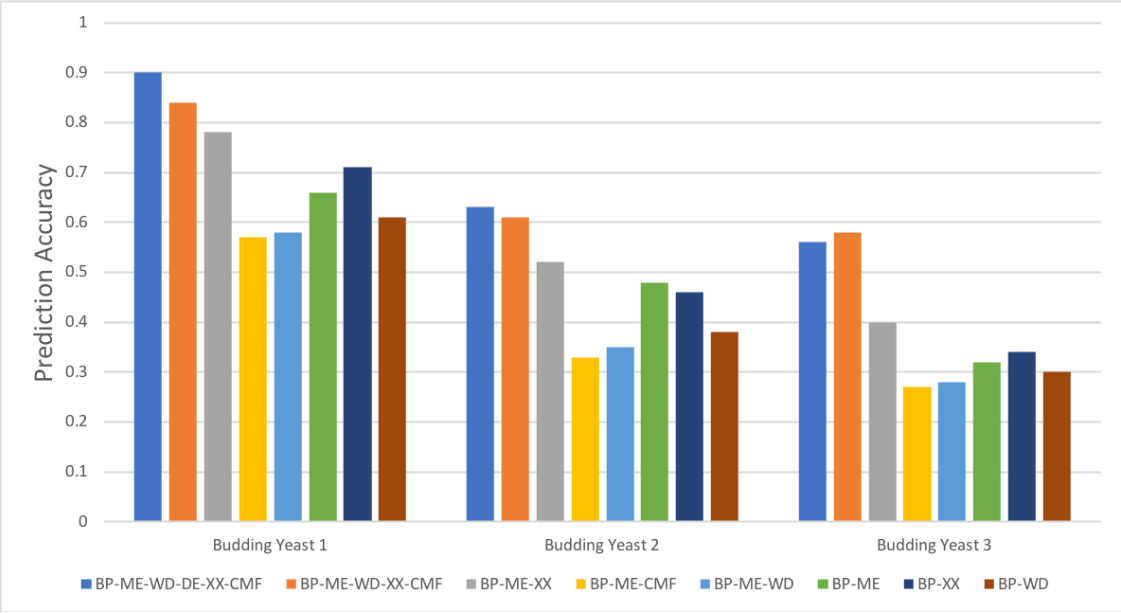

Figure 5.3: Prediction accuracy of different combinations of component algorithms. BP represents BioProspector, ME represents MEME, WD represents Weeder, DE represents DECOD, XX represents XXmotif. The result shows the MCAT performance assessment on three budding yeast *S.cerevisiae* data sets.

5.2 Comparison using *Schizosaccharomyces pombe* data sets

The *Schizosaccharomyces pombe* data set for testing was obtained from [36] and involves genes related to the cell cycle. In the paper, the authors extracted data from microarray scans and took up to 12,000 bp upstream sequences or to the next upstream open reading frame for motif analysis and applied MEME [4], an expectation-maximization algorithm; AlignACE [9], a Gibbs-sampling algorithm; and SPEXS (Sequence Pattern EXhaustive Search), a word-count algorithm [50] for motif analysis.

Here, when testing with MCAT, the data sets I used include a total of 23 genes from the core of the Cdc15 cluster (Cdc15), the 18 genes from the Cdc18 cluster (Cdc18), the 9 genes

Table 5.2: The table list the prediction accuracy of testing with *Schizosaccharomyces pombe* data sets. Cdc15, Cdc18, Eng1, and Wos2 represent clusters related to the cell cycle. Cdc15 and Cdc18 contain several motifs. The numbers refer to the prediction accuracy.

