Pattern Discovery for Deciphering Gene Regulation Based on Evolutionary Computation

CHAN, Tak Ming

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

The Chinese University of Hong Kong
September 2010

Thesis/Assessment Committee

Professor WONG, Man Hon (Chair)
Professor LEUNG, Kwong Sak (Thesis Supervisor)
Professor LEE, Kin Hong (Thesis Supervisor)
Professor LEUNG, Ho Fung (Committee Member)
Professor KWONG, Tak Wu Sam (External Examiner)

Abstract of thesis entitled:

Pattern Discovery for Deciphering Gene Regulation Based on Evolutionary Computation

Submitted by CHAN, Tak Ming

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in July 2010

Transcription Factor (TF) and Transcription Factor Binding Site (TFBS) bindings are fundamental protein-DNA interactions in transcriptional regulation. TFs and TFBSs are conserved to form patterns (motifs) due to their important roles for controlling gene expressions and finally affecting functions and appearances. Pattern discovery is thus important for deciphering gene regulation, which has tremendous impacts on the understanding of life, bio-engineering and therapeutic applications. This thesis contributes to pattern discovery involving TFBS motifs and TF-TFBS associated sequence patterns based on Evolutionary Computation (EC), especially Genetic Algorithms (GAs), which are promising for bioinformatics problems with huge and noisy search space.

On TFBS motif discovery, three novel GA based algorithms are developed, namely GALF-P with focus on optimization, GALF-G for modeling, and GASMEN for spaced motifs. Novel memetic operators are introduced, namely local filtering and probabilistic refinement, to significantly improve effectiveness (e.g. 73% better than MEME) and efficiency (e.g. 4.49 times speedup) in search. The GA based algorithms have been extensively tested on comprehensive synthetic, real and benchmark

datasets, and shown outstanding performances compared with state-of-the-art approaches. Our algorithms also "evolve" to handle more and more relaxed cases, namely from fixed motif widths to most flexible widths, from single motifs to multiple motifs with overlapping control, from stringent motif instance assumption to very relaxed ones, and from contiguous motifs to generic spaced motifs with arbitrary spacers.

TF-TFBS associated sequence pattern (rule) discovery is further investigated for better deciphering protein-DNA interactions in regulation. We for the first time generalize previous exact TF-TFBS rules to approximate ones using a progressive approach. A customized algorithm is developed, outperforming MEME by over 73%. The approximate TF-TFBS rules, compared with the exact ones, have significantly more verified rules and better verification ratios. Detailed analysis on PDB cases and conservation verification on NCBI protein records illustrate that the approximate rules reveal the flexible and specific protein-DNA interactions with much greater generalized capability.

The comprehensive pattern discovery algorithms developed will be further verified, improved and extended to further deciphering transcriptionial regulation, such as inferring whole gene regulatory networks by applying TFBS and TF-TFBS patterns discovered and incorporating expression data.

摘要

香港中文大學工程學院
計算機科學及工程學系
哲學博士
陳德銘

轉錄因子（TF）和轉錄因子結合位點（TFBS）的結合（binding）是轉錄調控中基礎的蛋白質-脫氧核糖核酸（DNA）相互作用。由於其控制基因表達的重要角色，TFs 和 TFBSs 會形成保守的模式（模體），最終影響生物功能和外觀。因此，模式發現對破譯基因調控甚爲重要，而破譯基因調控對生命的理解，生物工程和治療應用具有巨大的影響。本論文以進化計算（EC），特別是遺傳算法（GA）作為基礎框架，集中解決 TFBS 和 TF-TFBS 結合序列的模式發現問題，因爲 GA 十分有利於解決牽涉到龐大和嘈雜搜索空間的生物信息學問題。

針對 TFBS 模體發現問題，我們開發了三種基於 GA 的新型算法，即以優化為目標的 GALF-P，著重建模的 GALF-G，以及處理間隔模體的 GASMEN。我們引入新型的文化基因算子(memetic operators)，即局部過濾和概率細化，大大提高搜索的效用（如比 MEME 改進 73%）和效率（如 4.49 倍的提速）。我們對以上算法進行了廣泛全面的綜合測試，他們較其他尖端方法有更優越的表現。我們的算法也“演變”以能夠處理更廣義和寬鬆的情況，如靈活的模體寬度，擁有重疊控制的多模體發現，寬鬆的模體個體數目假設，有任意間隔的模體發現等等。

我們也進一步解決 TF-TFBS 結合序列的模式（簡稱規則）發現問題，以便日後更好地破譯調控中的蛋白質-DNA 相互作用。我們使用循序漸進的方式，首次以近似規則來廣義化之前的精確 TF-TFBS 規則。我們定制的算法比 MEME 改進超過 73%。TF-TFBS 近似規則比精確規則有顯著更多能夠被驗證的規則和更好的驗證率。蛋白質數據庫（PDB）的詳細實例分析以及 NCBI 蛋白質記錄的保守性驗證表明，近似規則能更廣義地揭示蛋白質-DNA 相互作用中的靈活性和特定性。

我們將進一步驗證，改進和擴展之前開發的模式發現算法，進一步破譯轉錄調節，如利用已發現的 TFBS 及 TF-TFBS 模式，結合微陣列數據預測整個基因調控網絡。

Acknowledgement

I would like to thank my supervisors Prof. Kwong-Sak Leung and Prof. Kin-Hong Lee. This thesis would not have been possible without their seasoned guidance, help and inspirations throughout my PhD research. I would also like to thank the internal examiners Prof. Ho-Fung Leung, Prof. Man-Hon Wong, as well as the external examiner Prof. Sam Kwong, for their valuable comments on my work. I am indebted to our collaborators, Prof. Stephen Tsui, Prof. Terrence Lau, Dr. Pietro Lio' and Prof. Yong Liang (also a senior group member) for the knowledge and research experience they have shared with me.

It is a pleasure to thank all the group members, especially those who have worked closely with me: Dr. Gang Li, David Lam, Peter Lo and Ricky Wong; those who are now doing research with me (in alphabetical order): Jacky Li, Kent Ling, Shaoke Lou, Bing Ni, Benjamin Tse and Jimmy Yip; as well as the graduated senior members who offered me the initial help and guidance: Wai-Shing Lau, Dr. Wenye Li, Dr. Yong Liang, Dr. Wing-Ho Shum, Sui-Man Tse and Dr. Jinfeng Wang. I would also like to thank the lab mates with whom I have studied, discussed and played together in Room 1013: Fai Chan, Roy Chan, Dr. Josh Chen, Norman Chen, Hung-Kwan Fung, Hoi-Fung Ko, Dr. Louis Tang and Dr. Li Teng. I am thankful to all the general office staff for their kind help. I would like to express my gratefulness to my old friends: Eric Li, Matthew Li, Ellis Liu, Felix Xiao and Feia Zhao for their encouragement and support.

Finally, I owe my deepest gratitude to my parents, and Cloris Ho, for their greatest love and support all along.

Dedicated to Cloris Ho and my parents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Summary

This chapter introduces the brief Bioinformatics background, presents the major contributions of this thesis, and finally gives the thesis outline.

1.1 Bioinformatics

In this section, briefings on Bioinformatics related to transcriptional regulation are first introduced, and then our focus on pattern discovery based on evolutionary computation is presented. Formal details will be described in the Background chapter.

1.1.1 Bioinformatics for Deciphering Gene Regulation

Bioinformatics is the application of informatics (computer science), usually based on mathematics and statistics models, to the field of molecular biology. Bioinformatics is an emerging and interdisciplinary field exhibiting more and more importance and becoming more and more crucial in life sciences.

In the recent past, Bioinformatics mainly helped to collect the massive data in an automatic way, such as creation and maintenance of databases for sequences and annotated genes, while major analysis and discovery awaited expensive, labor and time intensive biological experiments. Upon entering the post genomic era, the wet-lab oriented way is faced with challenges rising from the huge amount of data in need of rapid and systematic interpretation. As a result, nowadays Bioinformatics (also referred as computational biology) serves a critical role to analyze and interpret the data which are so huge that they cannot be handled by specific experiments alone. On the other hand, new biological data and discoveries also drive for novel models and problem formulations in Bioinformatics for insights into understanding life mechanisms, engineering biological systems and fighting against diseases. As biological data from experiments are usually noisy, rough and specific, Bioinformatics aims to bridge them to cleansed (curated), well-organized and generalized information, where patterns, knowledge, and discoveries can be further derived using computational techniques.

The central dogma in molecular biology describes that DNA (in a gene) is transcribed to RNA, and RNA is translated to make protein which mainly carries out the functionality. Despite the simple dogma, genes are not the only determining factors in real complicated biological systems. Regulation of genes also plays a crucial role in controlling the degrees of the genes activities (e.g. gene expressions which can be measured as the transcription rates of mRNA), eventually affecting the phenotypes such as functions and appearances. Therefore, deciphering the mechanisms of the gene regulation is crucial not only for the understanding of life but also for the bioengineering and therapeutic purposes.

1.1.2 Pattern Discovery in Transcriptional Regulation Based on Evolutionary Computation

Though gene regulation also happens at other levels such as post-translational regulation, transcriptional regulation is the fundamental and primary one, and will be our focus in this thesis. Transcriptional regulation is realized through interactions of certain proteins and DNA substrings from DNA sequences prior to a target gene, which are called transcription factors (TFs) and Transcriptional Factor Binding Sites (TFBSs) respectively. The analogy is the combination consisting of keys (TFs) and the control switches with keyholes (TFBSs) for a production line (gene expressions). When TFs bind to specific TFBSs, certain levels of gene expressions (transcription rates of mRNA) are observed. It is analogous to that the keys (TFs) insert into specific control switches (TFBSs) with the matching keyholes, and then control the production rates (gene expressions). However, these matchings of keys and/or keyholes have no distinguishing appearances with respect to individual residues (simply amino acids and/or nucleotides) if they are examined one by one. However, these amino acids and/or nucleotides, serving for specific regulatory purposes, magically form patterns that are not usual to happen in other non-regulatory parts of the sequences. This concept is termed "conservation" in biology, because subsequences carrying important functions (regulatory ones here) are much less likely to change (i.e. are conserved) throughout evolution. Thus subsequences related to similar functions or behaviors tend to be very similar and can be represented concisely by certain patterns. Therefore, discovering such patterns, e.g. those of the TFBSs and TF-TFBS pairs, is critical to decipher gene regulation, for further scientific (life secrets), engineering (synthetic biology) and medical (regulatory diseases like cancers) purposes.

Thanks to the technologies of sequencing and high-throughput genomic profiling, now we can readily study transcriptional regulation with the sequences potentially containing TFs/TFBSs as well as the gene expression profiles. A wide range of problems are covered such as TFBS identification (or motif discovery), expression clustering/bi-clustering (not our focus) and the inference of transcriptional regulatory networks [7, 12]. Due to the huge amount of information and data, computational methods are also essential to verify existing biological observations, narrow down only highly testable candidates for biological experiments, model available data for further predictions and discoveries, and gain insights into regulation in a systematic way.

In light of sufficient data, Evolutionary Computation (EC) has been widely applied and shown to be promising for various problems in Bioinformatics [28, 76]. EC offers a unique and under appreciated advantage to challenging, non-linear, dynamic problems in Bioinformatics, and hybridization of local operators (memetic algorithms) is possible and very useful for some problems [27]. In this thesis, we develop and apply novel EC based approaches, mainly memetic Genetic Algorithms (GAs), i.e. GAs with efficient local operators, for various pattern discovery problems, and try to reveal protein-DNA interactions in transcription regulation through discovering TF-TFBS associated patterns .

1.2 Contributions

Concentrating on pattern discovery in transcriptional regulation, we have contributed to various aspects by developing novel GA based algorithms to discover TFBS patterns (**TFBS motif discovery**) and approximate TF-TFBS associated patterns (**TF-TFBS rules**).

1.2.1 TFBS motif discovery

On the problem of TFBS motif discovery, challenging with respect to both optimization and modeling, we have developed two novel Genetic Algorithm with Local Filtering (GALF) algorithms: GALF-P (post-processing) [19] and GALF-G (generalized) [20]. More generic and complicated spaced motif discovery has also been handled by the newly developed Genetic Algorithm for Spaced Motifs Elicitation on Nucleotides (GASMEN) [17].

GALF-P [19], with focus on optimization, combines existing motif representations and introduces the memetic operator of local filtering, which effectively and efficiently improves the candidate solutions toward optimality. Post-processing with adaptive adding and removing is developed to handle general cases with arbitrary numbers of TFBS instances per sequence. GALF-P outperforms the state-of-the-art GA approach, GAME, significantly by over 20% in average F-scores and provides much more robust and consistent performance (standard deviations one order of mangnitude smaller for certain real datasets). GALF-P is also shown to be more efficient than GAME, by 4.49 times on average.

GALF-G [20], with extended focus on modeling, better captures the input uncertainty (in particular motif widths) in practice with the proposed generalized model tackling the motif width range of interest simultaneously. Moreover, a meta-convergence framework for GAs is proposed to provide multiple overlapping optimal motifs simultaneously in an effective and flexible way. GALF-G was tested extensively on over 970 synthetic, real and benchmark datasets, and is better than the state-of-the-art methods. The range model shows an increase in sensitivity compared with the single-width ones, while providing competitive precisions on the *E. coli* benchmark. Effectiveness can be maintained even using a very small population, exhibiting very competitive efficiency. In discovering multiple overlapping motifs in a real liver-specific dataset, GALF-G outperforms MEME by up to 73% in overall F-scores. GALF-G also helps to discover an additional motif which has probably not been annotated in the dataset.

GASMEN [17]: while existing algorithms mainly handle monad (contiguous) motifs, there are more generic and complicated spaced motifs with arbitrary non-conserved portions (gaps or spacers). Current methods for spaced motifs impose various constraints on gaps, which may affect the discovery of complex motifs. We develop Genetic Algorithm for Spaced Motifs Elicitation on Nucleotides (GASMEN), which searches from a wide range of possible widths (4-25) and relaxes substantial constraints of previous methods. GASMEN employs submotif indexing to partition the search space into smaller subspace for GA to easier reach optimality. Multiple-motif control is employed and probabilistic refinements are proposed to improve motif quality respectively. The preliminary results on real spaced motifs demonstrate that GASMEN is promising to find more accurate motifs and optimal widths, compared with the state-of-the-art method, SPACE. GASMEN is also capable of finding monad motifs, outperforming both Weeder and SPACE on most of the 8 real datasets, and shows the best balance of performance on the eukaryotic benchmark compared with GALF-G, MEME and Weeder.

1.2.2 TF-TFBS rules

TF-TFBS binding patterns (**TF-TFBS rules**) beyond motif discovery have also been investigated for a better understanding of transcriptional regulation.

Recent mining on exact TF-TFBS associated sequence patterns (rules) has shown great potentials and achieved very promising results. However, the exact rules cannot handle variations in real data, resulting in limited informative (verified) rules. In this chapter, we for the first time generalize the exact rules to approximate ones for both TFs and TFBSs, which are essential to handle biological variations. A progressive approach is proposed to alleviate the computational challenge. Firstly, TF-TFBS data are grouped by the available TFBS motifs from the representative TRANSFAC database with different approximation thresholds. Secondly, to target the approximate TF core motif discovery, a customized algorithm is developed with over 73% improvement over MEME. Associating the grouped TFBS consensuses and TF motifs we have the approximate TF-TFBS rules.

The rules discovered are evaluated comprehensively with Protein Data Bank (PDB) data. The approximate TF-TFBS rules exhibit a significant edge over the exact ones, with many more verified rules and up to 300% better verification ratios. 64% – 79% of the TF-TFBS rules are shown statistically significant (p-values < 0.05). Detailed analysis on PDB cases, homology modeling, and independent conservation verification on NCBI records demonstrate that the approximate rules reveal the flexible and specific protein-DNA interactions accurately. The approximate TF-TFBS rules discovered show great generalized capability of exploring more informative binding rules. Potential applications are to predict protein-DNA interactions given either side and to better decipher transcriptional regulation.

We summarize our extensive efforts and contributions to TFBS motif discovery and TF-TFBS binding pattern discovery in the Conclusion chapter, and further introduce the future work for better deciphering transcriptional regulation.

1.3 Thesis Outline

The rest of the thesis is arranged as follows. Chapter 2 gives the transcriptional regulation background in biology, with focus on TF-TFBS bindings, the general related computational background (search/optimization, Evolutionary Computation (EC)), and problem specific background for pattern discovery.

Chapters 3-6 describe our own research contributions to various TFBS motif discovery problems, and approximate TF-TFBS associated sequence pattern (rule) discovery. Chapter 3 presents GALF-P from the optimization aspect for the motif discovery problem. Chapter 4 further analyzes the problem from the modeling aspect and presents GALF-G for more general cases. Chapter 5 turns to recent generic spaced motif discovery and presents GASMEN which demonstrates outstanding performance. Chapter 6 investigates the approximate TF-TFBS associated sequence pattern discovery problem, describes our progressive approach and the promising verification results on real data.

Finally, Chapter 7 concludes the thesis and provides further discussion on future work.

□ **End of chapter.**

Chapter 2

Background

Summary

The biological and computational background related to pattern discovery for deciphering gene regulation is provided respectively in this chapter. Problem specific background is presented lastly.

2.1 Biological Background

In this section, the related biological knowledge on gene regulation will be briefly introduced in order to provide a basic understanding and it serves as a link to the applicable computational methods. Firstly, the basics of the central dogma are introduced. Secondly, transcriptional regulation involving TF and TFBS bindings is presented in greater detail, followed by the extensions to transcriptional regulatory networks. Finally, microarrays measuring gene expressions are mentioned.

2.1.1 The Biology Basics and the Central Dogma

DNA (deoxyribonucleic acid) consists of two strands, and each strand is made up of phosphates, deoxyribose sugars and nucleotides (including adenine "A", guanine "G", cytosine "C" and thymine "T") linked in series. The two strands are complementary (A paring with T, C pairing with G) so one strand can determine and thus represent the other. Each strand has a direction from the 5' end to the 3' end, and the complement is in a opposite direction, and that is why the two strands are called reverse complements. For simplicity, DNA sequences are often represented by the strings (from one strand) generated from the symbol set of nucleotides (called the alphabet) $\Sigma = \{A, C, G, T\}$. DNA contains the full genetic blueprint for the cell and for all other cells in the organism in the case of multicellular eukaryotes [28]. Thus analysis on DNA sequences can reveal the most important information of life.

A gene is a segment of DNA that contains the information necessary to produce a functional product, usually a protein. However, the information is not directly passed from a gene to the corresponding protein, and it needs RNA (in particular messenger RNA mRNA) to be an intermediate template for transfer, and the process is called transcription. When and which parts of a gene is transcribed is controlled by the process called transcriptional regulation, which will be introduced in the next subsection. RNA (Ribonucleic acid) can be represented by the symbols {A (adenine), G (guanine), C (cytosine), U (uracil)}, where U is the replacement for thymine (T) in DNA. The messenger RNA (mRNA) serves the template of DNA to carry the "encoded" information to make specific protein.

Protein is the final product of DNA after translation from the RNA template. It can be denoted by the sequence of amino acids which are defined by the genes and encoded in the genetic codes (three-letter codons translated from RNA). Protein carries out particular functions in the cell. One important function of protein we focus on is the regulatory function that controls the gene expressions (transcription rates), and such protein is called

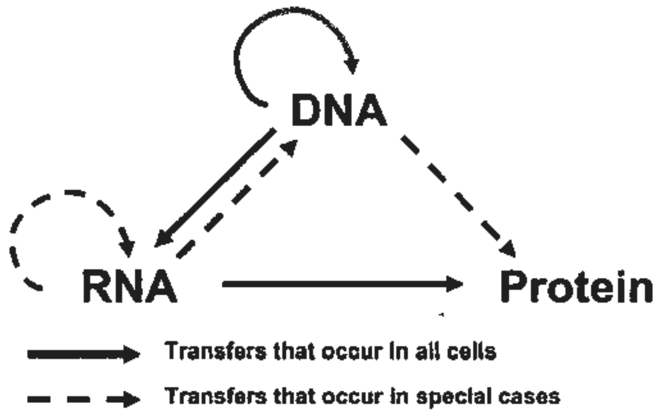


Figure 2.1: Central dogma. Only the general case (solid arrows) of transfers of information is discussed here. DNA $\rightarrow$ DNA is replication which is not discussed. DNA $\rightarrow$ RNA is transcription, and RNA $\rightarrow$ protein is translation.

Transcription Factor (TF).

The central dogma can be simplified for computer scientists as the flow of information from genes to their functional products in cells: DNA (of genes) generates the template in the form of a strand of RNA via transcription, and RNA in turn codes for protein in translation [28], which carries out some function in the cell. The relation between transcription and translation is illustrated in Figure 2.1.

2.1.2 Transcriptional Regulation with TF-TFBS Bindings

Despite the previous simple descriptions, in fact transcription is a series of complicated events and leads from DNA to messenger RNA (mRNA) with different degrees of the gene activities or expressions (transcription rates). For a given gene on the DNA sequence, the gene is also called a coding region and there is a regulatory region (non-coding region) prior to it (called upstream to it). The coding region is responsible for the transcription into mRNA which is finally translated into protein, as mentioned before. On the other hand, the regulatory region contributes to the control information of the gene's expressions.

In particular, the regulatory region contains one or more Transcription Factor Binding Sites (TFBSs), which are nothing other than some short DNA subsequences (usually 5-20 bp). They are special in that these DNA subsequences can form (hydrogen) bonds with specific regulatory proteins called Transcription Factors (TFs), as if they are recognized and bound by the TFs. The TFs will bind other regulatory proteins called co-factors, and finally a special protein (enzyme) called RNA polymerase is recruited to bind and initialize the transcription process. These TF-TFBS bindings, as the major protein-DNA interactions, have the effect of regulating the transcription rates. TF-TFBS bindings may act positively or negatively, and lead to the increase (enhancers) and decrease (suppressors) of expressions [14]. The regulatory regions are typically short in prokaryotes and have a small number of binding sites, while they may be very long in eukaryotes and contains sites for multiple TFs.

Simply speaking, transcriptional regulation describes the information flow from the regulator(s) such as the TF(s) to the regulated gene(s). This process reveals the mechanisms of transcriptional regulation of genes, but they are not fully understood yet. A simplified illustrative example with only one TF binding one TFBS is shown in Figure 2.2.

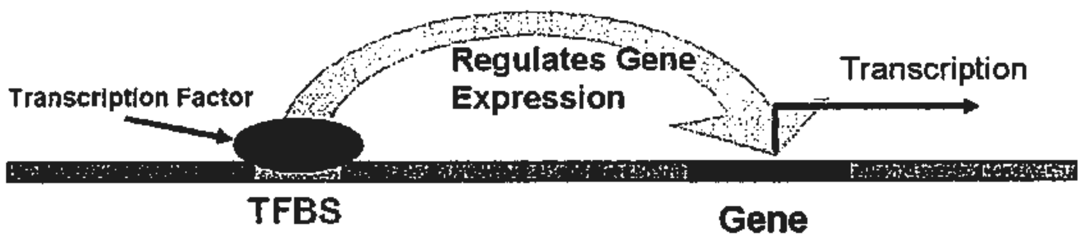


Figure 2.2: A simplified example of transcriptional regulation with one TF binding the TFBS.