Cluster	Motif Name	Motif Consensus	MCAT	DynaMIT	EMD
Cdc15	FKH (P)	GTAAACAAA	0.8	0.4	0.6
	MBF/DSC1	ACGCG			
	ACE2	CCAGCC			
	New 1v	(A/T)TGCAAC			
	New 3v	CC(T/A)CG(T/C)TCC			
Cdc18	MBF/DSC1(P)	ACGCG	1	0.5	0.5
	Dbl10	ACGCG(A/T)CGCG			
Eng1	Ace2	CCAGCC	1	0	0
Wos2	HSE	NGAAN	1	1	1

from the Eng1 cluster and the 7 genes from the Wos2 cluster. Cdc15 contains five motifs, Cdc18 contains two motifs and both Eng1 and Wos2 contain one motif. For each gene, the DNA sequence examined was up to 2000 bp upstream of the 5′ UTR or to the next upstream open reading frame. The result of prediction accuracy is shown in Table 5.1. MCAT is able to find 8 out of 9 motifs which is 89% of the motifs in the four clusters, while DynaMIT found 4 motifs which achieves 45% and EMD found 5 motifs which achieves 56%.

5.3 Comparison using the Tompa data set

Other data sets used in the performance assessment were from Tompa et al. [47], which consists of 56 data sets. The data sets consist of real upstream sequences that come from human, mouse, *Drosophila melanogaster*, and *Saccharomyces cerevisiae* based on the TRANSFAC database [33]. The original data sets include one benchmark that has the binding sites in their actual genomic promoter sequences; one 'generic' benchmark, which has motifs randomly planted in the promoter sequences from the corresponding organism; and one 'markov' benchmark, which has motifs planted in sequences generated based on an order 3 Markov model. The numbers of sequences and sequence lengths in these data sets vary: there were 1-35 sequences per data set with each sequence of length up to 3000 bp.

Figure 5.4 shows the performance of individual and ensemble motif finding tools with the two performance evaluation measures, sensitivity (nSN) and correlation coefficient (nCC). Compared to individual component algorithms MEME, Weeder, and BioProspector, MCAT achieved an average 320% improvement in sensitivity and 280% in correlation coefficient, while compared to ensemble tools EMD and DynaMIT, MCAT gained over 110% in sensitivity and 100% in correlation coefficient.