As mentioned in the Introduction chapter, the analogy is the scenario consisting of keys (TFs) and the control switches with keyholes (TFBSs) for a production line (gene expressions). When TFs bind to specific TFBSs, certain levels of gene expressions (transcription rates of mRNA) are observed. It is analogous to that the keys (TFs) insert into specific control switches (TFBSs) with the matching keyholes and control the production rates (gene expressions). However, these matchings of keys and/or keyholes have no distinguishing appearances but simply amino acids and/or nucleotides if they are examined one by one. However, these amino acids and/or nucleotides, serving for specific regulatory purposes, magically form conserved patterns that are not usual to happen in other non-regulatory parts of the sequences. Thus discovering such patterns, e.g. those of the TFBSs and TF-TFBS pairs, is critical to decipher gene regulation, for further scientific (life secrets), engineering (synthetic biology) and medical (regulatory diseases like cancers) purposes.

Other regulatory mechanisms at different levels exist, such as post-translational modification of factors, specific interactions with co-activators, thermodynamics of protein-protein and protein-DNA interactions [12]. This thesis will not go into the details and will focus on transcriptional regulation.

2.1.3 Gene Expression Microarrays

In order to begin the research on gene regulation, data representing their behaviors or interactions must be first obtained. Thanks to the new technologies of high-throughput genomic profiling approaches developed over the last few years, large amount of DNA gene expression data can be obtained from microarrays. Such DNA gene expression microarrays allow biologists to study genome-wide patterns of gene expression in any given cell type, at any given time, and under any given set of conditions [7, 12]. In these arrays, total RNA is reverse-transcribed to create either radioactive- or fluorescent-labeled cDNA that is hybridized with a large DNA library of gene fragments attached to a glass or membrane support. Imaging techniques are used to produce expression measurements for thousands of genes under various experimental conditions.

Application of these arrays is producing large amounts of data, potentially capable of providing fundamental insights into biological processes ranging from gene function to development, cancer, and aging, etc. These data are the essential information source for deciphering gene regulation.

2.1.4 Transcriptional Regulatory Networks

In real biological systems, there are more complicated interactions than Figure 2.2, involving various TFs, TFBSs and regulated genes since TFs themselves, as proteins, are also products of genes. In some particular scale, the related genes sharing regulatory TFs can be grouped and examined as a unit called a network motif, or a module, to describe the regulatory interactions. Common cases of the transcriptional regulatory modules are shown in Figure 2.3. For example, the feed-forward loop case depicts that one TF regulates the expression of a second gene and thus its TF, and both factors together regulate the expression of a third gene [12]. Such modules to some extend are useful for understanding the details of transcriptional regulation and distinguishing the different types of complication, such as the simple and explicit response of auto-regulation versus the subtle and gradual response for multiple inputs.

However, though modules provide great detail for small portions of the network, in fact they are not totally autonomous and may interact with each other, as a result forming a huge network with hundreds to thousands of genes. A simple example can be a mixture of the modules listed in Figure 2.3, where the TF involved in a feed-forward loop may also participate as

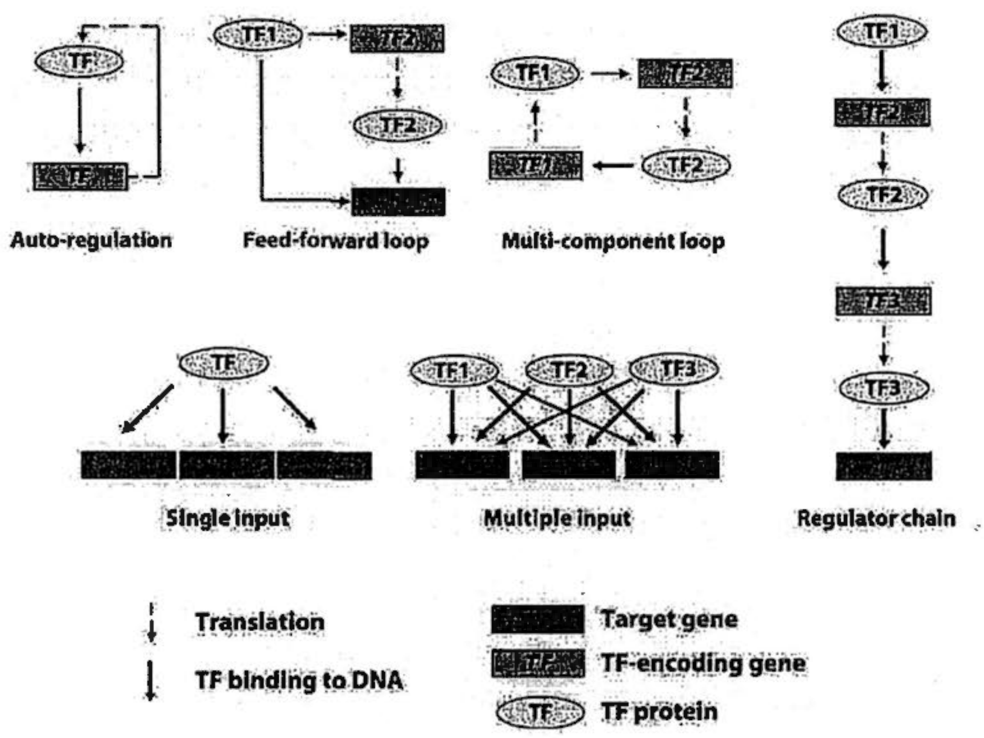


Figure 2.3: Transcriptional regulatory network motifs adopted from [12]

one of the multiple inputs for another genes or even a node in another regulator chain. So the ultimate goal for deciphering the gene regulation is to model all the interacting genes in cell with a whole network, describing their full causality relation and hopefully the dynamics as well at a system level. This is one most challenging goals and is likely to remain as a central topic of Bioinformatics for long.

To model the gene-gene interactions either qualitatively or quantitatively, gene expressions have to be known as the premise. The transcription rates of the genes can be measured by the microarray technology in a high throughput manner (genomic scale profiling) to represent the gene expressions. Microarrays allow biologists to study genome-wide patterns of gene expression in any given cell type, at any given time, and under any given set of conditions [6]. In these arrays, total RNA is reverse-transcribed to create either radioactive- or fluorescent-labeled cDNA that is hybridized with a large DNA library of gene fragments attached to a glass or membrane support. Imaging techniques are used to produce expression measurements for thousands of genes under various experimental conditions. Application of these arrays is producing large amounts of data, potentially capable of providing fundamental insights into biological processes ranging from gene function to development, cancer, and aging, etc. These data are the essential information source for deciphering gene regulation.

2.2 Computational Background

Because DNA and protein subsequences carrying important functions are less likely to change during evolution and across different species, they are "conserved" and form certain patterns. These patterns exhibit high similarities (called conservation) and such similarities are not likely to happen by chance from the background sequences. Widely available data and annotations enable computational methods to be applied to discover these patterns. The discovered patterns serve as testable candidates with high potentials for experimental verifications to reduce time and costs, and are promising for new biological knowledge discoveries.

Bioinformatics problems, including pattern discovery focused in this thesis, have common features such as the the huge amount of noisy data and requirement for search/optimization methods in huge search space. The related computation background thus is introduced. As our research is mainly based on Evolutionary Computation (EC), it will be elaborated in greater detail.

2.2.1 Heuristic Methods for Search/Optimization

Optimization refers to choosing the best element(s) from some set of available alternatives, for example, choosing the multi-dimension point(s) to maximize or minimize a multi-dimension real function. The process of choosing from the available alternatives can be referred as search. More generally, search in computer science is to find an element (or elements) with specified properties among a collection of elements (available alternatives).

Many problems in Bioinformatics require searching an exponentially growing space with respect to the problem size (NP-hard problems), such as TFBS motif discovery [53]. Moreover, the problem sizes are usually large according to the large amount of data available. As a result, some compromise has to be accepted for an algorithm to find a feasible solution in reasonable time, which is called a heuristic method.

Definition 1 *A* ***heuristic*** *is a technique designed to solve a problem that ignores whether the solution can be proven to be correct, but which usually produces a good solution or solves a simpler problem that contains or intersects with the solution of the more complex problem.*

Put another way, heuristics reflect knowledge about the domain that helps guide the search and reasoning in the domain. Following are some general heuristic methods for search or optimization:

Hill Climbing

The approach looks at all operations and choose the one leading to a better state closest to the goal. The process repeats until no improvement can be obtained for certain situation. Hill climbing assumes that local improvement will lead to global improvement, which, however, is usually not the case the Bioinformatics problems. The problems with hill climbing are obvious: local optima such as local maxima – there exist another peak other than the one reached, plateau – the values around are as good as each other and it does not know where to go, and ridges – on a ridge leading up when an operation cannot be directly applied to improve the situation. Many Bioinformatics problems such as motif discovery problems are critical to find the global or near-global optima.

Simulated Annealing

Simulated Annealing (SA) is inspired by the physical process of annealing metals to solid minimal-energy states. It can be treated as a stochastic variation on hill climbing in which downhill moves can be made. The search mainly moves uphill except occasionally with low probability it moves uphill instead. The probability of making a downhill move decreases with time (or steps, analogous to temperatures) so the length of the exploration path from a start state. The problems of SA include choosing an initial temperature and the elaborate annealing schedule (the rate at which the system cools) varying from problem to problem.

Evolutionary Computation

Evolutionary Computation (EC) is the family of multi-point global search approaches inspired by Darwin's theory of natural selection and evolution. It will be detailed in the following subsection.

2.2.2 Evolutionary Computation

Evolutionary Computation (EC), or the Evolutionary Algorithm (EA), is a family of heuristic optimization algorithms inspired by Darwin's theory of natural selection and evolution. Broadly speaking, EC approaches all use a population of competing solutions subjected to random variation and selection to achieve certain purpose. These solutions are called individuals and they form a population. The fitness of each individual reflects its worth in relation to the desired goal. The population is subject to selection and variations in different generations, yielding some offspring and each individual competes for survival.

There are numerous different techniques in terms of representations, genetic operators and population dynamics and meta-level evolutionary techniques such as self-adaptation. There are four representative members of EC and they are genetic algorithms (GAs), evolution strategies (ES), evolutionary programming (EP) and genetic programming (GP).

Components of EC include the representation, which is the definition of individuals, the evaluation function which is usually called fitness function, a population to maintain, selection mechanism for parents, variation operators such as recombination (crossover) and mutation, and the survivor selection mechanism which is also known as replacement [25]. The main procedure of EC is shown in Figure 2.4.

Genetic Algorithms

Genetic algorithms (GAs) are the most representative and widely used EAs. GAs typically use fixed length strings to represent individuals. In early work, the strings are typically binary ones, but nowadays different representations such as integers, real numbers, as well as problem specific representations are also commonly used. Selection is probabilistic and usually propor-

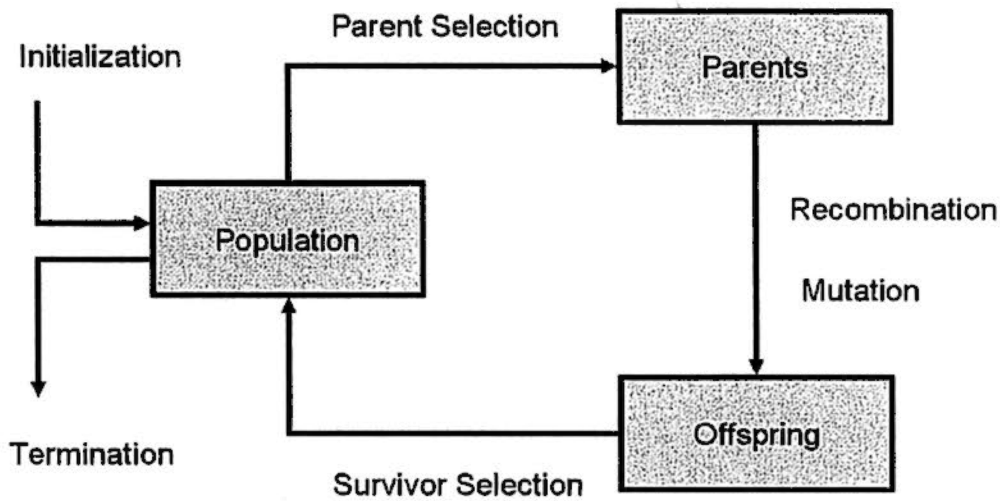


Figure 2.4: The general scheme of an evolutionary algorithm. Modified from [25]

tionate to fitness. There is some generation gap for the offspring to replace their parents. Variation operators include mutations and crossovers.

The working mechanisms of the genetic algorithm (GA) are briefly introduced (GALF [19] for motif discovery shown in brackets, which can be referred in the next chapter) as follows. A GA (e.g. GALF) maintains a population of candidate solutions, called individuals (e.g. a set of TFBS instances represented by their positions $A = \{p_1, p_2, ...\}$ in the sequences), and performs optimization or search (maximize the fitness f) iteratively. In each iteration named a generation, part of the individuals are chosen by parent selection, and generate offspring (new individuals) via genetic operators such as mutation and crossover (randomly changing a TFBS position (p) and mixing two set of TFBSs (A_i, A_j) respectively in GALF). The resulting population is subject to survivor selection based on fitness f (crowding [66] is used in GALF, i.e. keeping the fitter individuals from the pairs of similar parents and offspring), where unfit individuals will be eliminated to maintain a constant population size. The fittest surviving individual(s), towards convergence, e.g. unchanged for a long period, or at the end of all generations, will be output as the final solution(s). The balance between convergence (exploitation) and divergence (exploration) needs to be maintained by various general and problem specific operators for good performance.

Other related methods such as memetic approaches are also widely used with hybridization of EC approaches and local search techniques [25]. A memetic operator is the local operator (such hill climbing and expectation maximization) incorporated in an EC approach, and it is able to improve the effectiveness and efficiency considerably. The specific EC approaches for the particular problems related to transcriptional regulation will be reviewed in details in the problem specific background.

2.3 Problem Specific Background

In this section we will review the background with the specific pattern discovery problems for transcriptional regulation. The reviews combine both biological and computational points of views for the specific problems.

2.3.1 TFBS Motif Discovery

Since TFBSs are a critical component in gene regulation, identification of TFBS patterns (TFBS motif discovery) is a central problem for understanding gene regulation in molecular biology. TF motif discovery is also important to annotate new TFs for their binding domains, but it has been quite successful [5] while TFBS motif discovery is still very challenging [87, 99]. So we will focus on TFBS motif discovery extensively.

Problem Descriptions

The DNA binding domain(s) of a TF can recognize and bind to a collections of similar TFBSs in a sequence-specific manner, from which a conserved pattern called motif can be obtained. Based on this phenomenon, *de novo* motif discovery using computational methods have been proposed to identify and predict TFBSs and their corresponding motifs. Motif discovery provides significant insights into the understanding of the mechanisms of gene regulation. It serves as an attractive alternative for providing pre-screening and prediction of unknown TFBS motifs to the expensive and laborious biological experiments such as DNA footprinting [30] and gel electrophoresis [31]. The recent technology of Chromatin immunoprecipitation (ChIP) [65, 93] measures the binding of a particular TF to DNA using microarray technology at low resolution in a high-throughput manner, and produces more reliable input data of co-regulated genes for motif discovery [57].

For bioinformatics, motif discovery data can be retrieved by collecting the DNA sequences (hundreds to thousands in length) of regulatory regions of co-regulated genes that are considered bound by the same of similar TFs, because they should contain the conserved patterns/motifs of the similar TFBSs. The regulatory regions are fully available for many organisms with their full genomes sequenced already. The collecting criteria for co-regulated genes can be based on gene annotations which are widely available [89], or gene expression clustering from microarray data [6].

Though there are many variations of problem formulations for TFBS motif discovery as detailed in the upcoming chapters, the problem is generally formulated as follows:

Input: a set of N sequences $S = \{S_1, S_2, ..., S_N\}$, each of which is from the finite alphabet Σ ($= \{A, T, C, G\}$ for DNA sequences), where the length of each sequence is l, and the motif width w with a valid constraint $0 < w \ll l$. S is assumed to be a set of DNA sequences from regulatory regions of the genes bound by the same or similar TFs.

These genes can be obtained based on existing annotations of similar functionality (which are usually vailable) or the similar co-regulation patterns of the genes, i.e. similar expression patterns of microarrays (where there are abundant clustering tools for the task). The same l is set for each sequence with the purpose of analysis simplicity without lose of generality, and in real cases we can choose the minimal length l_{min} of S as l.

Definition 2 ***Canonical Motif Discovery:*** *Given the input N sequences S and w, find a set of instances $M = \{m_1, m_2, ..., m_N\}$ where each m_i is a subsequence with length w from sequence S_i, and they form certain pattern (motif) P (called instance/position-led), or vice versa (find pattern P and then M, called consensus-led), such that certain scoring function f, applied on M, or P, or M, P together, is maximized (or certain loss function d minimized).*

It is also called *de novo* motif discovery because no specific knowledge about the motif P is known beforehand, otherwise it is termed as motif matching [45] which is considered easier and better handled already. The canonical Definition 2 is proved NP-hard even with the most simplified assumptions [53]. Moreover, there are considerable variations that complex the canonical motif discovery definition. For example, there are different choices of the pattern representations, more descriptive being more difficult to search; w may be unknown and only a range of possible widths $[w_{min}, w_{max}]$ is known; it is not necessary one occurrence per sequence (OOPS), i.e. one m_i for sequence S_i), and zero or OOPS (ZOOPS) as well as any number of occurrences per sequence (ANOPS) can happen; and more than one motif are expected to be discovered. These complications are addressed in this thesis with novel GA based algorithms presented in the following chapters.

de novo motif discovery can be summarized by the following major components:

1. Motif Representation: the profile describing motif characteristics (e.g. the consensus), usually at a certain width w, including the motif occurrences or the retrieval method (e.g. all substrings within certain hamming distance from the consensus).

2. Evaluation Function: the quantitative criterion to rank and choose the potential optimal motifs from candidate motifs.

3. Search or Optimization: effective and efficient strategies to pick out the optimal motifs from the input sequences, according to the evaluation function.

Existing methods with categorization are reviewed below.

Categorization

Because the conservation of motifs is often degenerated due to TFBS mutations, the searching is difficult (NP-hard [53]). Extensive algorithms have been proposed for *de novo* motif discovery since the last decades. There are two major representations for TFBS motifs (conserved patterns): (i) Consensus Representation and (ii) Matrix Representation; and there are two main different searching paradigms: (a) Enumeration Methods and (b) Stochastic Searching [65]. They are briefly described as follows:

(i) Consensus Representation is based on discrete strings. A simple model is to minimize the mismatches between the consensus and the TFBS instances [10, 55, 77, 85].

(ii) Matrix Representation is usually a Position Frequency Matrix (PFM; see Table 2.1), or a Position Weight Matrix (PWM),

Sequences S	SIM A	TFBSs R	PFM Θ $(4 \times w(=7))$
S_1: acgtCGATTGCctaag	0000100000000000	CGATTGC	
S_2: taTGATCGAtgacgca	0010000000000000	TGATCGA	A: 0.0 0.2 0.6 0.1 0.1 0.0 0.7
S_3: cgaCAATTGAgcttac	0001000000000000	CAATTGA	C: 0.8 0.0 0.2 0.3 0.3 0.2 0.3
S_4: gCGCTCGAcaagctgt	0100000000000000	CGCTCGA	G: 0.0 0.8 0.0 0.0 0.0 0.8 0.0
S_5: cgttTGTCACAgtcta	0000100000000000	TGTCACA	T: 0.2 0.0 0.2 0.6 0.6 0.0 0.0
S_6: tcagcCACACCCagct	0000010000000000	CACACCC	
S_7: ccagagCGTCTGAttg	0000001000000000	CGTCTGA	Background: Θ_0:
S_8: gacttcaCGACTGAgc	0000000100000000	CGACTGA	$\theta_{0A} = 0.24$ $\theta_{0C} = 0.29$
S_9: gctgcccatCGATTGA	0000000001000000	CGATTGA	$\theta_{0G} = 0.24$ $\theta_{0T} = 0.23$
S_{10}: ccaggtacCGATTGCa	0000000010000000	CGATTGC	

Table 2.1: An artificial example of motif discovery. It shows the sequences S, the SIM A, the motif instances R, the PFM Θ and the background frequencies Θ_0. In sequences S, the nucleotides from the background are shown in lower case, while the nucleotides from the motif instances in upper case.

to show the quantitative frequencies or weights of nucleotides in the motif. Representative evaluations for a motif matrix include Information Content (IC) [96], maximum a posterior (MAP) [5] and the Bayesian models [41] (see the probabilistic models in Methods section).

The searching techniques with respect to the two representations, are discussed below.

(a) Enumeration Methods are usually applied [10, 78–80, 85] to the consensus representation, but they do not scale up for long widths. However, they are useful to provide candidates for further searching and evaluations [15, 57, 82]. Weeder [78, 79] is one well-known representative in this category.

(b)Stochastic Searching is usually applied to align TF-BSs and obtain the motif matrix for the matrix representation. Typical techniques can be categorized into **local searching** [5, 57] and **global searching**, where the latter can be classified into **(S) Single-point** and **(M) Multi-point or group-based searching**. Global searching is more likely to find the global optima compared with local searching. While Gibbs sampling is popular in motif discovery tools: e.g. BioProspector [56], AlignACE [84] and MotifSampler [98]). Its single-point nature requires numerous iterations to converge to the

Representations (i) Consensus (ii) Matrix and Evaluations		(a) Enumerations		(b) Stochastic Search		
					Global	
		Exhaustive	Non-exhaustive	Local	Single-point (Gibbs Sampling)	Multi-point (GAs)
(i)	Hamming	[10,85]	[80]	[15,82]		[55,77]
	Z-score		Weeder [78,79]			[21]
(ii)	IC		[97]		[51]	[77], GALF-P [19]
	Bayesian		BioOptimizer [40]		BioProspector [56] Motif Sampler [98]	GAME [101]
	MAP			MEME [5] MDScan [57]	AlignACE [84]	

Table 2.2: Summary of the representative motif discovery methods. The methods included in our comparison experiments are shown with their names. IC stands for Information Content.

global optima, otherwise the performance may be affected significantly. Alternatively, the multi-point global searching approach, GAs [33,36], has shown promising results in motif discovery [19,21,29,55,61,77,101]. There is great potential for them to be applied to more sophisticated models and provide multiple optimal motifs [61]. GAs are more effective than locally incremental and single-point search methods because GAs perform global search while maintaining a population of different solutions concurrently. Advantages of GAs compared with the conventional motif discovery methods [59] include the global search capability, which is more likely to locate global optima, the flexibility of representation and scoring, and good scaling property.

Table 2.2 summarizes the representations, the associated models and the searching techniques employed by the motif discovery methods. The table serves to show the representative methods in each category including those we have compared in our experiments.

Other Methods

Methods out of the scope of *de novo* motif discovery but worth introducing are briefly mentioned as follows:

Ensembles of multiple motif discovery programs have been recently shown to improve their performance [38, 65, 104]. However, modelling TFBS motifs is critically beneficial for better understanding and predicting novel motifs, and provides essential performance improvement for ensembles. As a result, we will focus on individual motif discovery methods in this thesis.

Incorporating additional information sources [24, 91] is another trend to improve the motif prediction accuracy. While extra requirements are needed for their success, the sequence-based motif discovery problem remains challenging [37, 87, 99] and calls for our serious attention because generalization and improvement on the sequence-based methods will without doubt help the integrated approaches.

2.3.2 TF-TFBS Associated Patterns

Protein-DNA interactions play a central role in genetic activities [62, 64]. The bindings of transcription factors (TFs) and transcription factor binding sites (TFBSs) are fundamental protein-DNA interactions in transcriptional regulation. Therefore, besides motif discovery on TFs or TFBSs, it is also important to directly identify TF-TFBS binding rules to understand protein-DNA interactions and further decipher gene regulation.

TF-TFBS Data

It is both expensive and time-consuming to identify accurate TF-TFBS binding sequence pairs experimentally either using the traditional DNA footprinting [30], gel electrophoresis [31], or recent Chromatin immunoprecipitation (ChIP) technology [65, 93]. TRANSFAC [72] is one of the largest and most representative databases for such regulatory elements including TFs, TFBSs, nucleotide distribution matrices of the TFBSs (TFBS motifs), and regulated genes. The data are annotated and curated from peer-reviewed and experimentally proved publications. Other annotation databases of TF families and binding domains are also available (e.g. PROSITE [39], Pfam [8]).

It is even more difficult and laborious to extract high-resolution 3D protein-DNA interaction (TF-TFBS binding) structures with X-ray crystallography or Nuclear Magnetic Resonance (NMR) spectroscopic analysis. Nevertheless, the high-quality verified structures serve as valuable verification sources for putative binding discoveries. The Protein Data Bank (PDB) [9] is the most representative repository with high resolution structures at atomic levels. However, the available 3D structures are far from complete. As a result, there is strong motivation to have automatic methods, particularly, computational approaches based on other available data, to provide testable candidates of novel TF domains and/or TFBS motifs with high confidence to aid and accelerate the wet-lab experiments.

Existing Methods

The first attempt of Bioinformatics methods to decipher TF-TFBS bindings was TF/TFBS **motif discovery**. Additionally, researchers have been trying hard for the protein-DNA **one-to-one binding codes**. Data mining methods have also been proposed, and recent work on mining exact TF-TFBS associated sequence patterns shows promising results. They are briefly reviewed as follows:

Motif discovery: as reviewed previously, amino acids from TF domains and TFBSs sequences are conserved according to their important functional similarities. By exploiting conservation in the sequences, computational methods called motif discovery has achieved certain success in discovering TF or TFBS motifs. Motifs are usually represented as the consensus strings [53] or position weight matrices (PWMs) of the residue distributions [96]. *de novo* motif discovery [65] identifies the conserved patterns without knowing their motifs beforehand, based on certain motif models and scoring functions [5,41,96] from a set of protein sequences/DNA promoter sequences with similar regulatory functions. A significant limitation of motif discovery to model TF-TFBS binding is the lack of linkage between the binding counterparts and thus cannot directly reveal TF-TFBS relationships.

One-to-one binding codes: Numerous studies have been carried out to analyze existing protein-DNA interaction structures comprehensively [44,62–64] or with focus on specific families (e.g. zinc fingers [48]). Various properties have been discovered concerning, e.g., bonding and force types, TF conservation and mutation [64], and bending of the DNA [44]. Some are already applicable to predict binding amino acids on the TF side, e.g. [43]. Alternatively, researchers have sought hard for general binding "codes" between proteins and DNA, in particular the one-to-one mapping between amino acids from TFs and nucleotides from TFBSs. Despite many proposed one-one binding propensity mappings [64,68,69], it has come to a consensus that there are no simple binding "codes" between single amino acids and nucleotides [88].

Data mining: Supervised learning approaches have also been proposed [107] to mine protein-DNA interactions. Derived or transformed information is usually employed such as base compositions, structures, thermodynamic properties [1,2] as well as expressions [81]. However, due to the stringent data requirement, many training based data mining methods concentrate only on specific families or particular datasets, where the generality of the results are limited. Furthermore, these methods usually extract complicated features are not trivial to interpret, such as neural networks, support vector machines (SVM) [75] and regressions [107], and thus are less applicable for general predictions.

2.3.3 TF-TFBS Associated Pattern Discovery

Different from complicated transformed features, sequences serve as the most handy and abundant primary data, and show great potentials to reveal protein-DNA interaction relationships [88]. Thus, it is desirable to mine or discover core binding patterns of both the TF and the TFBS based on the sequence information, since a huge amount of TF-TFBS binding sequences are widely available from existing large-scale regulatory element databases [72, 86].

The problem formulation is again based on "conservation", namely the binding cores of both TFs and TFBSs should be both conserved (associated), such that these associated TF-TFBS sequence patterns appear more frequently, preferably with statistical significance, than other randomly combined subsequences from the background. In particular, we would like to discover the short (about 4-6 nucleotides or amino acids) TF-TFBS associated patterns (called rules) based on their co-occurring frequency or certain motif models, such that these rules are true in real biological interactions of TF-TFBS bindings, i.e. experimentally verified 3D structures at high resolution [83]. This is a challenging problem because the given evidence is limited on sequence data with hundreds of TF sequences (hundreds of amino acids in length) as well as TFBS sequences (tens of nucleotides in length), the desired patterns are weak and short signals (4-6 in length on both sides), and they have to truly reflect the intricate biological properties of TF-TFBS bindings (protein-DNA interactions) at high resolution. What makes us delighted is that the following recent work and the later chapter do show the TF-TFBS rule discovery is very promising to achieve the target.

A recent association rule mining approach [52] exploits the exact TF-TFBS associated sequence patterns from TRANSFAC, and discovers informative rules verified on both literatures and

PDB structures. The study, however, is limited only on exact TF-TFBS associated sequence patterns, while variations such as mutations and noises are common in real biological data. As a result, the approach only generates a handful of exact rules [52], while there are still great potentials for many more flexible and verifiable rules to be discovered.

□ **End of chapter.**

Chapter 3

TFBS Motif Discovery with GALF-P: The Optimization Aspect

Summary

GALF-P is presented with the concentration on the search/optimization aspect of TFBS motif discovery. The problem formulation thus follows the existing one in order to compare different methods, especially GAs, clearly on the search/optimization performance.

3.1 Introduction

In this chapter, we will in general follow the canonical motif discovery definition in order to focus on the optimization aspect. As surveyed in the Background chapter, GAs are shown to be promising for TFBS motif discovery. The current GA methods employ either position-led or consensus-led representations respectively, while each type has its own disadvantages. In this chapter, Genetic Algorithm with Local Filtering (GALF) (see [18] for the preliminary version) is proposed employing the combined representation and a novel local filtering operator to achieve better effectiveness and efficiency. GALF-P, the extension of GALF with adaptive post-processing, is developed to handle more general cases and shows superior performance to the current state-of-arts approaches.

The rest of this chapter is arranged as follows: in Section 3.2, the problem details will be described. In Section 3.3, GALF and GALF-P will be presented in detail. Experimental results will be reported in Section 3.4, showing the superior and reliable performance of GALF-P. The last section will be the discussion and conclusion.

3.2 Problem Formulation

3.2.1 Definitions

Generally, the single TFBS identification problem in unaligned DNA sequences can be defined as two related motif discovery problems corresponding to the position-led and the consensus-led representations respectively in GAs as follows:

Input: a set of N sequences $S = \{S_1, S_2, ..., S_N\}$, each of which is from the finite alphabet Σ ($= \{A, T, C, G\}$ for DNA sequences), where the length of each sequence is l, and the motif width w with a valid constraint $0 < w \ll l$.

Definition 3 ***General Consensus Patterns*** *(position-led): find a set of instances $M = \{m_1, m_2, ..., m_N\}$ where each m_i is a subsequence with length w from sequence S_i, such that the sum of information content IC (proposed by [96])*

$$IC = \sum_{j=1}^{w} \sum_{b} f_b(j) \log \frac{f_b(j)}{p_b} \tag{3.1}$$

is maximized, where $f_b(j)$ is the normalized frequency of nucleotide $b \in \Sigma$ on the column j of all instances in M and p_b is *the background frequency of the same nucleotide (from S or the whole genome).*

Definition 4 ***Consensus Patterns*** *(consensus-led): find a string S_C with length w (which may not exist in S), and a set of subsequences $M = \{m_1, m_2, ..., m_N\}$ from S where each m_i is with length w from sequence S_i, such that the sum of Hamming distances (d_H) is minimized*

$$\sum_{i=1}^{N} d_H(S_C, m_i) \tag{3.2}$$

The equivalent definitions of these two problems were given by [53] who have proved both of them to be NP-hard. Definitions 3 and 4 only address a special case of the real single TFBS identification problem (fixed motif width w, one occurrence per sequence (OOPS)), where there can be zero or more than one motif instance according to the motif type in each sequence. This issue will be considered in the following section. There may also be multiple TFBSs corresponding to different types of motifs or consensuses. However, in this chapter single TFBS identification will be our major concern if not specifically stated. Further extensions on modeling will be addressed in the next chapter.

3.2.2 Solution Space

To analyze the search strategies in GAs, the solution/search space of TFBS identification is discussed here.

For the solution representation in Definition 3 (position-led), different assumptions will lead to different number of instances k_i in S_i according to the previous descriptions. For the most general case where $0 \leq k_i \leq l - w + 1$, the solution space is as prohibitively huge as $O((2^{l-w+2})^N)$ [101]. For the case in Definition 3, where $k_i = 1$, the search space is reduced to be $O((l - w + 1)^N)$. While allowing $k_i \leq 1$, search space becomes $O((l-w+2)^N)$. To make the computation tractable, all the GA approaches are only restricted to the solution space for $k_i = 1$ or $k_i \leq 1$. We will start with the case of $k_i = 1$ which is uniform and widely adopted in GA methods [21, 55, 77]. Then more general cases will be addressed by post-processing in later sections.

For the solution representation in Definition 4 (consensus-led), the solution space for all possible consensus strings is 4^w, which is independent of S and M. This representation is less expressive than the one of Definition 3 because the consensus string cannot accurately measure the conservation of nucleotides when they are not fully conserved in the motif.

3.3 Methods

The overall framework, namely GALF-P, which consists of the novel Genetic Algorithm with Local Filtering (GALF) and adaptive post-processing techniques (-P), is briefly introduced in Table 3.1. The details of the framework will be presented in the following subsections.

3.3.1 GA Representations

According to the two previous problem definitions, GA approaches for TFBS identification are categorized into two based on the position-led and consensus-led representations respectively.

For the position-led representation approaches [21, 101], each individual is represented by a vector $I = \{p_1, p_2, ..., p_N\}$ storing the set of possible starting positions for the TFBS instances in each sequence. I represents a possible solution set $M = \{m_1, m_2, ..., m_N\}$ in Definition 3, because each p_i is uniquely

Table 3.1: The framework of GALF-P. MAXGEN and MAXRUN are the maximal generations of GALF and maximal times to run GALF, respectively

```
i ← 0;
Repeat: i ← i+1; // GALF
  Initialization:
    g ← 0;
    Generate a random population;
  Repeat: g ← g+1
    Random pairing of the population;
    Single-point Crossover;
    Single-point Mutation;
    Local Filtering triggered every 10 generations;
    Shift Operator on the best individual
    when it stagnates for 10 generations;
    Replacement within the pairs;
  Until (g ≥ MAXGEN or Converged)
  Store the so-far-best individual I_Best;
Until (i ≥ MAXRUN)
// Post-processing:
  I_Best+ ← I_Best with adaptive adding;
  I_Best-(Output) ← I_Best+ with adaptive removing;
```

mapped to instance m_i with w known. Position-led approaches have more flexibility to move around in the search space because it is free to change any starting position p_i with one random operation, and it is easy to simultaneously change all the positions in an individual. However, the representation cannot provide a detailed view of quality for each TFBS instance because they are evaluated as a whole, and thus cannot distinguish a small portion of unsuitable positions easily.

For the consensus-led representation approaches [55, 77, 94], each individual is encoded as the potential consensus in a string pattern $C = c_1c_2...c_w$, which has the same format as S_C in Definition 4. The individuals of consensus-led methods can be generated or extracted randomly from the input sequences. One disadvantage of consensus-led approaches is the computation need to scan all sequences when evaluating a single individual. Furthermore, string patterns are not expressive or accurate enough when different nucleotides of the instances are weakly conserved at some columns of the motif instances.

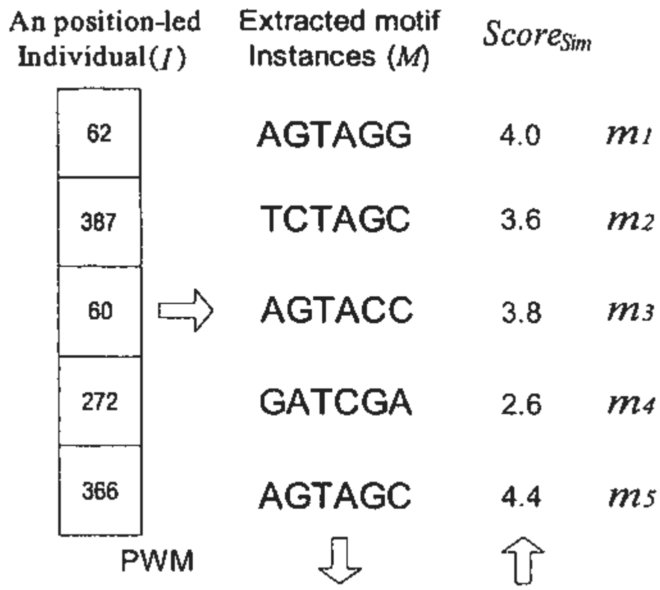
3.3.2 Representations in GALF

Although the two GA representations address TFBS identification differently, they are closely related to each other. For position-led representation, once the optimal I (in other words M) is found, C (S_C) can be easily determined by setting the most frequent letter at the ith column of M as c_i. On the other hand, once the optimal C (S_C) is discovered, the instance set M and I are determined at the same time.

Intuitively it is possible to improve both effectiveness and efficiency by combining the two representations and letting them complement each other with direct refinement on one representation based on the other one. As a result, the position- and consensus-led Genetic Algorithm with Local Filtering (GALF) is proposed. In GALF, the basic representation is based on the position-led one (I) for its flexibility to explore the search space easily. The evaluation function is the information content IC shown in Equation 3.3, which is similar to Equation 3.1 except that it only considers the non-zero frequencies.

$$IC = \sum_{j=1}^{w} IC(j) = \sum_{j=1}^{w} \sum_{f_b(j)>0} f_b(j) \log \frac{f_b(j)}{p_b} \qquad (3.3)$$

Meanwhile, the consensus string is not used directly since it is not accurate enough to measure weakly conserved instances. Therefore a Position-specific Weight Matrix (PWM) containing the consensus statistics will be employed to support more accurate measurement (Figure 3.1). Each cell in the PWM indicates the normalized frequency of the nucleotide in a particular position of the instance set M. Instead of the Hamming distance d_H in Equation 3.2 for the string pattern, a more accurate similarity score for evaluating each instance m_i with respect to the PWM can be obtained:



	1	2	3	4	5	6
A	0.6	0.2	0.0	0.8	0.0	0.2
T	0.2	0.0	1.0	0.0	0.0	0.0
C	0.0	0.2	0.0	0.2	0.2	0.6
G	0.2	0.6	0.0	0.0	0.8	0.2

(Consensus string *AGTAGC* is not used)

Figure 3.1: The position-led and consensus-led representations of an artificial individual and the $Score_{Sim}$ of its motif instances calculated from the PWM

$$Score_{Sim}(m_i) = \sum_{j=1}^{w} f_{m_i(j)}(j) \tag{3.4}$$

where $m_i(j) \in \Sigma$ is the nucleotide in column j of instance m_i, and $f_{m_i(j)}(j)$ is the corresponding frequency from the PWM. An illustration of the combined representation and the similarity score is shown in Figure 3.1. For example, $Score_{Sim}(m_2) = 0.2 + 0.2 + 1.0 + 0.8 + 0.8 + 0.6 = 3.6$.

3.3.3 Local Filtering Operator

One dilemma of position-led GA approaches is that an individual may be made up of a portion of positions (in other words motif instances) with high similarities between each other, yet another portion is "false positives" which are poorly aligned to the potential consensus. They cannot be distinguished nor modified efficiently by traditional genetic operators. Consensus-led approaches address this problem by scanning all the sequences each time to evaluate an individual, which imposes heavy computation. Moreover, string representations cannot measure the instances accurately.

With the complementing representations of I and PWM for consensus, the "false positives" in I can be efficiently filtered out by the novel local filtering operator. Based on $Score_{Sim}$ the local filtering operator scans for best replacements only in the sequences which contain the current worst instances to be filtered out. The procedure is as follows: firstly, the motif instances m_i within an individual is ranked by its $Score_{Sim}(m_i)$. Secondly, the sequence containing the instance with the lowest similarity score is scanned to find the replacing instance (i.e. the corresponding position) with the best $Score_{Sim}$ in that sequence. If the rank does not change, which means the best instance in this sequence is not better than that of the preceding ranked one from another sequence, then the local filtering is stopped. Else the preceding ranked $Score_{Sim}$ now becomes the lowest, and the corresponding sequence containing that instance is selected and scanned as in the first step. This step is repeated until the ranking does not change. Note that the PWM will not be updated in the local filtering for two purposes. One is to save computational load compared with on-line update, and the other is to try not to be too greedy. The pseudo-code is shown in Table 3.2.

Take the instances from Figure 3.1 as an example, after rank-

Table 3.2: Pseudo-code of local filtering operator

```
Input: Individual I = {p_1, p_2, ..., p_N}
Notation: p_i is the starting position of the motif instance m_i in
   Sequence i in I; Score_Sim(m_i) is the similarity score of m_i;
   N is the sequence number.
LOCAL_FILTER(I)
{
   Sort all the instances of I by Score_Sim(·) and obtain the
   order of sequences according to the ranking:
      Rnk(1), Rnk(2), ... Rnk(N);
   //where Score_Sim(m_Rnk(1)) is the highest score and
   //Score_Sim(m_Rnk(N)) is the lowest score
   for (k = N; k ≤ 2, k - -)
   {
      Scan sequence Rnk(k) to get q_Rnk(k) with best Score'_Sim;
      p_Rnk(k) = q_Rnk(k);
      if (Score_Sim(p_Rnk(k)) ≤ Score_Sim(p_Rnk(k-1)))
         Return the new I;
   }
}
```

ing the similarity scores, m_4 (2.6) is the worst instance and its preceding ranked instance is m_2 (3.6). So sequence 4 is scanned for the best instance against the consensus. Suppose $AGTAGG$ (4.0) is found, p_4 is updated. Since the score is better than m_2's, the sequence corresponding to m_2 will be scanned in the next iteration. The iteration goes on until for some sequence, the best instance found is still worse than its preceding ranked one. For example, if the best instance in sequence 2 is not better than m_3, local filtering is stopped.

When an individual is subject to the evolutionary process, only a small number of "false positives" need to be filtered and only a few sequences need to be scanned. Since this operator is greedy to some degree, in order to keep the contribution of evolutionary process, it is only triggered at the interval of a certain fixed number of generations.

3.3.4 Evolutionary Process

GALF showed better performance compared to different methods including other GAs [18]. With further investigation into the rough fitness landscape of motif discovery, we find it necessary to explore the search thoroughly to locate the global optima. In order to improve GALF, another evolutionary strategy is proposed to achieve more reliable performance. Different from tournament selections [18,21,29,55,77,94,101], pre-selection similar to [66] is employed to maintain the diversity in the population.

The evolutionary process is performed in the position-led representation space. All P individuals in the population are randomly partitioned into $P/2$ non-overlapping pairs. In reproduction, each pair of parents Pr_1, Pr_2 are subject to a certain crossover rate, generating two offspring Of_1 and Of_2. Both the offspring and individuals not chosen for crossovers are subject to mutation with certain mutation rate. Single-point mutation (U) and crossover (X) are used. Therefore, there are 4 possible cases, namely, X with U, X without U, U without X, and no operation.

For the first two cases with crossovers (X), replacement happens between Pr_1, Pr_2, Of_1, and Of_2. Each parent is paired with the more similar offspring, e.g. Pr_1 with Of_2, based on their Hamming distance. Accordingly Pr_2 is paired with Of_1. In each pair the one with better fitness will survive and replace the other, thus diversity and certain selection pressure are maintained.

For the third case, U without X, a mutant directly replaces its original version. The purpose is to maintain more diversity and variations. In order not to lose the potential optimum, the best-so-far individual is kept and stored separately. Faster convergence may be achieved if selection is applied where the better version replaces the worse one. However, local filtering already does the job when it is triggered, removing small variations of mutation. If such replacement is also performed, the diversity will be significantly decreased and premature convergence may happen.

Shift operator is also applied as it was in [18] to avoid stagnation of the best individual, though the operator will rarely be triggered with a very high variation rate.

3.3.5 GALF-P with Adaptive Post-processing

GALF and many other GA approaches (e.g. [21, 29]) have the limitation of assuming $k_i = 1$ in each sequence. To further extend GALF, adaptive post-processing is developed to add motif instances and remove false positives, resulting in the GALF-P framework (Table 3.1). To provide practitioners with more reliable output, in GALF-P, GALF can be run several times (MAXRUN in Table 3.1) to obtain the overall best individual I_{Best} before the post-processing is performed, similar to the way of GAME.

Post-processing in GALF-P includes adding and removing instances based on the information content IC in Equation 3.3. IC is widely employed in different TFBS identification approaches and many novel scoring functions serve as generalized extensions of IC (e.g. [40]). Since our focus is on the more effective and efficient search strategy in GAs, we have just adopted IC and more elaborate extensions on problem modeling will be addressed in future work.