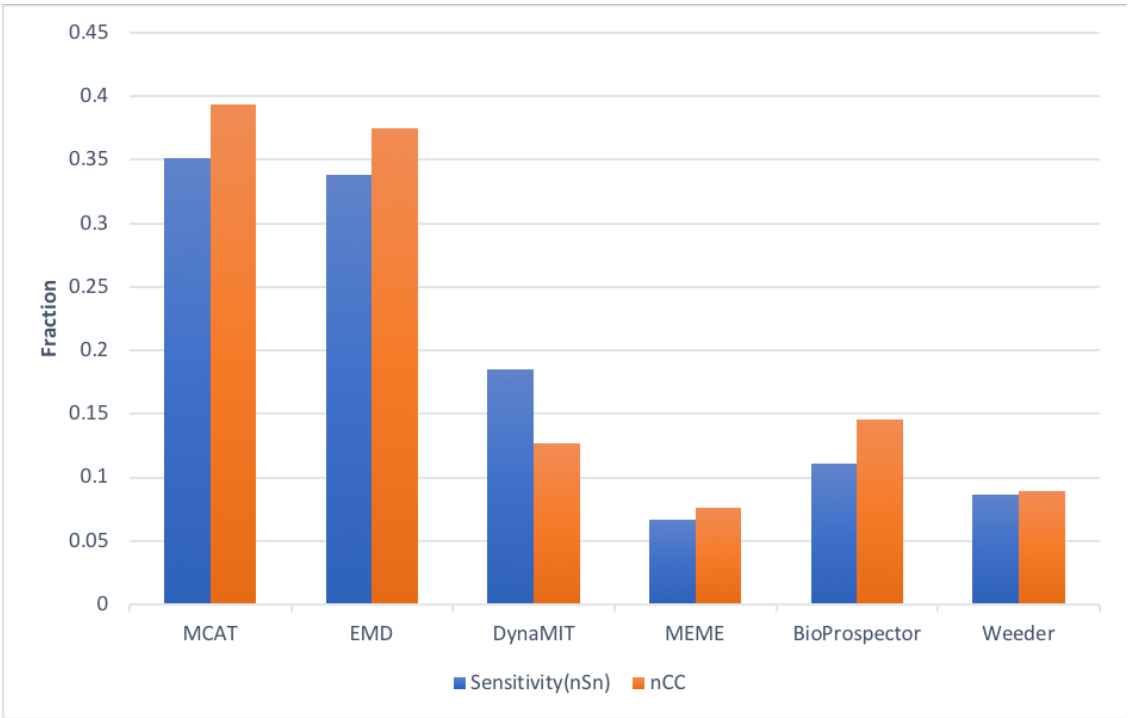

Figure 5.4: The figure lists the sensitivity (nSn) and correlation coefficient (nCC) of test for Tompa's benchmark. The blue bar represents sensitivity and the orange bar represents correlation coefficient. MCAT, EMD, and DynaMIT are ensemble motif finding tools, while MEME, BioProspector and Weeder are stand-alone motif finding tools.

Chapter 6

Discussion and Conclusions

We developed MCAT, a novel ensemble tool for motif finding in DNA sequences. For developing ensemble motif finding tools, one key problem is selecting the appropriate component algorithms. MCAT employees six individual motif finding algorithms, MEME, BioProspector, Weeder, CMF, DECOD, and XXmotif. It achieved good performance according to the results above. There is no best answer as to the selection of available methods for motif finding; it depends on the features of input data sets and performance evaluation [43]. While different combinations will produce different results, they do improve the results in many specific cases. In MCAT, it is essential that the six tools are diverse. MEME is based on EM methods, BioProspector is based on Gibbs sampling, Weeder is based on suffix trees and some variants. Both CMF and DECOD use a discriminative approach, while XXmotif is based on a scoring system with PWMs. These ensure that MCAT has a diversity so that motifs with high scores by different algorithms will be more confident and lead to good coverage.

Different combinations of component algorithms lead to different performance with the same data set. Figure 5.3 indicates the prediction accuracy of different combinations of component algorithms. In Figure 5.3, MCAT which combines the six tools has the better performance than any other combinations of tools, as well as EMD and DynaMIT. The performance of combining six tools or five tools are almost identical. However, Table 5.1 reveals that when it employs DECOD, the running time of MCAT increases by more than two times. The

reason for the DECOD does not make a large contribution to the final results might be that the algorithms of CMF and DECOD are highly similar, both of them are based on discriminative algorithm. And in order to get a strong result from DECOD, it is necessary to make the number of iterations high, which will significantly increase the running time. On the other hand, DynaMIT contains twelve motif finding algorithms, seven integration strategies. It allows users to employ different algorithms, integration strategies and visualizations. However, it failed to provide the guidelines for choosing different combinations of tools based on different data sets. In the process of testing DynaMIT, we noticed that some combinations work well while some not. Another problem for DynaMIT is that it lacks global parameters for configuration. In this case, users are unable to set parameters of target motifs such as the width of motifs and the number of output motifs, which may cause the low prediction accuracy. A future step is to analyze the quantitative relationship between the diversity of component algorithms and the final results, and provide guidelines for choosing the components of ensemble motif finding tools when dealing with different data sets.

Another key problem for an ensemble tool is how to combine independent results from different algorithms. The VoteRank of MCAT ensures that the top motifs have high PWMs scores and low similarity. In the process of combination, I want to make sure the PWMs generated in VoteRank are distinct, as well as the motifs in the final results. Many early motif finding tools output motifs that are extremely similar, which reduces the coverage of the results. Previous studies [22] summarized the ensemble methods, which mainly include naive ensemble methods, scoring functions, and clustering methods. Recently, more machine learning and deep learning methods have been applied on motif finding problem as well [56]. Generally the performance of these methods is better than that of individual motif finders.

To further improve the MCAT, identification of appropriate weights of different algorithms and the number of iterations to employ are important issues. In MCAT, I assign weights

to different algorithms based on the number of outputs and their performance. According to the result shown in Figure 5.1, the performance of MCAT with different weights is not significantly different than those of MCAT with all weights equal to one. One way to optimize the weights is to develop formulas to see whether that improves the performance, another way is to change the number of runs [20]. However, in MCAT, it is necessary to set an appropriate number of iterations because BioProspector and DECOD must run iteratively to obtain their final results. If the number of iteration is too low, it is unlikely to guarantee the accuracy of the results.

Overall, MCAT achieves an improved prediction accuracy when compared to individual motif finding tools and other ensemble approaches. Additional work includes further insights with new data sets and implementation for platforms other than Linux as well as a Web application.

Bibliography

[1] G. Aggarwal, E. Worthey, P. D. McDonagh, and P. J. Myler, *Importing statistical measures into Artemis enhances gene identification in the leishmania genome project*, BMC Bioinformatics, 4 (2003), p. 23.

[2] D. Altarawy, M. A. Ismail, and S. M. Ghanem, *MProfiler: A profile-based method for DNA motif discovery*, in IAPR International Conference on Pattern Recognition in Bioinformatics, Springer, 2009, pp. 13–23.

[3] T. L. Bailey, *DREME: Motif discovery in transcription factor ChIP-seq data*, Bioinformatics, 27 (2011), pp. 1653–1659.

[4] T. L. Bailey, M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li, and W. S. Noble, *MEME SUITE: Tools for motif discovery and searching*, Nucleic Acids Research, 37 (2009), pp. W202–W208.

[5] T. L. Bailey and M. Gribskov, *Combining evidence using p-values: Application to sequence homology searches*, Bioinformatics, 14 (1998), pp. 48–54.

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

[7] J. M. Carlson, A. Chakravarty, C. E. DeZiel, and R. H. Gross, *SCOPE: A web server for practical denovo motif discovery*, Nucleic Acids Research, 35 (2007), pp. W259–W264.

[8] D. Che, S. Jensen, L. Cai, and J. S. Liu, *BEST: Binding-site estimation suite of tools*, Bioinformatics, 21 (2005), pp. 2909–2911.

[9] X. Chen, L. Guo, Z. Fan, and T. Jiang, *W-AlignACE: An improved Gibbs sampling algorithm based on more accurate position weight matrices learned from sequence and gene expression/ChIP-chip data*, Bioinformatics, 24 (2008), pp. 1121–1128.

[10] G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner, *WebLogo: A sequence logo generator*, Genome Research, 14 (2004), pp. 1188–1190.

[11] M. K. Das and H.-K. Dai, *A survey of DNA motif finding algorithms*, BMC Bioinformatics, 8 (2007), p. S21.

[12] E. Dassi and A. Quattrone, *DynaMIT: The dynamic motif integration toolkit*, Nucleic Acids Research, 5, pp. e2–e2.

[13] P. D'haeseleer, *What are DNA sequence motifs?*, Nature Biotechnology, 24 (2006), p. 423.

[14] K. Ellrott, C. Yang, F. M. Sladek, and T. Jiang, *Identifying transcription factor binding sites through Markov chain optimization*, Bioinformatics, 18 (2002), pp. S100–S109.

[15] X. He, T. E. Patterson, and S. Sazer, *The Schizosaccharomyces pombe spindle checkpoint protein mad2p blocks anaphase and genetically interacts with the anaphase-promoting complex*, Proceedings of the National Academy of Sciences, 94 (1997), pp. 7965–7970.

[16] S. Heinrich, E.-M. Geissen, J. Kamenz, S. Trautmann, C. Widmer, P. Drewe, M. Knop, N. Radde, J. Hasenauer, and S. Hauf, *Determinants of robustness in spindle assembly checkpoint signalling*, Nature Cell Biology, 15 (2013), p. 1328.

[17] S. Heinrich, K. Sewart, H. Windecker, M. Langegger, N. Schmidt, N. Hustedt, and S. Hauf, *Mad1 contribution to spindle assembly checkpoint signalling goes beyond presenting mad2 at kinetochores*, EMBO Reports, 15 (2014), pp. 291–298.

[18] G. Z. Hertz and G. D. Stormo, *Identifying DNA and protein patterns with statistically significant alignments of multiple sequences.*, Bioinformatics, 15 (1999), pp. 563–577.

[19] J. Hu, B. Li, and D. Kihara, *Limitations and potentials of current motif discovery algorithms*, Nucleic Acids Research, 33 (2005), pp. 4899–4913.