Many methods add pseudo-counts to the PWM to avoid the error in computing zero logarithm for unobserved nucleotides when calculating IC in Equation 3.1. We alleviate this problem by ignoring the $f_b(j)\log\frac{f_b(j)}{p_b}$ term when $f_b(j) = 0$ in Equation 3.3, similar to the idea that events with zero probability do not contribute to entropy. This strategy works well for GALF assuming one instance per sequence ($k_i = 1$). However, the set of instances we get from GALF based on Equation 3.3 tend to be the most conserved one, i.e., each instance is the best in terms of fitness among all the instances in the same sequence. In order to accept weaker instances and reject false positives correctly, pseudo-counts are employed in the post-processing to relax the highest conservation from GALF. With the best individual I_{Best} output from GALF, its fitness is re-calculated to be $IC'_{I_{Best}}$ including pseudo-counts (1 for each nucleotide at each column).

Both the adding and removing stages of the adaptive post-processing are shown in Table 3.3. In the adding stage, the goal is to find an additional set M' whose instances on average (δ) increase $IC'_{I_{Best}}$ by more than ϵ_0, where ϵ_0 is a small constant value, intuitively proportionate to the motif width w, i.e. $\epsilon_0 = \beta * w$. ϵ_0 stands for a minimum non-trivial increase in fitness. In our experiments, β is fixed at a small value 0.001. Since the adding process adjusts δ adaptively, small changes in β do not affect the results. To include certain weaker instances in M', an initial lower bound is also set as $\delta = -\epsilon_0$. Each time when a temporary M' is created and it does not satisfy our goal of $\delta > \epsilon_0$, the adaptive lower bound will be set as $IC'_{I_{Best}} + \delta$ based on which a new M' will be created for the next iteration. The stage will converge as long as the maximal increase $\Delta > \epsilon_0$, which implies there is a non-empty M' with at least one instance to be added. In this case δ is incremented adaptively and definitely will be larger than ϵ_0 eventually. With M' added to I_{Best} we obtain I_{Best+}.

In the removing stage shown in Table 3.3, I_{Best-} is initialized as I_{Best+}, so is $IC'_{I_{Best-}}$. A new threshold $\epsilon'_0 = \max(\delta, \epsilon_0 * \gamma, \epsilon_0)$ is set. The maximum between δ and $\epsilon_0 * \gamma$ intuitively ensures the removal contributes non-trivially to the increase of IC compared to the adding, and ϵ_0 will be the minimum threshold when $\gamma = 0$. For initialization of the lower bound, $\delta' \leftarrow \epsilon'_0$. The stage iteratively removes the instance with greatest increase Δ' of $IC'_{I_{Best-}}$ among those instances satisfying the threshold criterion $IC'_{I_{Best-}} + \delta'$. If no such instance exists, the removing stage will be ended. In each iteration I_{Best-} and the corresponding $IC'_{I_{Best-}}$ are updated accordingly. The adaptive updating of $\delta' = (\epsilon'_0 + \Delta')/2$ takes into consideration both the current largest fitness increase Δ' and the initial ϵ'_0. Finally I_{Best-} is output as the solution.

The adding stage allows certain weaker instances in M' to be added and at the same time guarantees that the additional set M' on average should contribute positively and non-trivially (more than ϵ_0) to IC_{Best}. Similarly, the removing stage is stringent so that only the most probable false positives will be removed one by one. The two stages work adaptively to extend GALF for more general cases and refine the solution effectively. Both simulated and real experiments show that the adaptive post-processing is typically effective for identifying additional motif instances and removing false positives.

3.4 Results

3.4.1 Parameter Setting

The running configurations of GALF are as follows: there are 500 individuals in the population; in the experiments a maximal generation of 300 is shown to be sufficient and the stopping criterion for convergence is that the best individual does not change for 50 consecutive generations; and interval to trigger local filtering is 10 generations. For fair comparisons, we have deliberately set the same number of individuals and convergence criterion as GAME's.

In order to find out the optimal parameter settings for GALF, 54 different combinations of mutation rates (6 values: from 0.1 to 0.9 with step 0.2 and 1.0) and crossover rates (9 values: from 0.1 to 0.9 with step 0.1) are tested for the capability to locate the optimal results. 4 synthetic datasets are generated to include different sequence lengths, numbers of sequences, motif

Table 3.3: Pseudo-code of adaptive post-processing.

// Adding Stage:
Obtain the best individual $I_{Best} = \{m_1, ..., m_N\}$ output by GALF;
Calculate $IC'_{I_{Best}}$ with peudo-counts;
$\epsilon_0 \leftarrow \beta * w$; $\delta \leftarrow -\epsilon_0$;
$\Delta \leftarrow \max_{m_{i,k} \notin I_{Best}}(IC'_{+m_{i,k}} - IC'_{I_{Best}})$;
if $(\Delta \leq \epsilon_0)$ // Which means $M' = \emptyset$
{ $\gamma \leftarrow 0$; Return $I_{Best+} \leftarrow I_{Best}$; } // Adding stops
while $(\delta \leq \epsilon_0)$
{
  $M' \leftarrow \{m_{i,k} | m_{i,k} \neq m_i, IC'_{+m_{i,k}} > IC'_{I_{Best}} + \delta\}$;
  $\delta \leftarrow \mathrm{avg}_{m_{i,k} \in M'}(IC'_{+m_{i,k}} - IC'_{I_{Best}})$;
}
$\gamma = |M'|$;
Return $I_{Best+} \leftarrow I_{Best} \bigcup M'$;

// Removing Stage:
$I_{Best-} \leftarrow I_{Best+}$;
$\epsilon'_0 \leftarrow \max(\delta, \beta * w * \gamma, \beta * w)$; $\delta' \leftarrow \epsilon'_0$;
while (1)
{
  Calculate $IC'_{I_{Best-}}$ of I_{Best-} with peudo-counts;
  $M' \leftarrow \{m_{i,j} | m_{i,j} \in I_{Best-}, IC'_{-m_{i,j}} > IC'_{I_{Best-}} + \delta'\}$;
  if $(M' = \emptyset)$
  { Return I_{Best-}; }
  $\Delta' = \max_{m_{i,j} \in M'}(IC'_{-m_{i,j}} - IC'_{Best-})$;
  $I_{Best-} \leftarrow I_{Best-} -$ {the instance corresponding to Δ'};
  $\delta' \leftarrow (\epsilon'_0 + \Delta')/2$;
}

$IC'_{+m_{i,k}}$ is the IC if $m_{i,k}$ is added to I_{Best} and $IC'_{-m_{i,j}}$ is the IC if $m_{i,j}$ is removed from I_{Best-}. All IC values are calculated with pseudo-counts.

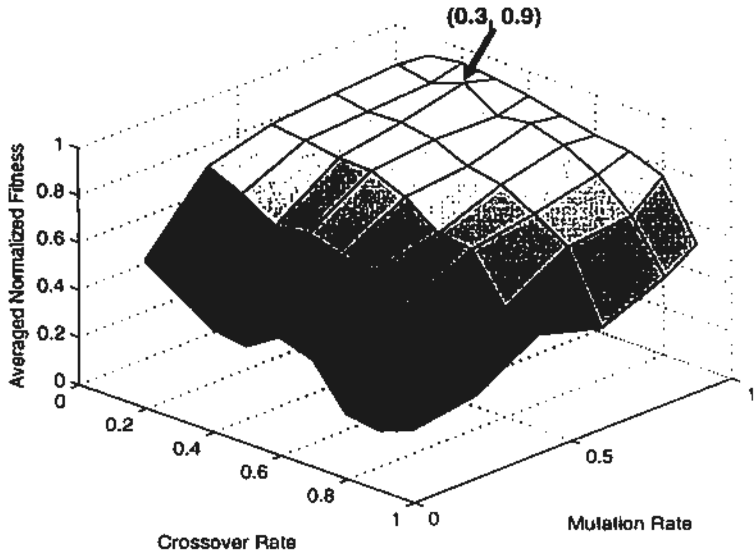


Figure 3.2: The normalized fitness averaged on all the datasets for each combination of crossover and mutation rate setting.

widths and different conservation degrees. Note that they are totally different from the synthetic datasets experimented in the next section to avoid over-training that may favor our approach. GALF is first run 20 times for each setting on each dataset, and then the average fitness is normalized for different settings. Averaging the normalized fitness for different datasets we have the average normalized fitness for evaluation. Figure 3.2 shows the averaged normalized fitness for each setting. In general, GALF favors high mutation and moderate crossover rates to keep the diversities that local filtering reduces. The best configuration is 0.9 and 0.3 for mutation and crossover rates respectively and this setting will be fixed in the following experiments. Since the post-processing in GALF-P only needs the best output from GALF of 20 runs, any setting in the high plateau in Figure 3.2 is also acceptable, although lower crossover rates need more time to converge.

3.4.2 Evaluation with Synthetic Data

In order to evaluate the performance of GALF-P for TFBS identification, a total of 800 synthetic datasets with length 300 bp for each sequence are generated with the following 8 combinations of scenarios: (1) motif width: short (8 bp) or long (16 bp); (2) number of sequences: small (20) or large (60); (3) motif conservation: high or low. For each combination, 100 datasets are generated and embedded with the instances of a random motif. In the high conservation scenario, in every column of the motif instances, the dominant nucleotide is generated with 0.91 probability (while all other 3 with 0.03 each). In the low conservation scenario, only 60% of the columns in the motif instances are as highly conserved as in the previous high conservation scenario, while 40% are lowly conserved, where the dominant nucleotide is generated only with 0.55 probability (while all other 3 with 0.15 each) in every instance. To simulate the noisy situation in real data, in each synthetic dataset, the sequences have 10% probability of containing no motif instances. In the rest of them which contain motif instances, there is 10% probability that the sequences have more than one instance. The number of additional instance(s) in the sequences follows the geometric distribution with $p = 0.5$, i.e. $P(k) = (1-p)^{k-1}p$, and therefore $k+1$ instances are embedded in such a sequence.

The performance of GALF-P is compared with GAME, MEME, Bioprospector(BioPro.), BioOptimizers based on MEME and Bioprospector (BioOpt. M. and BioOpt. B. respectively) on the synthetic datasets, with fixed motif widths. The metrics for evaluation are the precision, recall and the F-score for information retrieval [90]. Precision and recall are defined as follows:

$$Precision = \frac{\#_c}{\#_p}, \text{ and } Recall = \frac{\#_c}{\#_t} \tag{3.5}$$

where $\#_c$, $\#_p$ and $\#_t$ are the number of correctly predicted motif sites, the number of all the predicted motif sites and the number of all true motif sites embedded in the sequences, respectively. Note that shifting up to 3 bp is allowed for a correctly predicted site, according to [101]. The F-score combining both precision and recall is defined as:

$$F = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.6}$$

A high F-score indicates both precision and recall are high.

The average results for each combined scenario are shown in Table 3.4. Best F-scores are bolded. Since BioOpt.M. does not improve any result of MEME with respect to the evaluation, only MEME results are shown to save space. GALF-P achieves the best average F-score and average recall. GALF-P not only has comparable performance to the best approaches in the relatively easy scenarios (high conservation), but also gives the best results in all difficult ones (low conservation) when other approaches deteriorate significantly and find no true motifs in some datasets (details not shown). These difficult scenarios are more close to the real datasets and the results match well with the real dataset experiments in the next section. Thus we believe that GALF-P is superior to other methods in finding the optimal motifs in more realistic (usually difficult) cases. Note that the assumption of one instance per sequence ($k_i = 1$) for GALF is violated in all the synthetic datasets. However, GALF-P can still achieve respectively 0.97, 0.99 and 0.98 for average precision, recall and F-score in one scenario, demonstrating the adaptive post-processing is effective to tackle the general assumptions in real motif problems.

3.4.3 Experiments on Real Datasets

In this section, the one-run results of GALF-P are compared with the reported ones from [101] of GAME, MEME, Bioprospec-

Table 3.4: Average results for the synthetic datasets experiment: Width for the motif width, Num for the number of sequences, Con for conservation degree, P for precison, R for recall and F for F-score.

Scenario	GALF-P			GAME			MEME			BioPros.			BioOpt.B.		
Width/Num/Con	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Short/Small/Low	0.38	0.56	**0.44**	0.29	0.32	0.30	0.49	0.34	0.39	0.43	0.36	0.39	0.44	0.36	0.39
Short/Large/Low	0.52	0.59	**0.55**	0.42	0.32	0.36	0.63	0.33	0.42	0.58	0.37	0.45	0.59	0.37	0.45
Long/Small/Low	0.87	0.91	**0.89**	0.78	0.87	0.82	0.91	0.86	0.88	0.96	0.74	0.83	0.96	0.74	0.83
Long/Large/Low	0.91	0.90	**0.91**	0.92	0.90	0.90	0.96	0.85	**0.90**	0.98	0.68	0.80	0.98	0.68	0.80
Short/Small/High	0.73	0.90	0.80	0.71	0.80	0.75	0.87	0.84	**0.85**	0.82	0.75	0.78	0.83	0.76	0.79
Short/Large/High	0.81	0.86	**0.83**	0.83	0.83	**0.83**	0.91	0.76	**0.83**	0.88	0.68	0.76	0.88	0.68	0.76
Long/Small/High	0.97	0.99	**0.98**	0.94	0.99	0.97	0.98	0.99	**0.98**	1.00	0.94	0.97	1.00	0.94	0.97
Long/Large/High	0.97	0.97	0.97	0.98	0.99	**0.98**	0.99	0.98	**0.98**	1.00	0.92	0.96	1.00	0.92	0.96
Average	0.77	0.84	**0.80**	0.73	0.75	0.74	0.84	0.74	0.78	0.83	0.68	0.74	0.83	0.68	0.74

Table 3.5: The 8 real datasets. N is the number of sequences, l is the sequence length, w is the motif width, and $\#_t$ is the number of TFBSs embedded.

Dataset	CREB	CRP	ERE	E2F	MEF2	MYOD	SRF	TBP
N	17	18	25	25	17	17	20	95
l	200	105	200	200	200	200	200	200
w	8	22	13	11	7	6	10	6
$\#_t$	19	23	25	27	17	21	36	95

tor (BioPro.), BioOptimizers based on MEME and Bioprospector (BioOpt. M. and BioOpt. B.) on the 8 real datasets tested by [101]. The details of the datasets are shown in Table 3.5. The CRP dataset contains the binding sites for cyclic AMP receptor, and has been widely tested since [97] was published. The ERE dataset contains the binding sites for the ligand-activated enhancer protein estrogen receptor (ER) [47]. The E2F datsets correspond to TFBSs of the E2F family from mammalian sequences [46]. CREB, MEF2, MyoD, SRF and TBP are chosen from the ABS database of annotated regulatory binding sites [13]. More details of the datasets can be found in [101]. Different ranges of motif widths, numbers of sequences as well as numbers of embedded TFBSs are covered. The evaluation criteria are also the precision, recall and F-score. Again up to 3 shifts are allowed for a correctly predicted site.

The results in terms of F-scores are compared in Table 3.6 (the whole table containing precisions and recalls is not shown for simplicity). The best results are bolded. GALF-P has the best results in 7 out of the 8 datasets as well as the overall average. GAME is ranked second best in most of the datasets with one best F-score. On average, GALF-P (0.83) and GAME (0.77) give significantly better F-scores than the other methods (0.56-0.61). Furthermore, GALF-P achieves the best average precision (0.81), recall (0.87) and F-score (0.83) while GAME is the second best in terms of all these three metrics (0.78, 0.77 and 0.77 respectively).

Table 3.6: Comparisons of F-scores on the 8 real datasets.

Dataset	GALF-P	GAME	BioOpt.M.	BioOpt.B.	MEME	BioPro.
CREB	**0.76**	0.73	0.59	0.67	0.59	0.67
CRP	**0.85**	0.80	0.67	0.67	0.67	0.78
ERE	**0.79**	0.75	0.72	0.75	0.71	0.68
E2F	0.81	**0.90**	0.74	0.70	0.76	0.46
MEF2	**0.97**	0.88	0.88	0.61	0.88	0.71
MYOD	**0.72**	0.48	0.00	0.00	0.00	0.00
SRF	**0.84**	0.80	0.74	0.74	0.67	0.70
TBP	**0.89**	0.84	0.40	0.75	0.36	0.71
Average	**0.83**	0.77	0.59	0.61	0.58	0.56

3.4.4 Comparisons between GALF-P and GAME

Since GALF-P and GAME are the best two methods and our focus is on GAs, further experiments are performed to compare GALF-P and GAME. To make a detailed comparison, we run both GALF-P and GAME with fixed motif widths on the same 8 datasets, each run 20 times. In each run, the GA procedures, namely GALF, and the GA procedure in GAME are both run 20 times to obtain the best individuals, due to the stochastic nature of GAs, before post-processing is applied. Their best and average performances are compared based on the above metrics.

The best results in terms of F-scores (with the associated precisions and recalls) are shown in Table 3.7. Numeric formats for the corresponding precisions and recalls are shown in parentheses. The better results between GAME and GALF-P are bolded.

GALF-P gives better precisions compared to GAME in 7 out of the 8 datasets, thanks to the superior performance of GALF, which achieves very high precisions (0.88 on average). On the other hand, GALF-P obtains comparable recalls (6 better than or the same as GAME) among which the optimal recalls for MEF2 and MYOD are obtained. Moreover, GALF-P achieves better precision, recall and F-score than GAME averaged over the 8 datasets.

For the average performance in 20 runs shown in Table 3.8, the differences between GAME and GALF-P are even larger.

Table 3.7: Comparisons of GALF-P and GAME on the 8 datasets for 20 runs: Best results (in terms of F-scores, together with the corresponding precisions and recalls). Datasets satisfying one instance per sequence are labelled with "*"s.

Dataset	GAME			GALF-P		
	Precision	Recall	F-score	Precision	Recall	F-score
CREB	**14/18 (0.78)**	14/19 (0.74)	**0.76**	16/23 (0.70)	**16/19(0.84)**	**0.76**
CRP	18/21 (0.86)	**18/23 (0.78)**	0.82	**17/17 (1.00)**	17/23 (0.74)	**0.85**
ERE*	20/38 (0.53)	**20/25 (0.80)**	0.63	**19/23 (0.83)**	19/25 (0.76)	**0.79**
E2F	24/30 (0.80)	**24/27 (0.89)**	0.84	**24/29 (0.83)**	**24/27 (0.89)**	**0.86**
MEF2*	17/19 (0.89)	**17/17 (1.00)**	0.94	**17/18 (0.94)**	**17/17 (1.00)**	**0.97**
MYOD	10/21 (0.48)	10/21 (0.48)	0.48	**21/37 (0.57)**	**21/21 (1.00)**	**0.72**
SRF	33/45 (0.73)	**33/36 (0.92)**	0.81	**33/43 (0.79)**	**33/36 (0.92)**	**0.84**
TBP*	81/101 (0.80)	81/95 (0.85)	0.83	**86/93 (0.92)**	**86/95 (0.91)**	**0.91**
Average	0.73	0.81	0.76	**0.82**	**0.88**	**0.84**

The better results between GAME and GALF-P are bolded. GALF-P achieves better precisions for all but one dataset and better recalls for 5 datasets. In 7 of the 8 sets GALF-P obtains better F-score than GAME. As a result, the average precision, recall and F-score averaged over the 8 sets are all significantly better for GALF-P (by more than 20%). It implies that GALF-P is more stable and reliable in identifying the TFBSs correctly. We discover that during some runs for datasets CREB, MEF2 and MYOD, GAME was trapped in local optima, indicated by the lower reported fitness values compared with the best ones GAME achieved in the 20 runs. As a result, GAME failed to identify any of the motifs in some runs. This suggests that GAME's GA procedure is not elaborately designed or fully optimized, producing inconsistent results in difficult problems with many local optima. On the other hand, the average results of GALF-P (precision 0.80; recall 0.87; F-score 0.82) are consistent and comparable with its best results (precision 0.82; recall 0.88; F-score 0.84), demonstrating the robust performance of GALF-P, which is also indicated by the generally smaller standard deviations for CREB, MEF2 and MYOD in Table 3.8.

Table 3.8: Comparisons of GALF-P and GAME on the 8 datasets for 20 runs: Average results (precisions, recalls and F-scores are averaged separately). With the $\pm$ symbols are the standard deviations. Datasets satisfying one instance per sequence are labelled with "*"s.

Dataset	GAME			GALF-P		
	Precision	Recall	F-score	Precision	Recall	F-score
CREB	0.43±0.36	0.42±0.36	0.42±0.35	**0.70±0.00**	**0.84±0.00**	**0.76±0.00**
CRP	0.79±0.02	**0.78±0.00**	0.78±0.01	**0.99±0.03**	0.73±0.02	**0.84±0.03**
ERE*	0.52±0.03	**0.78±0.08**	0.62±0.05	**0.82±0.01**	0.76±0.01	**0.79±0.00**
E2F	**0.79±0.02**	**0.87±0.02**	**0.83±0.02**	0.77±0.02	0.85±0.01	0.81±0.01
MEF2*	0.52±0.37	0.55±0.40	0.53±0.37	**0.91±0.09**	**0.98±0.08**	**0.95±0.09**
MYOD	0.14±0.20	0.14±0.19	0.14±0.20	**0.57±0.00**	**1.00±0.00**	**0.72±0.00**
SRF	0.71±0.01	0.86±0.01	0.78±0.01	**0.75±0.03**	**0.89±0.06**	**0.82±0.05**
TBP*	0.81±0.08	0.74±0.11	0.77±0.09	**0.87±0.04**	**0.87±0.02**	**0.87±0.02**
Average	0.59	0.64	0.61	**0.80**	**0.87**	**0.82**

3.4.5 Complexity and Efficiency

To evaluate the efficiency, we analyze the complexity of the evolutionary process of the GA in GAME, and GALF in GALF-P. Suppose there are N sequences, each with the same length l. Motif width is w. Population size is P which is the same for GALF and GAME.

In summary, the overall complexities for GAME and GALF respectively are:

$$C_{GAME} = O(G_1 * P * N * w)$$
$$C_{GALF} = O(G_2 * P * N * (w + 0.1 * (\log N + l/k)))$$

where 0.1 indicates local filtering is triggered once every 10 generations, $1/k$ is the averaged percentage of sequences scanned in local filtering, and G_1 and G_2 denote the different maximum generations required in GAME and GALF respectively.

In fact, C_{GALF} has higher complexity than C_{GAME} when the same generations are used and N and/or l are sufficiently large. However, due to the local filtering, GALF achieves convergence within a maximum $G_2 = 300$ generations in the experiments, while GAME requires $G_1 = 3000$ as the maximum generations. Notice that in real cases, usually $w \geq 5$ and $l \leq 1000$ in the promoter regions. The break even point of $C_{GALF} > C_{GAME}$

requires: $N \approx 2^{G_1/(G_2*0.1)*w} = 2^{100*w}$ when quick sort is used in local filtering ($N \approx 100 * w$ even if bubble sort is used), or $l \approx k*w*G_1/(G_2*0.1) = 100*w*k$. k drops significantly according to the real dataset experiments, with the average recorded $k = 4.52$. So it is seldom that $C_{GALF} > C_{GAME}$ in the real cases (w is about 10 to 20 and l is within a few thousand bp (usually within 3000 bp)) of TFBS identification and thus GALF is usually more efficient than GAME.

However, it is not easy to compare the efficiency between the GA in GAME and GALF. Subject to premature convergence in real problems, the maximal generations may not be used up. Another difficulty is that GAME is implemented in JAVA while GALF-P is implemented in C. Moreover, GAME can only be timed with the GA and post-processing as a whole (and thus we time GALF-P in the same way). The comparison on running time is not a reliable indicator of the efficiency of the algorithms, thus the result quality rather than computing time is the major concern. Nevertheless it can be a reference for the practitioners who have arguments on the slow running time of GAs.

In the previous experiments, GALF-P and GAME are both executed on the same Pentium D 3.00 GHz machine with 1GB memory, running Windows XP. GALF-P is on average 4.49 times (3.11 to 10.29 times) faster than GAME (Table 3.9). GALF-P and GAME require 61.91s and 291.11s on average respectively, showing that GAs can provide a reasonable computation solution for the problem.

3.5 Discussion and Conclusion

As a GA based method for TFBS identification, GAME shows better performance than other approaches. However, the basic GA in GAME is not elaborately designed or fully optimized. In the noisy circumstances for motif discovery in real applications,

Table 3.9: Average computation time on the 8 datasets between GAME and GALF-P.

	GAME	GALF-P	Speedup
CREB	133.00	42.75	3.11
CRP	380.05	98.20	3.87
ERE	334.20	83.20	4.02
E2F	288.65	86.95	3.32
MEF2	112.05	34.40	3.26
MYOD	91.05	26.25	3.47
SRF	224.05	49.10	4.56
TBP	765.80	74.40	10.29
Average	291.11	61.91	4.49

GAME is likely to be trapped by local optima and the GA results significantly affect the final output in despite of any elaborate post-processing.

In this chapter, GALF, employing the combined representations associated with a novel local filtering operator and advanced evolutionary process, has been proposed to provide a more effective and efficient GA search algorithm than GAME and other approaches. We have further extended GALF to the GALF-P framework by integrating carefully designed adaptive post-processing. GALF-P gives superior results in the difficult (realistic) synthetic datasets and outperforms GAME in terms of precision, recall and F-score averaged on the 8 datasets tested in [101]. Moreover, GALF-P shows more stable and reliable performance than GAME and hence should be favored by practitioners. A recent version of GALF-P is also available to identify instances on both forward and reverse strands.

Further efforts will be put in for several issues, the most important one of which is the fitness function. Since our concern in this chapter is mainly on improved GA-based searching methods rather than developing a new model for the fitness function, the widely adopted IC (also serves as a core part of the Bayesian scoring function for GAME) is employed. Nevertheless, we believe appropriate domain knowledge can be incorporated for a more realistic fitness model. More complete work on the modeling will be addressed in the future. Another challenging and interesting topic is to design a novel multi-modal GA to discover multiple motifs in a single run, rather than several runs with masking techniques.

□ **End of chapter.**

Chapter 4

TFBS Motif Discovery with GALF-G: The Modeling Aspect

Summary

GALF-G is presented to address the modeling aspect of TFBS motif discovery. The modeling generalizes substantial assumptions, allowing uncertain motif widths, relaxing OOPS and ZOOPS assumptions, and discovering multiple TFBS motifs simultaneously.
Additional file 1 available at:
http://www.cse.cuhk.edu.hk/%7Etmchan/GALFG/

4.1 Introduction

Although the previous GALF-P shows outstanding results in search/optimization based on an existing model, the TFBS motif discovery problem is still challenging with respect to the modeling. Real TFBSs of a motif may vary in their widths and their conservation degrees within a certain range. Deciding a single motif width by existing models may be biased and misleading. Additionally, multiple, possibly overlapping, candidate motifs are desired and necessary for biological verification in practice. However, current techniques either prohibit overlapping TFBSs or lack explicit control of different motifs.

In this chapter, we propose a new generalized model to tackle the motif widths by considering and evaluating a width range of interest simultaneously, which should better address the width uncertainty. Moreover, a meta-convergence framework for genetic algorithms (GAs), is proposed to provide multiple overlapping optimal motifs simultaneously in an effective and flexible way. Users can easily specify the difference amongst expected motif kinds via similarity test. Incorporating Genetic Algorithm with Local Filtering (GALF) for searching, the new GALF-G (G for generalized) algorithm is proposed based on the generalized model and meta-convergence framework.

GALF-G was tested extensively on over 970 synthetic, real and benchmark datasets, and is usually better than the state-of-the-art methods. The range model shows an increase in sensitivity compared with the single-width ones, while providing competitive precisions on the *E. coli* benchmark. Effectiveness can be maintained even using a very small population, exhibiting very competitive efficiency. In discovering multiple overlapping motifs in a real liver-specific dataset, GALF-G outperforms MEME by up to 73% in overall F-scores. GALF-G also helps to discover an additional motif which has probably not been annotated in the dataset.

4.2 Motivations

Challenges

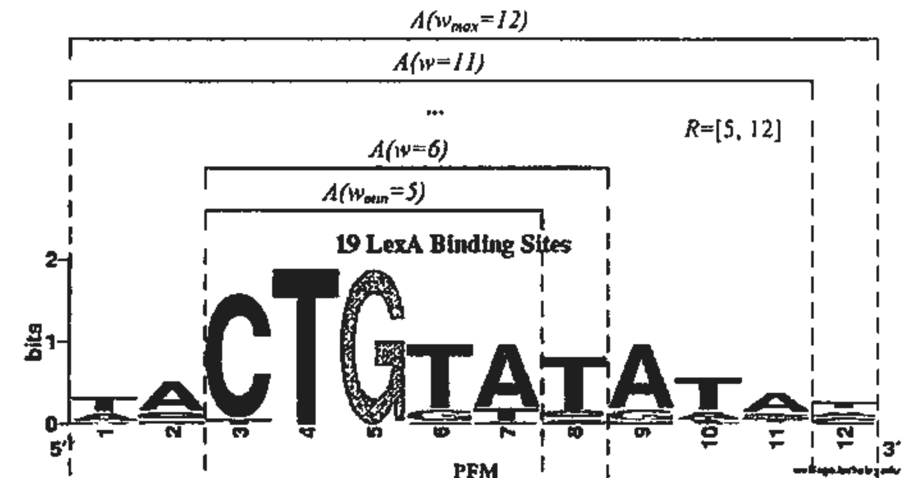
Great challenges exist for *de novo* motif discovery algorithms to succeed. Challenges mainly include (i) NP hardness (ii), width uncertainty and (iii) multiple (overlapping) motifs, of which the latter two demand for more focus.

- **(i) NP hardness:** The most well-known challenge is the NP hardness [53] due to the unknown conservation degree, where extensive approaches have been proposed to achieve optimality under certain models, as surveyed in the last sub-sections.

- **(ii) Width uncertainty:** An often overlooked challenge in real-life problems is the uncertainty in the motif widths.

  In real datasets, it is not easy to determine a single motif width (1) experimentally or (2) biologically. (1) Experimental: Annotated TFBSs are often affected by limited experimental resolutions, and it is thus difficult to choose any single width to fit the TFBSs before a motif can be discovered. (2) Biological: The most conserved binding contacts are between the short binding core of the target TFBS and the binding domain of a TF. The binding core may be fixed-width (<6bp). However, the short binding core may not provide enough binding affinity for its corresponding TF to recognize. Instead, a TF contain flexible segments of polypeptide chain, and these flexible arms work together with the DNA binding domain of the TF to add additional affinity [32]. The complication makes the effective width not easy to be fixed at a single value. For example, the TFBS widths vary in the familial binding cases of the Zn2-Cys6 motif [74].

  Existing methods usually assume a known and fixed TFBS motif width or model a distribution around an expected width when there are uncertainties involved. The conservation contributed from different motif parts by varying the widths may be under-utilized in a single-width approach, and the so-called expected value may be misleading and biased. Statistical significance to rank different widths, e.g. E-value [35], is computational intensive and still only picks a single-value width at the end. In the illustrative example of a real motif with 19 LexA binding sites in Figure 4.1, if a single width is chosen, it may be 5 if only the stringent core part (3-7) is chosen; or it may be 12 if considering all columns (1-12). In the former case, the short motif may not be ranked higher than those non-TFBS frequent patterns happening by chance. In the latter case, since both highly and weakly conserved columns are evaluated equally, it may include additional false positives. On the contrary, modelling those uncertain bases with a range concept can better capture the different resolutions for assessing the motif signals, and thus potentially better describe the real TFBS motif.



	1	2	3	4	5	6	7	8	9	10	11	12
A	0.26	0.68	0.00	0.00	0.00	0.05	0.79	0.05	0.79	0.21	0.63	0.32
C	0.10	0.16	0.95	0.00	0.00	0.00	0.05	0.11	0.00	0.05	0.16	0.32
G	0.05	0.11	0.00	0.00	1.00	0.16	0.00	0.05	0.16	0.05	0.05	0.00
T	0.57	0.05	0.05	1.00	0.00	0.79	0.16	0.79	0.05	0.68	0.16	0.37

Figure 4.1: An example of the generalized model on the motif of 19 real LexA binding sites (the first 12 columns) from the SequenceLogo website. Each $A(w_i)$ is chosen based on the maximal $P(A(w_i))$, where the bits bounded by the red dashes reflect $P(A(w_i))$ for illustrative purpose. In practice, $P(A(w_i))$ can be chosen flexibly.

- **(iii) Multiple (overlapping) motifs:** Another challenge which is not well handled is the overlapping nature of TF-

BSs for different motifs because competitive binding exists amongst different TFs in the same regulatory region. Current techniques used are mainly masking/erasing and implicit maintaining.

- Masking/erasing: These techniques can only discover one motif in a single execution, and thus several executions are required for outputting multiple motifs. Masking/erasing techniques also prohibits the subsequent discovery of the TFBSs overlapped with those previously masked ones. However, in real cases, different kinds of TFBSs may overlap with each other due to competitive binding of TFs.
- Implicit maintaining: There are existing methods to sample different motifs simultaneously but with little or no mechanism to explicitly distinguish different solutions or flexibly control the overlapping degrees of TFBSs. As a result, highly redundant motifs may be produced. If there are limited number of output solutions, redundant top-scored variant motifs will dominate and less-fit but different solutions will be missed. If non-redundant and different solutions need to be provided, a large output number has to be set and post-processing is required [34] with additional costs.

Therefore, it is desirable to discover multiple motifs more effectively and efficiently with certain flexible and explicit overlapping control.

Chapter Outline

To overcome all these drawbacks of the existing de novo motif discovery algorithms, we propose **the generalized model** which presents a new angle to handle the variable motif widths to better reflects the biological uncertainty. Then we present **the meta-convergence framework** to support multiple optimal solutions with flexible overlapping control using similarity tests. Based on the generalized model and the framework, a new algorithm called **GALF-G** is developed.

The rest of the chapter is arranged as follows. The generalized model, the meta-convergence framework and the new algorithm GALF-G are first presented briefly in the Proposed Methods section, followed by the Detailed Implementations section. Extensive experimental results are then reported in the Experiments section, including single/multiple motif discovery problems with fixed-width/variable widths inputs. A large number of both synthetic and real benchmark datasets are used in the experiments. Discussion and conclusive remarks are finally given.

4.3 Proposed Methods

In this section, we present **the generalized model** and **the meta-convergence framework** in brief, which form the **GALF-G** algorithm.

4.3.1 The Generalized Motif Model

To tackle the challenge raised from the uncertainty of motif widths, we propose a new generalized model by considering a width range of interest simultaneously. A range is more practical and suitable for real biological cases for two reasons:

- First, it is easier to define a rough range than a particular width. All widths within contribute accordingly to the motif solution, and thus it is less sensitive than a wrongly chosen single width.

- Second, TFBSs of a motif in reality vary in their widths and exhibit certain higher degrees of conservation compared to the non-site fragments (the background). A range model can more appropriately capture the different conservation degrees than any single width.

Assume the width input is $R = [w_{min}, w_{max}]$ and $|R| = w_{max} - w_{min} + 1$, and a candidate solution, i.e. a set of TFBSs to form a motif, is defined as A, with the TFBS positions denoted by $\{p_i\}$. The formal problem denotations and formulations are shown in the Methods section: The Proposed Model and Evaluation. The generalized model evaluates A based on the whole range R. An illustrative example is shown in Figure 4.1. The model or scoring function (illustrated by the heights of color nucleotides in the figure) for a fixed width w_i is well established, e.g. a probabilistic model, denoted as $P(A(w_i)|w_i)$, where $P(A(w_i))$ is a part from the complete candidate solution A with respect to w_i. The generalized model can then be formulated by summing them together as

$$P(A) = \sum_{w_i \in R} P(A(w_i)|w_i)P(w_i). \tag{4.1}$$

For the most common case when there is no prior knowledge on which width is more likely to happen, w_i can take a uniform distribution, i.e. $P(w_i) = 1/|R|$ for each w_i. On the other hand, any prior distribution such as the Poisson one used in Bayesian models [40] can be also adopted. For each w_i-component where $w_{min} \leq w_i < w_{max}$, there are more than one choice and we only pick the component $A(w_i)$ by $\arg max(P(A(w_i)|w_i))$ (caps in Figure 4.1). The additional computational cost compared to a fixed width model is $O(|R|^2)$, which is feasible since motif ranges (width variations) are usually short ($\leq$ 10bp). The major difference of the generalized model from the previous ones is that all the widths from the input range R contribute to the solution score/fitness in the model, rather than choosing a certain single width by $argmax(P(A(w_i)|w_i)P(w_i))$, which has the risk of bias on a certain single value. If only one width is input, the generalized model reduces to one of the existing fixed-width models.

Intuitively, the generalized model is a weighted sum of the probability of different widths from the range R. It is compatible with the existing probability models and is even applicable to non-probability models, as long as there is a consistent expression of $P(A(w_i))$; here it refers to an evaluation function in general. We employ the fixed-width probabilistic model in our generalized model, which will be discussed in detail in the Methods section.

4.3.2 The Meta-convergence Framework

For practitioners in molecular biology and medical research, it is desirable that multiple optimal candidate motifs can be provided concurrently for biological verification. Due to the limitations of masking/erasing and implicit maintaining, it is desired to explicitly maintain different solutions with flexible (typically overlapping) control efficiently. To address these issues, we propose a meta-convergence framework employing Genetic Algorithm (GA) with the similarity test as the overlapping control.

(i) The similarity test is first introduced to fulfill flexible overlapping control over different motifs. Positional information is considered since the generalized model involves a width range R of positions. In particular, to compare two candidate solutions/individuals A_a and A_b, the test calculates the relaxed Hamming distance h between each pair of their aligned TFBS positions: $p'_i(A_a)$ and $p'_i(A_b)$ in sequence i,

$$h(p'_i(A_a), p'_i(A_b)) = \begin{cases} 0 & \text{if } |p'_i(A_a) - p'_i(A_b)| \leq tol \\ 1 & \text{otherwise.} \end{cases} \tag{4.2}$$

where *tol* is the shift tolerance. The similarity test is passed, if

$$dr = (\sum_{i=1}^{m} h(p_i'(A_a), p_i'(A_b)))/m < st \tag{4.3}$$

, where dr is defined as the difference ratio, m indicates the number of sequences, and st is the similarity threshold. When $dr < st$, A_a and A_b are considered to be similar, i.e. belong to the same motif kind. The intuitive settings of tol, st for different purposes, and how the test is applied are detailed and included in Methods: Meta-convergence Framework Details.

The similarity test proposed allows users to control the differences between the expected motifs in an easy and intuitive way. On the contrary, the other possible comparisons based on the PFM involve complicated cut-off which is not trivial to specify and counterintuitive for common users.

(ii) Meta-convergence, with the similarity test, monitors the convergence of different optimal solutions and adaptively controls the numbers of GA runs rather than using a relatively large fixed number of GA runs in previous works [19, 101]. Furthermore, only a small number of candidates are subject to the similarity test to compete for the multiple optimal motifs, compared with the other method [61] that compares the whole population of solutions with non-trivial overhead. Therefore, the efficiency can be significantly improved. More details can be found in Methods: Meta-convergence Framework Details.

4.3.3 GALF-G

Incorporating Genetic Algorithm with Local Filtering (GALF) with the generalized model and the meta-convergence framework, GALF-G (G for generalized) is proposed to discover multiple optimal motifs with flexible overlapping control using the similarity test. To fit into the generalized model with range input, the operators in GALF are extended accordingly and detailed in the Methods section: GALF-G implementations.

In the following section, we will report the results of GALF-G tested on both synthetic and real benchmark datasets for various cases, namely fixed-width, variable width, for single motif [with single ($K = 1$) or multiple outputs ($K > 1$) for single motif] and multiple motifs ($K > 1$) discoveries.

4.4 Detailed Implementations

4.4.1 The Proposed Model and Evaluation

Denotations and Formulations

With our focus on the matrix representation (PFM), the motif discovery problem is formulated as follows. Defined on the alphabet $\Sigma = \{A, T, G, C\}$ for DNA sequences, the input data are a set of sequences $S = \{S_i | i = 1, 2, ..., m\}$, where each S_i is a sequence with length l_i of nucleotides from the alphabet. The motif width w is assumed to be known for the time being. TFBS instances are represented by $R = \{r_i^k\}$ where each r_i^k is the kth instance of width w in S_i. If we assume each sequence has at most one instance (ZOOPS), then $r_i^{k=0,1}$ is collapsed to be r_i ($r_i =$ null if $k = 0$) for short. Table 2.1 illustrates an artificial example of motif discovery. A site indicator matrix (SIM) A, which is also used to represent the solution, locates the TFBS instances as sites, where $A_{ij} = 1$ if a motif instance (site) starts at position j of S_i and 0 otherwise. Alternatively, we can use the position $p_i^k = j$ to represent a instance r_i^k given w. Thus we have a compact position representation of $A = \{p_1, p_2, ..., p_m\}$ especially for ZOOPS, where some the positions can be NULL. A profile of the motif can be built from aligning the TFBS instances indexed by A. The profile is represented as a $4 \times w$ Position Frequency Matrix (PFM) Θ, where Θ_{jb} is the frequency of nucleotide b in column j of the motif. The nucleotides from background (non-motif sites) are represented by Θ_0, where Θ_{0b} is the frequency of nucleotide b in the background and is treated as known from the input.

The motif discovery problem (of a known width w) can be thus formulated as finding A (with only the TFBS sites being considered) and the corresponding PFM Θ such that one of the above scoring/fitness functions is maximized according to different assumptions.

The Probabilistic Models

To complete our generalized model, the important component comes from the existing models handling a known width input. In this chapter, we employ the probabilistic models which have most intuitive explanation with the generalized model. For a candidate solution A (which also indicates Θ), the full Bayesian model of likelihood [40,41] can be written as

$$\begin{aligned} p(\Theta, A|S, \Theta_0) &\propto p(S|\Theta_0, \Theta, A)p(A|p_0)p(\Theta)p(p_0) \\ &\propto \prod_{j=1}^{w}\prod_{b\in\Sigma}\Theta_{jb}^{n_{jb}}\prod_{b\in\Sigma}\Theta_{0b}^{n_{0b}}p_0^{|A|}(1-p_0)^{L^*-|A|}p(\Theta)p(p_0) \qquad (4.4) \end{aligned}$$

where Θ is the motif PFM, Θ_{0b} is the background distribution of nucleotide b, n_{jb} is the count of nucleotide b in column j of the PFM, n_{0b} is the count of nucleotide b in the background, $|A|$ is the total number of sites in the motif, $L^* = \sum_{i=1}^{m}(l_i+1)$ is approximately the number of all possible sites (the number of invalid sites is trivial and can be ignored), and $p_0 = |A|/L^*$ is the estimated abundance ratio which represents the probability of any position being a site in the dataset. $\Theta_{jb} = n_{jb}/|A|$ (strictly it should be $\widehat{\Theta}_{jb}$ as an estimate, but we just use Θ_{jb} for simplicity). Similarly $\Theta_{0b} \approx n_{0b}/L^*$ (ignoring the relatively small affect of A).

In Bayesian analysis, noninformative priors of the independent $p(\Theta)$ and $p(p)$ are integrated out for convenience. Alternatively, by assuming them as constant we have the log likelihood as follows:

$$\log p(\Theta, A|S, \Theta_0) \propto |A| \sum_{j=1}^{w} \sum_{b\in\Sigma} \Theta_{jb} \log \Theta_{jb} \tag{4.5}$$

$$+ \sum_{b\in\Sigma} (L^* \Theta_{0b} - |A| \sum_{j=1}^{w} \Theta_{jb}) \log \Theta_{0b}$$

$$+ |A| \log p_0 + (L^* - |A|) \log(1 - p_0)) \tag{4.6}$$

By ignoring the constant parts and approximating $L^* \log(1 - p_0) \approx -L^* * p_0 = -|A|$ since p_0 is very small, the equivalent score psi' can be written as

$$\psi'(\Theta, A|S, \Theta_0) = |A|(\sum_{j=1}^{w} \sum_{b\in\Sigma} \Theta_{jb} \log \frac{\Theta_{jb}}{\Theta_{0b}} + \log \frac{p_0}{1 - p_0} - 1). \tag{4.7}$$

which is exactly the approximation form used in the Bayesian analysis [40]. With one step further to ignore the penalty of $-|A|$, we have the approximation form for a known p [40] and it is also coined as the Kullback-Leibler divergence with parameter (we use this form in the generalized model since we find the previous one imposes too much penalty on the number of TFBSs):

$$\psi(\Theta, A|S, \Theta_0) = |A|(\sum_{j=1}^{w} \sum_{b\in\Sigma} \Theta_{jb} \log \frac{\Theta_{jb}}{\Theta_{0b}} + \log \frac{p_0}{1 - p_0}). \tag{4.8}$$

Furthermore, if we assume each sequence S_i has exactly one site, i.e. one occurrence per sequence (OOPS), then p_0 also becomes constant. As a result we only have to consider part of Equation 4.8

$$IC = \sum_{j=1}^{w} IC(j) = \sum_{j=1}^{w} \sum_{b \in \Sigma} \Theta_{jb} \log \frac{\Theta_{jb}}{\Theta_{0b}} \tag{4.9}$$

which is the well known information content (IC) [96]. $IC(j)$ is defined as the positional IC for column j.