[20] J. Hu, Y. D. Yang, and D. Kihara, *EMD: An ensemble algorithm for discovering regulatory motifs in DNA sequences*, BMC Bioinformatics, 7 (2006).

[21] P. Huggins, S. Zhong, I. Shiff, R. Beckerman, O. Laptenko, C. Prives, M. H. Schulz, I. Simon, and Z. Bar-Joseph, *DECOD: Fast and accurate discriminative dna motif finding*, Bioinformatics, 27 (2011), pp. 2361–2367.

[22] J. Kim, S. Yu, and S. Yoon, *Ensemble algorithms for DNA motif finding*, in Electronics, Information and Communications (ICEIC), 2014 International Conference on, IEEE, 2014, pp. 1–2.

[23] S. H. Kim, D. P. Lin, S. Matsumoto, A. Kitazono, and T. Matsumoto, *Fission yeast slp1: An effector of the mad2-dependent spindle checkpoint*, Science, 279 (1998), pp. 1045–1047.

[24] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton, *Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment*, Science, 262 (1993), pp. 208–214.

[25] C. E. Lawrence and A. A. Reilly, *An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences*, Proteins: Structure, Function, and Bioinformatics, 7 (1990), pp. 41–51.

[26] H. Li, J. Hou, L. Bai, C. Hu, P. Tong, Y. Kang, X. Zhao, and Z. Shao, *Genome-wide analysis of core promoter structures in Schizosaccharomyces pombe with DeepCAGE*, RNA Biology, 12 (2015), pp. 525–537.

[27] A. Lihu and Ş. Holban, *A review of ensemble methods for denovo motif discovery in ChIP-seq data*, Briefings in Bioinformatics, 16 (2015), pp. 964–973.

[28] X. Liu, D. L. Brutlag, and J. S. Liu, *Bioprospector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes*, in Biocomputing 2001, World Scientific, 2000, pp. 127–138.

[29] S. Luehr, H. Hartmann, and J. Söding, *The XXmotif web server for exhaustive, weight matrix-based motif discovery in nucleotide sequences*, Nucleic Acids Research, 40 (2012), pp. W104–W109.

[30] P. Machanick and T. L. Bailey, *MEME-ChIP: Motif analysis of large DNA datasets*, Bioinformatics, 27 (2011), pp. 1696–1697.

[31] S. Mahony and P. V. Benos, *STAMP: A web tool for exploring DNA-binding motif similarities*, Nucleic Acids Research, 35 (2007), pp. W253–W258.

[32] M. J. Mason, K. Plath, and Q. Zhou, *Identification of context-dependent motifs by contrasting ChIP binding data*, Bioinformatics, 26 (2010), pp. 2826–2832.

[33] V. Matys, E. Fricke, R. Geffers, E. Gössling, M. Haubrock, R. Hehl, K. Hornischer, D. Karas, A. E. Kel, O. V. Kel-Margoulis, et al., *TRANS-*

FAC: Transcriptional regulation, from patterns to profiles, Nucleic Acids Research, 31 (2003), pp. 374–378.

[34] A. Morgulis, E. M. Gertz, A. A. Schäffer, and R. Agarwala, *A fast and symmetric DUST implementation to mask low-complexity DNA sequences*, Journal of Computational Biology, 13 (2006), pp. 1028–1040.

[35] T. Okumura, H. Makiguchi, Y. Makita, R. Yamashita, and K. Nakai, *Melina II: A web tool for comparisons among several predictive algorithms to find potential motifs from promoter regions*, Nucleic Acids Research, 35 (2007), pp. W227–W231.

[36] A. Oliva, A. Rosebrock, F. Ferrezuelo, S. Pyne, H. Chen, S. Skiena, B. Futcher, and J. Leatherwood, *The cell cycle–regulated genes of Schizosaccharomyces pombe*, PLoS Biology, 3 (2005), p. e225.

[37] G. Pavesi, P. Mereghetti, G. Mauri, and G. Pesole, *Weeder web: Discovery of transcription factor binding sites in a set of sequences from co-regulated genes*, Nucleic Acids Research, 32 (2004), pp. W199–W203.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., *Scikit-learn: Machine learning in Python*, Journal of machine learning research, 12 (2011), pp. 2825–2830.