The Fitness Function and Evaluation

Recalling the generalized model in Equation 4.1, we can now choose $P(A(w_i)|w_i) = \exp(\psi(w_i))$ accordingly from the previous probabilistic models, where $\psi(w_i)$ is a simplified notation for exactly $\psi(\Theta, A|S, \Theta_0)$ in Equation 4.8 given w_i. For computational convenience, we represent the fitness function f in log likelihood form as

$$f = \log(\sum_{w_i \in R} p(w_i) \exp(\psi(w_i))). \tag{4.10}$$

In the evaluation, a candidate solution consists of A (and the derived Θ) with the maximal width w_{max}. For each particular w_i from the range R, we have to choose the fragment (a continuous w_i-submatrix $A(w_i)$ from the full matrix Θ) that maximizes $\psi(w_i)$ (see Figure 4.1). It is equivalent to maximizing IC for width w_i since p in Equation 4.8 is now fixed for all $A(w_i)$. With the log format of f, we can avoid overflow with the exp function by taking out the largest log component during mediate computation and adding it back upon finishing the evaluation.

For the convenience of implementations of searching and consistency with other methods for evaluation (which output single-width motifs), a core fragment, located by the width w_{cor} and offset w_0, is to be selected. w_{cor} and w_0 are also determined based on IC. Starting from the two ends of the maximal PFM with w_{max}, we iteratively remove each columns j with positional $IC(j)$ lower than the average. The remaining submatrix (or

$A(w_{cor})$) is thus with width w_{cor} and offset w_0. Complexity of the whole evaluation grows quadratic to $|R| = w_{max} - w_{min} + 1$. Since the ranges are usually restricted within 5 - 10bp, f is computationally feasible in practice with additional $O(|R|^2)$ overhead compared with a fixed width model for w_{max}. The offset w_0, combined with the position p_i of A in the i^{th} sequence, is also used to determine the aligned position ($p'_i(A)$) in the similarity test in Equation 4.2.

4.4.2 Meta-convergence Framework Details

Similarity test settings

The shift tolerance in Equation 4.2 is set as $tol = 3 + (|R| - 1)/2$. The first part of tol is chosen for convenience to separate two TFBSs and the latter part is the tolerance for the range involved.

In the case of competition for the same slot in slot dispatching, the threshold can be flexibly specified by the users (for general usage, the default is: $st = 0.3$, which is used throughout this chapter). Users can customize st based on their needs, either with a large value (e.g. ≥ 0.5) to force solutions of highly different motifs, or with a small value (e.g. ≤ 0.1) to allow fine variations of the same motif type. On the other hand, for deleting individuals in the case of near convergence, the threshold is automatically fixed at the value of $st' = 0.5$ to make room for the other solutions. st' is not sensitive because the similar optimal motifs are finally controlled by the user-specified threshold st. However, if st' is set to be too low, many similar variations to the converged motif will remain in the population, and time will be wasted to converge repeatedly to the same motif kind.

Meta-convergence

In greater detail, the meta-convergence framework can incorporate any GA procedure (Genetic Algorithm with Local Filtering

(GALF) [19] in our case). Like in the previous approaches [19, 101], up to a maximum number of the GA executions, MAXRUN, can be run but it will stop running if the convergence test is satisfied. Additionally in meta-convergence, K+1 slots are maintained where K is the number of optimal solutions expected. Each slot stores the best solution of a different of motif kind. and is allocated a counter Cnt, which keeps track of its motif convergence count. At the end of each GA run, a number (NUM) of best solutions (individuals) will be dispatched and subject to the similarity test to the K+1 slots. The corresponding counter will increment for each update of a solution of the same motif kind and reset if the motif is replaced by a new one. A convergence threshold MAXIND is used to monitor convergence. MAXIND is a relatively small number because each dispatched solution is already a converged one obtained by GA. In general, the meta-convergence framework needs at most MAXRUN GA runs to obtain K optimal solutions while the previous methods such as GAME and GALF-P need K*MAXRUN runs. The whole procedure of meta-convergence is illustrated in Figure 4.2.

Similarity test applied in the framework

Solutions that pass the similarity test, i.e. those belong to the same motif kind in a particular slot, will compete for the same slot based on their fitness. On the other hand, the solution of a new motif will occupy an empty slot or the slot storing the solution with the worst fitness. After each GA run, when a slot is near convergence (we define this situation as Cnt > MAXIND/2), solutions similar to it will be eliminated, again based on the similarity test, to make room for the other optimal solutions in the next GA run. When the solution of a particular motif in the slot has converged (i.e. Cnt $\geq$ MAXIND), the motif will be taken out from the search process, i.e. all the exactly matched TFBSs belonging to this motif will be deleted, making

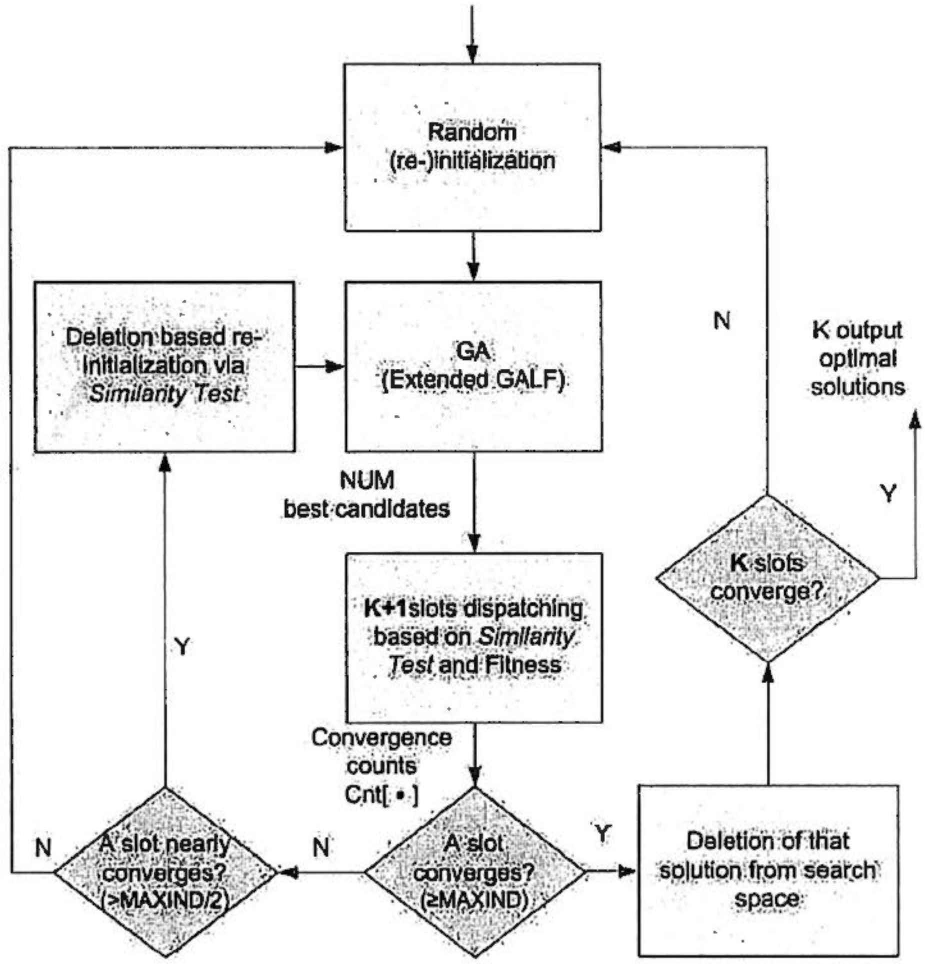


Figure 4.2: The procedure of meta-convergence.

room for efficient discovery of other motifs. The extra $(\mathrm{K}+1)^{th}$ slot is used to keep certain sub-optimal solution in the early stage in order not to lose them, because otherwise the Cnt may fluctuate especially for the K = 1 case when there are several motifs with close fitness competing for the only slot.

4.4.3 GALF-G Implementations

We employ the genetic algorithm (GA) based GALF [19] as the searching procedure. However, since GALF was previously based on simpler assumptions, it has to be extended accordingly to suit the need of the generalized model.

Extended GALF Operators

Local filtering (LF) is the feature operator of GALF, which employs the combined representations for the whole motif (PFM Θ) and individual instances (SIM A). However, it was based on the simple OOPS and fixed-width assumptions. As a result, extensions have to be made for more general cases addressed by GALF-G.

Generally, LF refines each individual (candidate solution) by iteratively scanning the sequence containing the currently worst instance and choosing the best replacement. To evaluate each instance (site) of the individual, the similarity score with the consensus concept is proposed. However, the relation between this heuristic score and the fitness is implicit. In GALF-G, we propose to use the log likelihood ratio for an instance fragment starting at the w_0^{th} column with width w',

$$logp(r_i, w_0, w') = \sum_{j=w_0}^{w_0+w'-1} \log \frac{\Theta_{jr_i(j)}}{\Theta_{0r_i(j)}} \tag{4.11}$$

to evaluate each instance r_i, where $r_i(j) \in \Sigma$ is the nucleotide in column j of r_i, $\Theta_{jr_i(j)}$ is the corresponding frequency from the PFM and $\Theta_{0r_i(j)}$ is the corresponding background frequency. It measures the ratio of r_i generated by the motif PFM over the background, and is more closely related to $\psi(w_i)$ in Equation 4.10. The effectiveness of the log likelihood ratio and the mutation operator are verified (results not included here) on the 8 datasets tested in [101]. In range input cases, with the w_{cor} core fragment stored, we encourage LF to match instances with a longer width ($\geq w_{cor}$) so that the width w' is chosen randomly from $[w_{cor}, w_{max}]$ and thus LF can be applied with fewest modifications.

Because now the fitness f can handle the general case with any motif instances, the new GALF-G can now search based on zero or one occurrence per sequence (ZOOPS) assumption rather than OOPS. However, it is unwise to randomly generate null positions for non-sites at the very beginning during searching. It is because when most of the individuals are poor in their fitness, fewer instances will be strongly biased and the population will suffer from undesirable premature convergence. To alleviate this problem, we initialize the population with OOPS assumption and refine the abundance ratio (p_0 in Equation 4.8) in later generations using a new mode of LF. The convergence (CONVER) mode of LF is triggered when the best individual stagnates for more than 1/4 of the convergence count MAX-CONVER, or when it is toward the maximal generation of the GA. The convergence mode LF is applied to all individuals to adjust the motif abundance. The procedure is similar to normal LF except that the full w_{max} fragment will be chosen for each instance and the worst instances are to be removed rather than refined, if eliminating it makes the overall fitness f increase.

Other Extensions

We adopt the single-point mutation and pre-selection from GALF-P [19] and choose multi-point (close to uniform) crossover instead of single-point because it provides higher diversity. Since the new model adjusts widths automatically, the shift operator in GALF-P [19] is no longer needed.

To handle general cases other than the ZOOPS assumption, where there may be several occurrences in a sequence, we employ a refinement process for additional instances upon the meta-convergence of GALF runs. Generally, if a fixed width is input, instances have to increase f in order to be added, while in the width range case, the threshold of f is relaxed slightly [see Additional file 1 of [20] for the details].

Table 4.1: Pseudo-code of the local filtering (LF) operator

```
Input: An individual I with the collapsed SIM A = {p_1, p_2, ..., p_m}
  where p_i is the site, i.e. position, (may be null) for instance r_i;
  m is the sequence number.
LOCAL_FILTERING(I)
{
  Choose a random w' for NORMAL or CONVER (w_max) mode
  Choose the offset w_0 randomly from [1, w_max - w' + 1]
  Sort all the instances by logp(·, w_0, w') and obtain their
  corresponding sequence ranking: Rnk(1), Rnk(2), ... Rnk(m);
  where logp(r_Rnk(1)) ≥ logp(r_Rnk(2))... ≥ logp(r_Rnk(m))
  // logp of a null instance is set to be -∞
  for (k = m; k ≤ 2, k - -)
  {
    if ( mode == NORMAL ) {
      Scan sequence Rnk(k) to get q_Rnk(k) with best logp;
      p_Rnk(k) = q_Rnk(k);
      if (logp(p_Rnk(k)) ≤ logp(p_Rnk(k-1))) Return I;
    }
    if ( mode == CONVER ) {
      if (f(I - {p_Rnk(k)}) > f(I)))    p_Rnk(k) = NULL;
      else    Return I;
    }
  }
}
```

Combining the meta-convergence framework with extended GALF based on the generalized model, as well as the refinement procedure, we have the proposed GALF-G to discover multiple TFBS motifs. The pseudo-codes of the new LF, the extended GALF and GALF-G are shown in Tables 4.1, 4.2 and 4.3.

4.5 Experiments

In this section, The summary of the experiments is introduced, and then the experimental results are reported and analyzed in corresponding categories. Finally experiments concerning the efficiency of GALF-G are presented.

4.5.1 Experiment Summary

First of all, the evaluation measurements are introduced here. For most experiments except the benchmark ones [37, 87], the

Table 4.2: The extended GALF. INTL is the interval of generations to trigger LF. MAXGEN is the maximal number of generations to run and MAXCONVER is the convergence count.

```
for(i=0; i < MAXGEN; i++)
{
  Evaluation on the population;
  NORMAL mode LF on the population every INTL generations;
  Randomly pair the N individuals into N/2 pairs;
  for(each pair of the individuals)
  {
    Uniform crossover and Single-point mutation;
    Evaluation and Selection within the pair;
  }
  C = the best individual;
  if(C stagnates for ≥ 1/4MAXCONVER)
    CONVER mode LF on the population;
  if(C stagnates for ≥ MAXCONVER)   break;
}
Output NUM best individual(s) C[·];
```

Table 4.3: The framework of GALF-G. MAXGEN and MAXRUN are the maximal generations of GALF and maximal times to run GALF, respectively. MAXIND is the convergence count for best individuals from different runs.

```
Initialize K+1 Slot[·] for K motif types and the counters Cnt[·];
Initialize a random population with N individuals;
for(g=0; g < MAXRUN; g++)
{
  Re-initialize the population accordingly;
  Run the extended GALF;
  C[·] = the NUM best individuals output by GALF; //GALF in Table 4.2
  for(i=0; i < NUM; i++)
  {
    for(j=0; j < K+1; j++)
    {
      if(SimilarityTest(C[i], Slot[j]) is passed)
      {
        Slot[j] = the one with better f between C[i] and Slot[j];
        Cnt[j]++;
        if( Cnt[j] ≥ MAXIND )
          Mark Slot[j] as converged and erase Slot[j];
        break;
      }
    }
    if (C[i] does not suit any existing slot)
    {
      if (An empty slot exists)   Put C[i] to that slot;
      else    C[i] competes with the slot with lowest f;
    }
  }
  if(The K best solutions of the K+1 slots converge)   break;
}
Refinement on Slot[·] and output the best K ones in terms of f.
```

measurements employed are the site level (prefix s) ones: positive predictive value/precision $sPPV$, sensitivity/recall sSn and F-score sF with shift restrictions, similar to [19, 101]. The advantage is that they reflect both site level and part of the nucleotide level performances concisely. For the benchmark experiments, we have to follow their standard measurements which employ looser site level measurements but introduce additional nucleotide level (prefix n) PPV ($nPPV$) and sensitivity (nSn), as well as performance coefficient (PC) [37, 80, 87, 99] and correlation coefficient (CC) [87, 99] on both levels [see Additional file 1 of [20] for details of evaluation measurements for different experiments].

(i) Single motif discovery experiments ($K = 1$) were firstly performed to test the generalized model. GALF-G was verified on the 800 synthetic datasets from [19], and compared with other state-of-the-art algorithms with fixed-width inputs as a special/degenerative case. GALF-G was then further tested on the 8 real datasets employed in GAME [101] with both fixed-width (the assumed true widths from [101]) inputs and range (variable widths) inputs relatively close to the true widths. The challenges raised by the eukaryotic benchmark [87, 99] are then addressed, where there is no dataset-specific prior knowledge on the motif widths and only single motif outputs ($K = 1$) and compared.

(ii) Multiple motifs experiments ($K > 1$) were then performed for two scenarios. In the first scenario, since multiple candidates are desirable for biological testing even for single motif discovery [37], GALF-G was tested and compared with the state-of-the-art algorithms on the 62 *E. coli* benchmark datasets [37], without dataset-specific prior knowledge on the motif widths. In the second scenario, since it is also desirable to discover different real motifs simultaneously, GALF-G, GAME and MEME were tested on the real liver-specific dataset with multiple (overlapping) motifs. Investigating into the exceptional case of GAME's 8 datasets using GALF-G with multiple motifs discovery, we discovered a putative motif not annotated in the dataset previously has been identified.

4.5.2 Parameter Setting

Besides the parameters discussed specifically (such as motif widths and output motif number K), and except the efficiency experiments (with different PS), the other parameter setting exactly follows GALF-P [19] with the purpose of minimum tuning. In the extended GALF: default population size PS: 500; maximal number of generations MAXGEN: 300; interval of generations to trigger local filtering (LF)–INTL: 10; convergence count MAX-CONVER: 50; mutation rate: 0.9; crossover rate: 0.3; and maximal runs of GALF MAXRUN: 20. The quite large population size follows the setting of GAME for fair and consistent comparisons, though it turns out that a smaller population size also works comparably well (in the efficiency experiments).

4.5.3 Single Fixed-width Motif Discovery on Synthetic Data

GALF-G was first verified in the special cases of fixed-width single motif discovery ($K = 1$) on the 800 synthetic datasets used to test GALF-P in [19], which had performed best for these fixed width cases (as shown in the previous chapter).

We compared GALF-G with GALF-P, GAME, MEME, BioProspector (BioPro.), and BioOptimizers based on MEME and BioProspector. Weeder was not compared because it cannot be run on the long-width (16) datasets due to its width limit of 12. Details on generating the datasets were provided in [19]. The average F-scores sF on the site level for each scenario are presented in Table 4.4, with the best results shown in bold. The full

Scenarios Width /Num /Con	GALF-G	GALF-P	GAME	MEME	BioPro.
Short /Small /Low	**0.48 ±0.29**	0.44 ±0.27	0.30 ± 0.30	0.39 ±0.35	0.39 ± 0.31
Short /Large /Low	**0.55 ±0.22**	**0.55 ±0.22**	0.36 ± 0.30	0.42 ±0.29	0.45 ± 0.23
Long /Small /Low	**0.89 ±0.13**	**0.89 ±0.14**	0.82 ± 0.22	0.88 ±0.14	0.83 ± 0.14
Long /Large /Low	**0.91 ±0.06**	**0.91 ±0.05**	0.90 ± 0.07	0.90 ±0.07	0.80 ± 0.11
Short /Small /High	0.84 ±0.07	0.80 ±0.09	0.75 ± 0.23	**0.85 ±0.07**	0.78 ± 0.12
Short /Large /High	**0.86 ±0.04**	0.83 ±0.05	0.83 ± 0.10	0.83 ±0.04	0.76 ± 0.06
Long /Small /High	**0.98 ±0.02**	**0.98 ±0.03**	0.97 ± 0.03	**0.98 ±0.02**	0.97 ± 0.03
Long /Large /High	**0.99 ±0.01**	0.97 ±0.02	0.98 ± 0.01	0.98 ±0.01	0.96 ± 0.02
Average	**0.81**	0.80	0.74	0.78	0.74

Table 4.4: Average site level F-scores for the 800 fixed-width synthetic datasets experiments. $\pm$ indicates the standard deviation (over the 100 datasets generated for each scenario). Width: the motif width, Num: the number of sequences and Con: conservation degree.

Table 4.5: The t-test p-values between GALF-G and MEME for the scenarios according to Table 4.4. [ ] indicates the case when the counterpart is better in the average sF. Those p-values within the significance level 0.05 are shown in bold.

Scenarios	GALF-G better	MEME better
Short /Small /Low	**0.0246**	[0.9754]
Short /Large /Low	**0.0002**	[0.9998]
Long /Small /Low	0.3006	[0.6994]
Long /Large /Low	0.1397	[0.8603]
Short /Small /High	[0.8432]	0.1568
Short /Large /High	**0.0003**	[0.9997]
Long /Small /High	[0.5000]	0.5000
Long /Large /High	**0.0000**	[1.0000]

table with precisions ($sPPV$), recalls (sSn), including BioOptimizer results (almost identical to MEME and BioProspector), is not shown. GALF-G and GALF-P are in general the best among all scenarios, especially in the difficult scenarios (for example, short widths and low conservation). GALF-G is slightly better than GALF-P in the last 4 scenarios. To compare GALF-G with another close competitor, MEME, the two-sample Welch's t-test [102] was employed. The respective p-values of GALF-G better than MEME, and MEME better than GALF-G, with respect to sF for the corresponding scenarios, are shown in Table 4.5.

In 4 of the 6 scenarios where GALF-G shows better average sF (scenarios except 5, 7), GALF-G is better than MEME within the significance level 0.05. On the other hand, MEME shows no convincing significance of being better than GALF-G in the other 2 scenarios.

We do not expect great differences between GALF-G and other algorithms here, because under the fixed-width cases the generalized model is similar to other models in representative power. The experiments demonstrate the search capability of GALF-G is comparable to or better than the previous best GALF-P on the synthetic datasets. The main reason is that they use similar effective searching techniques based on local filtering [19]. The results from the synthetic data can be interpreted intuitively with respect to searching difficulties, because their respective conservation degrees are explicitly generated.

For variable-width (range) cases, the complicated nature of different conservation degrees of TFBSs is not easy to model or evaluate with synthetic data, hence it is more appropriate to test different methods with substantial real datasets, and the experimental results are presented in the following sub-sections.

4.5.4 Single Motif Discovery on Real Datasets

In this sub-section, GALF-G was evaluated and compared with other methods on the 8 real datasets used to test GAME [101], for both fixed and variable widths cases in single motif discovery ($K = 1$). Information of the 8 datasets is described in Table 3.5 from the previous chapter.

The comparison studies for fixed and variable widths cases are given as follows:

(i) Fixed-width single motif discovery ($K = 1$) experiments were performed, where GALF-P was previously tested and compared with GAME in a fixed-width manner. GALF-G shows comparable overall F-scores sF (0.81) to the best average results from GALF-P (0.82) and is better than GAME (0.61) by 33% on average from 20 runs. While GALF-P shows significantly smaller variations than GAME in the performance [19], GALF-G shows even more stable and robust performance than GALF-P, which is discussed further in the Efficiency Experiments.

We have also tried Weeder [78, 79] on part of the datasets because Weeder can only handle widths 6, 8, 10 and 12. Weeder is optimized for several width range modes [79] rather than fixed widths and will be formally compared in the following range experiments. For the fixed-width experiments, only CREB, MyoD, SRF and TBP were tested. The averaged $sPPV$, sSn and sF of Weeder for the 4 datasets are 0.43, 0.63 and 0.51, respectively. On the other hand, GALF-G is better where the corresponding values are 0.79, 0.83 and 0.81.

Similar to the conclusion on fixed-width synthetic experiments, GALF-G demonstrates competitive searching capacity on the fixed-width real data experiments, while GALF-G makes a looser assumption.

(ii) (K = 1) variable-width (range) experiments were performed, where GALF-G was compared with GAME, MEME, Weeder, and FlexModule from CisGenome [42] on the previous 8 real datasets. The additional FlexModule is a Gibbs sampling [51] motif discovery module implemented in the recent integrated system CisGenome [42] for analyzing transcriptional regulation.

For each dataset, 3 different width ranges were input for testing where

$$R_i = [w_{min((i)}, w_{max(i)}] = [w_i - 3, w_i + 3] \ (i = 1, 2, 3). \quad (4.12)$$

Each range represented variations of $\pm$ 3bp on the width w_i while the lower bound for $w_{min((i)}$ was set to 5 because it is rare for a motif width being smaller than 5. With increasing i, $w_i = w_{true} + (i - 1)$ reflects larger divergence of shift from the biological truth w_{true} [See Additional file 1 of [20] for the running parameters].

The average results of executing each program 20 times are shown in Tables 4.6 and 4.7. Weeder is deterministic, and MEME performs constantly in different runs for a same dataset (as contrast to different datasets in Table 4.4), so there are no standard deviations shown for them.

In most cases (19/24) GALF-G achieves the best F-scores sF on the site level, as well as the average $sPPV$, sSn and sF averaged on all the cases. The overall F-score of GALF-G is 19% better than GAME, 14% better than MEME, 85% better than Weeder, and 21% better than FlexModule. The standard deviations of GALF-G are also lower than GAME and FlexModule in most cases. The t-test on sF shows that GALF-G is better than MEME in 20 cases within significance level 0.01, and in 1 case within significance level 0.02, while MEME is better in 3 cases within level 0.01. It should be noted that GALF-G significantly outperforms the other algorithms in sSn, probably because the generalized model not only predicts motifs as precise as the other models, but also accepts more correct TFBSs based on a wider range than single widths.

The above experiments demonstrate that with a range relatively close to the true widths, GALF-G with the generalized model shows favorable performance even compared with the results based on E-values. In fact, the performance with the input width ranges close to the true widths is comparable to that with fixed-width inputs, except for the MyoD dataset. The exceptional case of MyoD will be investigated separately and shown containing multiple motifs later.

To summarize, on the 8 real datasets for single motif discovery, GALF-G demonstrates competitive performance in fixed-width experiments, and provides obvious improvement over other methods in variable-width (range) experiments. For the cases

Datasets	GALF-G			GAME		
	sPPV	sSn	sF	sPPV	sSn	sF
CREB						
R_1	0.76 ± 0.00	0.68 ± 0.00	**0.72 ± 0.00**	0.34 ± 0.37	0.35 ± 0.36	0.34 ± 0.36
R_2	0.75 ± 0.06	0.68 ± 0.04	**0.71 ± 0.05**	0.33 ± 0.34	0.34 ± 0.35	0.33 ± 0.34
R_3	0.76 ± 0.00	0.68 ± 0.00	**0.72 ± 0.00**	0.39 ± 0.36	0.38 ± 0.35	0.38 ± 0.35
CRP						
R_1	0.94 ± 0.00	0.73 ± 0.02	**0.82 ± 0.01**	0.79 ± 0.02	0.78 ± 0.00	0.78 ± 0.01
R_2	0.89 ± 0.02	0.74 ± 0.00	**0.81 ± 0.01**	0.82 ± 0.00	0.78 ± 0.00	0.80 ± 0.00
R_3	0.79 ± 0.06	0.71 ± 0.04	0.75 ± 0.05	0.93 ± 0.03	0.66 ± 0.03	0.77 ± 0.01
ERE						
R_1	0.64 ± 0.02	0.83 ± 0.02	0.72 ± 0.02	0.53 ± 0.00	0.80 ± 0.00	0.63 ± 0.00
R_2	0.67 ± 0.03	0.85 ± 0.03	0.75 ± 0.03	0.55 ± 0.04	0.79 ± 0.02	0.65 ± 0.02
R_3	0.77 ± 0.05	0.84 ± 0.01	**0.80 ± 0.03**	0.60 ± 0.04	0.80 ± 0.03	0.69 ± 0.03
E2F						
R_1	0.79 ± 0.02	0.84 ± 0.03	**0.81 ± 0.02**	0.76 ± 0.09	0.84 ± 0.10	0.80 ± 0.10
R_2	0.79 ± 0.00	0.81 ± 0.00	**0.80 ± 0.00**	0.72 ± 0.00	0.85 ± 0.00	0.78 ± 0.00
R_3	0.79 ± 0.00	0.81 ± 0.00	**0.80 ± 0.00**	0.75 ± 0.00	0.78 ± 0.00	0.76 ± 0.00
MEF2						
R_1	0.93 ± 0.00	0.82 ± 0.00	**0.88 ± 0.00**	0.65 ± 0.29	0.75 ± 0.33	0.69 ± 0.30
R_2	0.94 ± 0.00	1.00 ± 0.00	**0.97 ± 0.00**	0.73 ± 0.26	0.77 ± 0.28	0.75 ± 0.27
R_3	1.00 ± 0.00	1.00 ± 0.00	**1.00 ± 0.00**	0.93 ± 0.00	0.83 ± 0.03	0.88 ± 0.01
MyoD						
R_1	0.33 ± 0.04	0.42 ± 0.05	**0.37 ± 0.04**	0.13 ± 0.10	0.16 ± 0.10	0.14 ± 0.10
R_2	0.21 ± 0.01	0.23 ± 0.02	**0.21 ± 0.05**	0.12 ± 0.11	0.16 ± 0.16	0.11 ± 0.11
R_3	0.25 ± 0.00	0.29 ± 0.00	**0.25 ± 0.06**	0.13 ± 0.12	0.14 ± 0.15	0.13 ± 0.14
SRF						
R_1	0.72 ± 0.04	0.87 ± 0.03	**0.79 ± 0.03**	0.71 ± 0.02	0.87 ± 0.04	0.78 ± 0.03
R_2	0.74 ± 0.03	0.78 ± 0.04	0.76 ± 0.03	0.66 ± 0.02	0.87 ± 0.01	0.75 ± 0.02
R_3	0.70 ± 0.02	0.74 ± 0.08	0.72 ± 0.05	0.70 ± 0.06	0.77 ± 0.05	0.73 ± 0.02
TBP						
R_1	0.86 ± 0.01	0.82 ± 0.02	**0.84 ± 0.01**	0.80 ± 0.08	0.75 ± 0.12	0.77 ± 0.09
R_2	0.87 ± 0.02	0.86 ± 0.02	**0.87 ± 0.01**	0.79 ± 0.05	0.78 ± 0.04	0.78 ± 0.03
R_3	0.87 ± 0.02	0.86 ± 0.02	**0.86 ± 0.02**	0.71 ± 0.17	0.74 ± 0.18	0.72 ± 0.18
Average	0.74	0.75	**0.74**	0.61	0.66	0.62

Table 4.6: Average results (precision ($sPPV$), recall (sSn) and F-scores (sF) are averaged separately) of GALF-G and GAME on the 8 datasets. Each range $R_i = [w + (i-1) - 3, w + (i-1) + 3]$ in general indicates different shifts i from the true width w. ± shows the standard deviation (based on 20 independent runs of each dataset with each range). The results with best sF among this table and Table 4.7 are shown in bold.

Datasets	MEME			Weeder			FlexModule		
	sPPV	sSn	sF	sPPV	sSn	sF	sPPV	sSn	sF
CREB					medium				
R_1	0.73	0.58	0.65	0.44	0.84	0.58	0.68 ± 0.04	0.76 ± 0.04	0.72 ± 0.04
R_2	0.83	0.53	0.65	0.44	0.84	0.58	0.62 ± 0.22	0.69 ± 0.24	0.65 ± 0.23
R_3	0.83	0.53	0.65	0.44	0.84	0.58	0.67 ± 0.07	0.72 ± 0.07	0.69 ± 0.07
CRP					large				
R_1	0.93	0.61	0.74	0.41	0.71	0.52	0.94 ± 0.14	0.55 ± 0.11	0.69 ± 0.12
R_2	0.89	0.70	0.78	0.41	0.71	0.52	0.97 ± 0.07	0.56 ± 0.06	0.70 ± 0.06
R_3	0.89	0.70	**0.78**	0.41	0.71	0.52	0.96 ± 0.13	0.50 ± 0.10	0.65 ± 0.11
ERE					large				
R_1	0.88	0.60	0.71	0.29	0.64	0.40	0.74 ± 0.03	0.85 ± 0.01	**0.79**± 0.02
R_2	0.88	0.60	0.71	0.29	0.64	0.40	0.73 ± 0.02	0.85 ± 0.02	**0.79**± 0.02
R_3	0.88	0.60	0.71	0.29	0.64	0.40	0.68 ± 0.17	0.77 ± 0.24	0.72 ± 0.21
E2F					large				
R_1	0.78	0.67	0.72	0.23	0.93	0.37	0.56 ± 0.28	0.58 ± 0.29	0.57 ± 0.28
R_2	0.83	0.70	0.76	0.23	0.93	0.37	0.60 ± 0.29	0.60 ± 0.29	0.60 ± 0.29
R_3	0.78	0.67	0.72	0.23	0.93	0.37	0.63 ± 0.25	0.62 ± 0.25	0.63 ± 0.25
MEF2					medium				
R_1	0.93	0.82	0.88	0.01	0.06	0.02	0.86 ± 0.02	1.00 ± 0.00	**0.93** ± 0.01
R_2	0.93	0.82	0.88	0.01	0.06	0.02	0.79 ± 0.27	0.90 ± 0.31	0.84 ± 0.29
R_3	0.93	0.82	0.88	0.01	0.06	0.02	0.88 ± 0.02	0.99 ± 0.04	0.93 ± 0.02
MyoD					small				
R_1	0.00	0.00	0.00	0.07	0.10	0.08	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
R_2	0.00	0.00	0.00	0.07	0.10	0.08	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
R_3	0.00	0.00	0.00	0.07	0.10	0.08	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
SRF					large				
R_1	0.65	0.86	0.74	0.54	0.63	0.58	0.64 ± 0.00	0.87 ± 0.02	0.73 ± 0.01
R_2	0.70	0.86	**0.78**	0.54	0.63	0.58	0.63 ± 0.01	0.82 ± 0.05	0.71 ± 0.02
R_3	0.70	0.86	**0.78**	0.54	0.63	0.58	0.64 ± 0.00	0.86 ± 0.01	0.74 ± 0.00
TBP					small				
R_1	0.70	0.67	0.69	0.56	0.90	0.69	0.47 ± 0.32	0.59 ± 0.40	0.53 ± 0.35
R_2	0.70	0.67	0.69	0.56	0.90	0.69	0.41 ± 0.34	0.51 ± 0.42	0.45 ± 0.38
R_3	0.70	0.67	0.69	0.56	0.90	0.69	0.45 ± 0.34	0.55 ± 0.41	0.49 ± 0.37
Average	0.71	0.61	0.65	0.32	0.60	0.40	0.61	0.63	0.61

Table 4.7: Average results of MEME, Weeder and FlexModule in the same comparison experiments described in Table 4.6. Weeder was run with the width mode (small: 6, 8; medium: 6, 8, 10; large 6, 8, 10, 12) that are closest to the ranges R for each dataset.

without much prior information on the exact widths, experiments will be described in the next sub-sections.

4.5.5 Single Motif Discovery Challenges on Eukaryotic Benchmarks

The improved eukaryotic benchmark [87] has thus been employed for being more suitable than the one by Tompa et al [99] to evaluate motif discovery algorithms. The algorithm benchmark suite [87] extracts motifs from TRANSFAC and includes representative eukaryotic species. There are 50 datasets with backgrounds generated by Markov models and 50 with real cis-regulatory region backgrounds. The widths are not given in the benchmark and thus a uniform width range input has to be set for all experiments. The additional evaluation measure corresponding to this benchmark is the nucleotide level correlation coefficient (nCC) [37,87,99].

GALF-G was tested on the corresponding algorithm benchmark suite [87] and compared with MEME and Weeder, the two most widely used algorithms [see Additional file 1 of [20] for the running parameters of GALF-G]. The average results of nSn, $nPPV$, nPC and nCC are shown in Table 4.8. For Markov backgrounds, GALF-G is 31% better than MEME, 214% than Weeder in nPC, and 42% better than MEME, 165% than Weeder in nCC. Similar conclusions can be drawn for the real backgrounds. It should be noted that while MEME and Weeder perform poorly in one of the two backgrounds, GALF-G maintains the competitive performance well in both.

In the improved eukaryotic benchmark [87], which is considered more suitable to test motif discovery algorithms, GALF-G shows superior performance to the widely-used MEME and Weeder, while only top-scored motifs are compared. However, as stated in [99], it is more meaningful in practice to provide mul-

Table 4.8: Average performances (nSn, $nPPV$, nPC and nCC) of GALF-G, MEME and Weeder on the algorithm benchmark suite (50 datasets with Markov backgrounds and 50 with real backgrounds).

Algorithms	Markov				Real			
	nSn	nPPV	nPC	nCC	nSn	nPPV	nPC	nCC
GALF-G	0.117	0.184	0.102	0.138	0.116	0.156	0.095	0.126
MEME	0.115	0.107	0.077	0.097	0.103	0.092	0.063	0.083
Weeder	0.133	0.043	0.032	0.052	0.202	0.071	0.055	0.096

tiple motifs for testing [57] where the experiments are reported as following.

4.5.6 Multiple Motifs Outputs on the *E.coli* Benchmark

In this sub-section, GALF-G was tested, to address a more realistic scenario, where multiple candidate motifs are desired for identifying the true TFBSs in biological research, on the *E. coli* benchmark.

The *E. coli* benchmark ECRDB62A [37] has 62 datasets, on average about 300 bp in the sequence length varying from 86 to 676 bp, 12 sequences per dataset, around 1.85 sites per sequence and the average site width is 22.83 with standard deviation > 10, which indicates very diversified widths.

Specifically, minimal parameter-tuning policy was employed as if the programs were run by a common user with minimum prior knowledge in practice. Results of AlignACE [84], BioProspector [56], MDScan [57], MEME [5], MotifSampler [98] and Weeder [79] were obtained for comparison. A uniform width of 15 was input for those fixed-width algorithms, namely AlignACE, BioProspector, MDScan and MotifSampler. On the other hand, MEME was run with the default setting for widths and the optimal one was chosen automatically within. Weeder was run with the large width mode. For GALF-G, we ran it on the benchmark datasets with both the uniform fixed width 15 and also the widest range accepted for the program of $R = [10, 20]$ with $|R| = 10$ around the central width 15. For all algorithms, 5 motifs were output for detailed comparisons.

We employ the evaluation criteria from [37], namely precision PPV, sensitivity Sn, performance coefficient PC and F-score F, on both nucleotide (prefix n) and site (prefix s) levels (We use the standard notation of PPV instead of the non-standard specificity definition in their work). In the comparisons shown in Table 4.9, the accuracy of the best prediction out of the top 5 scoring predictions is evaluated with respect to nPC. With both fixed-width and range inputs, GALF-G outperforms the other algorithms in all evaluation criteria. For example, GALF-G (15) outperforms the best among the other algorithms by 49% in nPC, 29% in nF, 28% in sPC and 18% in sF. GALF-G (rg), with width range input $[10, 20]$, outperforms the other best algorithms by 46% in nPC, 29% in nF, 25% in sPC and 24% in sF. By comparing the two different input settings for GALF-G we can see that with little sacrifice in other measures (< 0.01 on the nucleotide level and < 0.02 on the site level), the generalized model based on the range (rg) demonstrates improved site level sensitivity, in particular 15% (or 0.082) in sSn compared with GALF-G (15) and 34% (or 0.172) compared with the best among other algorithms.

Besides the best predictions out of the 5 outputs, investigation was also done to analyze the top-scored motifs as well as the rest individually for different algorithms. The statistics in terms of nPC, which reflects both $nPPV$ and nSn, are shown in Table 4.10. As indicated before in [37], the top-scored predictions are not necessarily the best predictions, implying that outputting only a single prediction may not be a good choice in practice or for comparison studies. However, the top-scored predictions from GALF-G are significantly better than the best among the other algorithms, by 30% (w15) and 36% (rg) re-

Algorithms	Nucleotide level (n)				Binding site level (s)			
	nPC	nSn	nPPV	nF	sPC	sSn	sPPV	sF
GALF-G (15)	**0.260**	0.290	**0.309**	0.300	**0.386**	0.538	**0.520**	0.529
GALF-G (rg)	0.254	**0.297**	0.304	**0.301**	0.379	**0.620**	0.502	**0.555**
AlignACE	0.128	0.198	0.152	0.172	0.234	0.355	0.335	0.345
BioProspector	0.174	0.205	0.270	0.233	0.294	0.424	0.374	0.397
MDScan	0.149	0.177	0.230	0.200	0.240	0.328	0.355	0.341
MEME	0.158	0.259	0.199	0.225	0.295	0.461	0.436	0.448
MotifSampler	0.153	0.179	0.237	0.204	0.302	0.331	0.476	0.390
Weeder	0.152	0.162	0.204	0.181	0.307	0.543	0.387	0.452

Table 4.9: Prediction accuracy on the ECRDB62A benchmark of *E. Coli* at nucleotide, binding site levels. GALF-G (15) was run with the fixed width 15 and GALF-G (rg) was run with the range [10, 20]. The best results are bold.

Algorithms	Best	Worst	Mean	STD	Top-scored
GALF-G (15)	0.260	0.094	0.121	0.031	0.169
GALF-G (rg)	0.254	0.080	**0.129**	0.040	**0.177**
AlignACE	0.128	0.029	0.072	0.045	0.083
BioProspector	0.174	0.097	0.124	0.041	0.130
MDScan	0.149	0.068	0.106	0.034	0.099
MEME	0.158	0.002	0.054	0.069	0.116
MotifSampler	0.153	0.010	0.062	0.065	0.069
Weeder	0.152	0.031	0.081	0.106	0.064

Table 4.10: The statistics of the top 5 predictions in terms of nPC on the ECRDB62A benchmark. GALF-G (15) is run with the fixed width 15 and GALF-G (rg) is run with the range [10, 20]. STD is the standard deviation. The best mean and top-scored results are bold.

spectively. We can also see that, for GALF-G, the generalized model based on the range provides better performance than on the fixed width, with respect to both the top-scored and the mean predictions. This implies that the generalized model using ranges is useful when the prior width information is usually not strong in practice.

On this benchmark for multiple motif outputs, GALF-G outperforms other state-of-the-art algorithms considerably. The generalized model exhibits improved sensitivity while maintaining competitive precision, and thus achieves better overall performance on the site level.

4.5.7 Multiple Motif Types in Real Datasets

In gene regulation, TFBSs of different kinds of motifs may appear in the same promoter region. They either work together to regulate the transcription or compete for the TF binding when part of the TFBSs overlap with each other. Thus it is meaningful to discovery multiple TFBS motifs, possibly with overlaps in some of their TFBSs, from a dataset simultaneously. The following experiments tested GALF-G under the corresponding scenario.

The liver-specific dataset

The liver-specific dataset [49] contains 19 sequences, embedded with several major motifs (with 6-19 sites) varying in widths, namely HNF-1, HNF-3, HNF-4 and C/EBP, and some other motifs with fewer sites, such as CRE, BRF-3 and BRF-4 with only one occurrence for each of them. Some TFBSs from different types of motifs overlap with each other in the dataset. For example, a TFBS of HNF-1 (width 15) overlaps with a TFBS of HNF-4 (width 12) with 7 bp in a particular sequence, while co-occurring TFBSs of HNF-1 and HNF-4 in some other sequences do not overlap at all. The total number of (overlapping) TFBS instances is 60. The widths vary dramatically from 7bp to 31bp.

On this dataset, GALF-G, GAME and MEME were compared using the width range input $R = [8, 16]$, which is considered a common range for TFBSs, to discover different types of motifs. The expected width for GAME was 12, the mean of the input range. Different numbers of motifs, K, ranging from 5 to 20 with step 5, were output and evaluated.

The site level (with shift restrictions) results of $sPPV$, sSn and F-scores sF (with shift restrictions) based on all TFBSs are shown in Figure 4.3 for different K. MEME fails to produce comparable recalls or F-scores to the others. It is probably

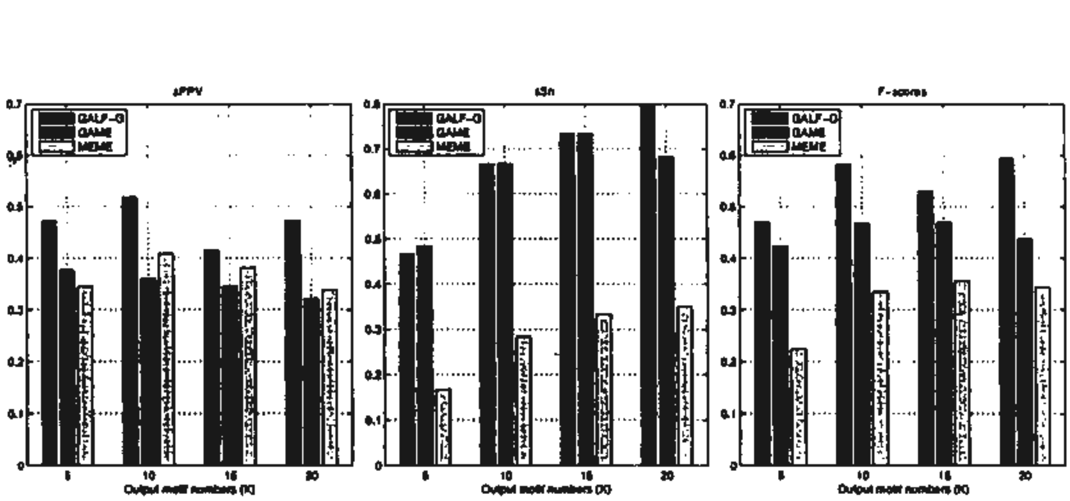


Figure 4.3: The results of precision ($sPPV$), recall (sSn) and F-scores (sF) with shift restrictions for different number of output motifs ($K = 5, 10, 15, 20$) on the liver-specific dataset.

caused by the masking techniques not allowing overlapping of motifs. GAME masks TFBSs individually rather than the whole motifs, so better sSn (recall) can be obtained from a diverse GA population. With overlapping control on the GA, GALF-G shows recalls comparable to or better than GAME. Moreover, GALF-G has the best $sPPV$ (precision) while GAME generally has the worst. Both GALF-G and MEME show an increasing trend of recalls as K increases. The sudden drop of GAME for $K = 20$ is probably because the expected width no longer suits some of the motifs while GAME actually performs fixed-width search in its GA. GALF-G provides the best balance between precisions and sensitivities, and thus gives the best F-scores in all cases. Averaged on all K, the F-scores are: GALF-G: 0.54, GAME: 0.45 and MEME: 0.31 where GALF-G outperforms the other two by 20% and 73% respectively.