[39] P. A. Pevzner and S.-H. Sze, *Combinatorial approaches to finding subtle signals in DNA sequences.*, in International Society for Computational Biology, 2000, pp. 269–278.

[40] H. K. Saini and D. Fischer, *Meta-DP: Domain prediction meta-server*, Bioinformatics, 21 (2005), pp. 2917–2920.

[41] G. K. Sandve, O. Abul, V. Walseng, and F. Drabløs, *Improved benchmarks for computational motif discovery*, BMC Bioinformatics, 8 (2007), p. 193.

[42] S. Schbath, *Statistics of motifs*, Atelier de formation, 1502 (2006).

[43] C. Schweikert, S. Brown, Z. Tang, P. R. Smith, and D. F. Hsu, *Combining multiple ChIP-seq peak detection systems using combinatorial fusion*, BMC Genomics, 13 (2012), p. S12.

[44] G. D. Stormo, T. D. Schneider, L. Gold, and A. Ehrenfeucht, *Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli*, Nucleic Acids Research, 10 (1982), pp. 2997–3011.

[45] M. C. Teixeira, P. T. Monteiro, M. Palma, C. Costa, C. P. Godinho, P. Pais, M. Cavalheiro, M. Antunes, A. Lemos, T. Pedreira, et al., *YEASTRACT: An upgraded database for the analysis of transcription regulatory networks in Saccharomyces cerevisiae*, Nucleic Acids Research, 46 (2017), pp. D348–D353.

[46] M. Thomas-Chollier, C. Herrmann, M. Defrance, O. Sand, D. Thieffry, and J. van Helden, *RSAT peak-motifs: Motif analysis in full-size ChIP-seq datasets*, Nucleic Acids Research, 40 (2011), pp. e31–e31.

[47] M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. De Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, et al., *Assessing computational tools for the discovery of transcription factor binding sites*, Nature Biotechnology, 23 (2005), pp. 137–144.

[48] N. T. L. Tran and C.-H. Huang, *A survey of motif finding web tools for detecting binding site motifs in ChIP-seq data*, Biology Direct, 9 (2014), p. 4.

[49] S. J. van Heeringen and G. J. C. Veenstra, *GimmeMotifs: A denovo motif prediction pipeline for ChIP-sequencing experiments*, Bioinformatics, 27 (2010), pp. 270–271.

[50] J. Vilo, A. Brazma, I. Jonassen, A. J. Robinson, and E. Ukkonen, *Mining for putative regulatory elements in the yeast genome using gene expression data.*, in International Society for Computational Biology, 2000, pp. 384–394.

[51] E. Wijaya, S.-M. Yiu, N. T. Son, R. Kanagasabai, and W.-K. Sung, *Motifvoter: A novel ensemble method for fine-grained integration of generic motif finders*, Bioinformatics, 24 (2008), pp. 2288–2295.

[52] V. Wood, R. Gwilliam, M.-A. Rajandream, M. Lyne, R. Lyne, A. Stewart, J. Sgouros, N. Peat, J. Hayles, S. Baker, et al., *The genome sequence of Schizosaccharomyces pombe*, Nature, 415 (2002), p. 871.

[53] J. Yang, X. Chen, A. McDermaid, and Q. Ma, *DMINDA 2.0: Integrated and systematic views of regulatory DNA motif identification and analyses*, Bioinformatics, 33 (2017), pp. 2586–2588.

[54] P. Yang, Y. Hwa Yang, B. B Zhou, and A. Y Zomaya, *A review of ensemble methods in bioinformatics*, Current Bioinformatics, 5 (2010), pp. 296–308.

[55] F. Zare-Mirakabad, H. Ahrabian, M. Sadeghi, A. Nowzari-Dalini, and B. Goliaei, *New scoring schema for finding motifs in DNA sequences*, BMC Bioinformatics, 10 (2009), p. 93.

[56] H. Zhang, L. Zhu, and D.-S. Huang, *WSMD: weakly-supervised motif discovery in transcription factor ChIP-seq data*, Scientific Reports, 7 (2017), p. 3217.