Besides the previous evaluation that treats all the TFBSs as a whole, type specific investigation was also carried out on the output results of GALF-G. With the help of STAMP [67], the predicted motifs with $K = 5$ GALF-G were searched for matches of annotated TFBS motifs from the TRANSFAC database V11.3, based on ALLR (Average Log Likelihood Ratio). ALLR was considered to be the most effective in comparisons of single columns for motifs [67].

The relevant matches for the top 2 motifs are displayed in Sequence Logo formats in Figure 4.4. The top 2 high-scored motifs, labeled in STAMP by Motif (width: 13) and Motif v2 (width: 11), match HNF-1 and HNF-4 in TRANSFAC respectively with high statistical significance, i.e., low E-values ($<$ 0.05). For Motif v4 (width: 16), it matches part of HNF-3 alpha without high statistical significance (E-value 2.71e-01), because only part of the HNF-3 TFBSs are identified in the predicted motif. It indicates that, top-scored motifs output by GALF-G in general match true TFBS motifs with high confidence. The other two motifs do not have relevant top 10 matches in TRANSFAC. C/EBP cannot be discovered as a whole motif, possibly due to its low conservation compared to the HNF motifs. STAMP also provides the phylogenetic profile where Motif (HNF-1) and Motif v2 (HNF-4) are grouped together, and so is Motif v4 (HNF-3), implying they belong to the same HNF family. For $K = 10$, similar results are obtained, with matches mainly including HNF-1 and HNF-4.

In-depth investigation on the MyoD dataset

The MyoD dataset seems to be an exceptional case among the 8 real datasets tested by GAME [101]. Only GALF-G ($sPPV$: 19/22, sSn: 19/21, sF : 0.88) and GALF-P ($sPPV$: 21/37, sSn: 21/21, sF : 0.72) are able to show acceptable site level results (with shift restrictions) in the fixed-width ($w = 6$) experiments, while in the variable width experiments none of the programs succeed in providing good results.

To investigate into this exception, GALF-G was set to output $K = 3$ different motifs with the annotated width 6. Besides the fittest output being the annotated MyoD motif, the other two

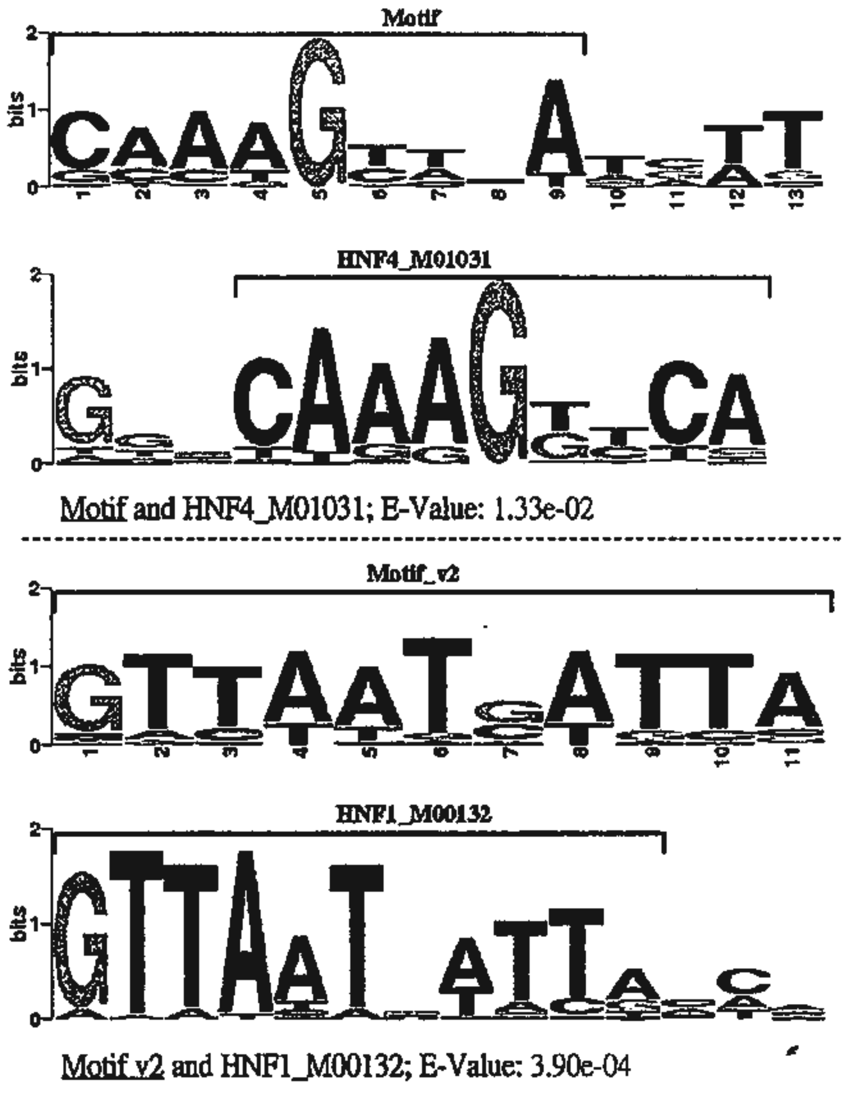


Figure 4.4: The matches from TRANSFAC for the top 2 high-scored motifs. The red brackets indicate the aligned blocks.

are only marginally lower in their fitness compared to the best one (differences $< 2\%$). That is probably the reason why most existing algorithms perform poorly in this dataset – they either locate a sub-optimal because of the low signal-to-noise ratio, or obtain inappropriate rankings of the motifs due to the subtle differences in the modelling. It indicates that the accurate width information is still crucial for such subtle and short motifs.

We searched the 2nd ranked motif, Motif v2, for matches from the TRANSFAC Database using STAMP, based on the various column comparison metrics provided by STAMP. Consistent matches, such as E2A [3,11], p53 [105,106], E47 [50] and E-box [71] motifs, were obtained with high rankings (within top 10s), and these motifs are closely related to MyoD for muscle cell regulation according to the references [3,11,50,71,105,106]. The most consistent matches are shown in Figure 4.5. Thus there is a high probability that Motif v2 is a true motif which may not have been annotated previously in the MyoD dataset.

In summary, GALF-G outperforms GAME and MEME by 14% and 73% on average in sF respectively on the liver-specific dataset for multiple motifs discovery. Additionally, GALF-G sheds light to an additional motif which may not have been annotated previously in the MyoD dataset.

4.5.8 Efficiency Experiments

Although effectiveness is the major concern for motif discovery, practitioners also prefer efficient algorithms which have capability for large scale data. In this sub-section, we tested GALF-G with different GA population sizes to investigate the trade-off between effectiveness and efficiency of meta-convergence.

Firstly, different population sizes ($PS = 500$ (default: In the previous work, in order to be consistent with GAME's $PS =$ 500, GALF-P employed the same setting as default, and this

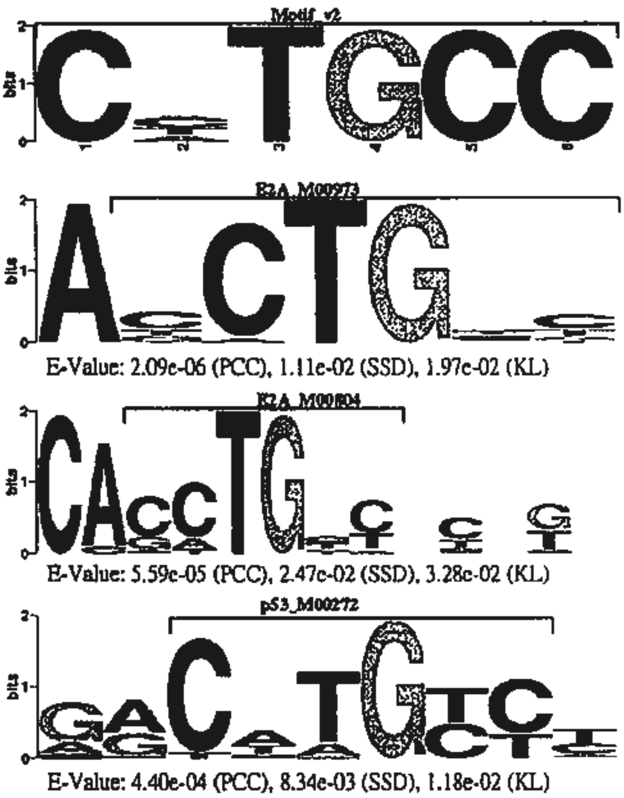


Figure 4.5: The matches from TRANSFAC to the 2nd motif output by GALF-G on the MyoD dataset. The red brackets indicate the aligned blocks.

is followed in GALF-G for the minimum parameter-tuning purpose), 200, 100, 50, 10) were used to run GALF-G, GALF-P and GAME (results from [19]) on the 8 real datasets [101] for fixed-width single motif discovery. For each PS, they were run 20 times on the same Pentium D 3.00GHz machine with 1GB memory, running Windows XP, and the results were averaged. The effectiveness (site F-scores sF) and efficiency are shown in Figures 4.6 (a) to (c).

For the default $PS = 500$, the average time (in seconds) follows that: GALF-G (43.38) $<$ GALF-P (61.91) $<$ GAME (291.11). Since the standard deviation of GAME's effectiveness is already large with $PS = 500$, we only focus on GALF-G and GALF-P to compare the effects (except the special MyoD case better to run with $K > 1$) of different PS. In Figure 4.6 (a),

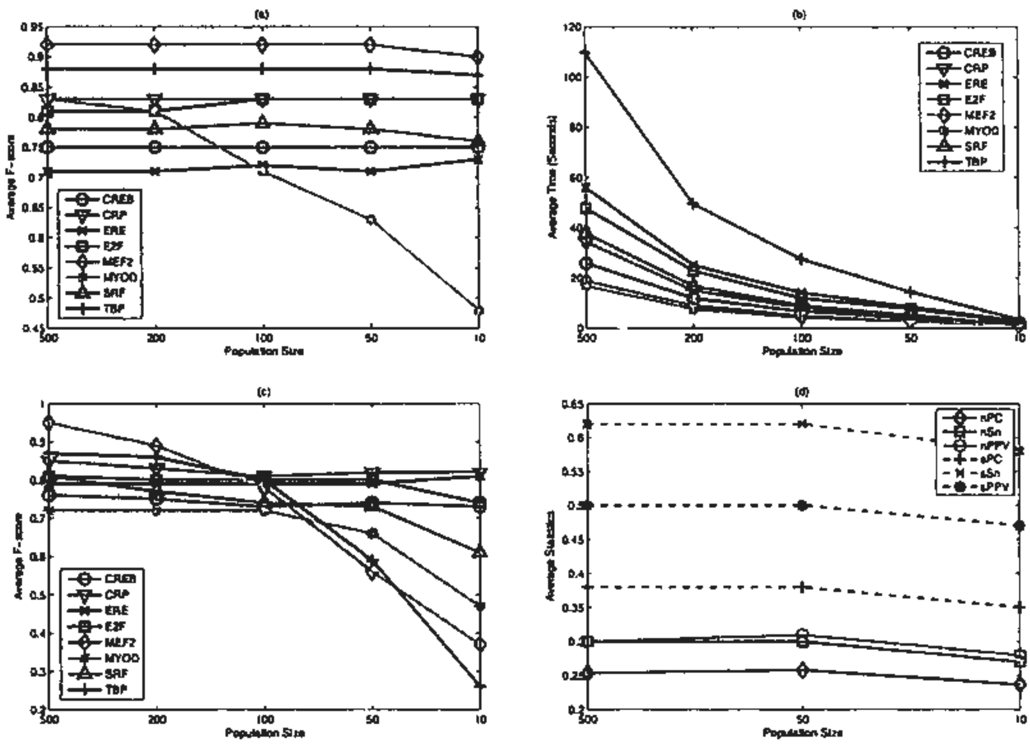


Figure 4.6: Different population sizes: (a) The average site level F-scores sF of GALF-G on the 8 real datasets with fixed width inputs. (b) The average time of GALF-G according to (a). (c) The average F-scores of GALF-P on the 8 real datasets with fixed width inputs. (d) The statistics on both nucleotide and site levels on ECRDB62A of GALF-G with range inputs.

the overall performance for $PS = 500$ are similar, as well as the standard deviations: GALF-G 0.004; GALF-P 0.029. However, when the population size drops to $PS = 10$, the performance of GALF-P drops significantly, and the standard deviation becomes 0.17 on average, and even ≥ 0.40 for MEF2 and TBP datasets (Figure 4.6 (c)). On the contrary, the average performance of GALF-G is maintained, and the overall standard deviation is only 0.031, still a very small number. Furthermore, the average time of GALF-G for $PS = 10$ is just 1.80 seconds, which is over 24 times speedup of the default PS, as shown in Figure 4.6 (b).

It is interesting that even with a population size of 10, GALF-

G still performs comparably well, while GALF-P degenerates significantly. The major reason is due to the meta-convergence framework with similarity test, which is not used in GALF-P. With an extremely small population, GALF may not be able to provide the optimal motif in every run. However, since different motifs are controlled and maintained on a meta level in GALF-G, converged sub-optimal motifs will be replaced by better ones and eventually the global optimum can be found.

The above results imply that, GALF-G is able to provide comparable and consistent performance for fixed-width single motif discovery with a small population for competitive efficiency.

On the *E. coli* benchmark for multiple outputs ($K = 5$) with range inputs, we observed similar performance maintenance with different PS for GALF-G in Figure 4.6 (d), thanks to the meta-convergence mechanism to maintain different optimal motifs in the solutions. The average time on each dataset for the three PS is 655.80 (500), 74.40 (50) and 16.05 (10) seconds respectively, where the $PS = 10$ demonstrates a speedup of over 40 times compared to that of the default size ($PS = 500$). For $PS = 10$, the standard deviation of nPC is 0.0098, which is still small compared with 0.0070 for the default PS.

According to the efficiency experiments, GALF-G is able to maintain competitive effectiveness with very high efficiency. Therefore GALF-G has great potential to work on ever larger scale datasets successfully.

4.6 Discussion and Conclusion

To conclude, we summarize the proposed work of GALF-G, discuss about the challenges and point out future directions.

4.6.1 Summary

In this chapter, the generalized motif model is proposed for realistic motif discovery problems. It models a possible range of widths rather than any single width. The model has the potential to address the biological uncertainty better and is more practical in reality because TFBSs of the same motif may vary in widths and exhibit different degrees of conservation. The meta-convergence framework is proposed to support multiple and possibly overlapping optimal motifs, based on the flexible and easy control of the similarity test for users. GALF-G is developed by incorporating the extended GALF searching methodology into the meta-convergence framework based on the generalized model.

GALF-G has been tested extensively on over 970 datasets, including 800 synthetic datasets, 8 real datasets (further 24 range cases), 100 eukaryotic and 62 *E. coli* benchmark datasets, as well as a real liver-specific dataset with multiple overlapping motifs. GALF-G has shown its competitiveness and better effectiveness for different kinds of motif discovery problems with both fixed-width and range inputs. The generalized model not only predicts the motifs accurately but also include more correct TFBSs. The searching capacity for optimal solutions and efficiency of the meta-convergence framework have also been demonstrated with the synthetic and real datasets. GALF-G has also discovered an additional motif which might not have been annotated previously in the MyoD dataset.

4.6.2 Discussion

However, the motif discovery problem remains challenging due to the weak underlying motif signals input data, as well as the diversity and complexity of TF binding TFBSs [4]. There are also a number of potential improvements for the generalized motif model and GALF-G in our future work, such as further analysis on the effect of different width ranges, more efficient evaluation when handling different width fragments, flexible width distributions for different motif types, validation of the putative motif in MyoD dataset, etc. The candidate fixed-width model for the generalized model still needs more investigation to better suit the biological observation. Integrating the generalized model for motif discovery with additional evidence such as expression data to increase the prediction power is another attractive research direction to us.

□ **End of chapter.**

Chapter 5

Generic Spaced TFBS Motif Discovery with GASMEN

Summary

GASMEN based on GAs is presented for spaced TFBS motif discovery, as a generic extension for the previous contiguous (monad) motif discovery.

5.1 Introduction

In the previous two chapters, we have discussed the GALF algorithms for TFBS motif discovery with the assumption that motifs appear to be contiguous conserved blocks. In this chapter, we address the more complicated case of generic spaced motif discovery, where there can be arbitrary non-conserved spacers (wild card portions) within a TFBS motif. In this section, spaced motif discovery is first introduced, followed by the brief survey of existing methods and GA for motif discovery. Finally the chapter outline is presented.

5.1.1 Spaced Motif Discovery

During TF-TFBS binding, the DNA binding domains of a TF can recognize and bind to a collection of similar TFBSs, from which a conserved pattern called *motif* can be obtained. The DNA segments ("binding cores") that directly interact with binding domains are more specific and thus more conserved, while conservation is not as critical in the portions between binding cores (the so-called gaps or spacers). There are a number of real spaced motifs [103]. Moreover, multiple TF binding, a common machinery in eukaryotes, also results in longer composite motifs with gaps.

5.1.2 Motivations

Existing consensus and matrix (position weight matrix PWM) representations are proposed for monad (contiguous) motifs, so they may not capture the complex spaced motifs well because gaps reduce the total motif scores when evaluated by functions for contiguous motifs.

On the other hand, current algorithms designed for spaced motifs have certain constraints on the gaps. They either restrict all gaps in a motif to be the same and fixed, or restrict the gap number to be 1 and for dyads only (i.e. two monad motifs separated by 1 gap) [56, 92]. Some of them only accept fixed motif widths and specified gaps [56]. Other methods such as MITRA [26] first discover monad motifs and then combine them for possible dyads [70]. Beyond the methods for dyads, recently SPACE [103] is proposed to employ frequent itemset mining techniques to discover generic spaced motifs, with flexible gap numbers and ranges. SPACE is shown to outperform the other spaced motif algorithms [26, 56] on various real and benchmark datasets. However, because the complexity of frequent itemset mining is unbounded, constraints are imposed in SPACE: all candidate motifs are restricted to be derived exactly from the input data occurrences. As a result, SPACE may not be able to capture short monad motifs (as shown in the experimental results). Multiple values of the minimal conserved percentage and the number of occurrences have to be provided beforehand carefully and cannot be too small. The computational time can still be overwhelmingly long to finish the exponential frequent itemset mining.

Because TFBS motifs are often degenerate, search or optimization is difficult (NP-hard [53]). Evolutionary computation has shown great success and potential in motif discovery, in particular with GA [19, 20, 60, 77, 95, 100, 101]. GA maintains a population of candidate motifs called individuals, and optimizes them iteratively through generations. Various genetic operators (e.g. mutations and crossovers) are applied to generate offspring (new candidates) from the parents (previous population). According to the schemata theory, by selection based on the evaluation function, the fit schemata will gradually dominate and the fittest (optimal) individuals will remain. However, previous GA methods are mostly applied on discovering only monad motifs (e.g. our previous work [19,20]), with few studies on even dyads. Furthermore, the input motif widths are either fixed [19, 77, 95] or restricted in certain small ranges [20, 100, 101]. Thus it is desired to apply novel GA on generic spaced motif discovery with flexible width ranges.

5.1.3 Chapter Outline

In this chapter, we propose a novel GA to discover generic spaced motifs, which searches a wide range of possible widths (4-25) and relaxes substantial constraints of the previous methods. The detailed method is elaborated in Section 5.2. Experimental results on various real datasets are reported in Section 5.3. Concluding remarks and future work are available in Section 5.4.

5.2 Methods

In this section, the definitions for generic spaced motifs are first introduced, and then details of the proposed GA are presented.

5.2.1 Spaced Motif Formulations

We follow the definitions employed by SPACE [103] for generic spaced motifs, with a number of relaxed constraints. A **spaced motif** (or simply a motif) M is a width W ($= 25$ to adopt longest possible motifs) string formed by characters of {A, C, G, T, n}, where each maximal substring of consecutive "n" represents a gap (or spacer) and each maximal substring of other characters represents a **conserved segment** ("binding core"). The width for any conserved segment should be $\geq w$ for a predefined minimal width w, and any w**-segment** without "n" is called M's **submotif**. Different from SPACE, no predefined (and relatively large) coverage ratio r is required for segment percentage in our definition, only a minimum coverage number $c = 4$ of non-n characters is set to guarantee a non-trivial biological motif. With this flexible setting users need not worry about choosing r (multiple values are tried in SPACE) and the definition covers more general motifs, especially for short motifs as shown later. The effective spaced motif is thus the substring of M with "n" from the two ends eliminated, and as a result it covers a sufficient range of widths ($4 - 25$) for real biological DNA motifs.

Consider a width-W spaced motif M and any width-W string O formed by characters of {A, C, G, T} from the input sequences. O is called an occurrence of M if, for every submotif (sliding window with width w) $M[i, ..., i{+}w{-}1] \in \{A, C, G, T\}^w$,

$I[i, ..., i+w-1]$ is at most d hamming distance H, i.e. $H(M[i, ..., i+w-1), I[i, ..., i+w-1]) \leq d$. Note that gaps ("n") are not considered as mismatches because they are not in any submotif by definition. In practice, we require the minimal occurrence number $q = 4$ to form a valid motif rather than trying different pre-defined occurrence number thresholds [103], because an appropriate evaluation function can automatically suppress poor motifs with few occurrences.

The following example illustrates the 5 occurrences for a given motif M with $W = 25$, $w = 4$ and $d = 1$, where the effective motif is with width 18 (ignoring the "n"s at the end). For example, CAGT (0), AGT<u>T</u> (1), GT<u>T</u>A (1) from occurrence $O1$ are all within $H \leq d = 1$ (H shown in brackets) from the corresponding submotifs, so they are valid. On the other hand, GTGTCA... is not valid because $H = 2 > d$ between <u>GT</u>GT and submotif CAGT. The example also implies that the consensus of submotifs may not exist in any of the occurrences and submotifs from one motif may only match segments from different occurrences exactly. Therefore, it would be restrictive to generate motif candidates only from occurrences in the input data through replacing characters to "n" in the previous method [103]. In our proposed methods shown later, different submotifs are able to be extracted from different occurrences according to the natural definition of spaced motifs, and thus the previous contraints are relaxed.

- `M=CAGTCAnnACGTnGACGTnnnnnnn`
- `O1=CAGTTAccACGTcGACCTgcgcgcg`
- `O2=CAGACAggACGTgCAGGTcgctata`
- `O3=CACTCAttATGTaGACGTatagcgc`
- `O4=GAGTCAttATGTtGACCTtttatat`

- `O5=CTGTCTggACGTgGTCGTtaactct`

5.2.2 Proposed GASMEN

With the relaxed constraints, it is even more challenging to discover generic spaced motifs effectively and efficiently. To tackle the challenge, we propose the novel **GA** for **S**paced **M**otif **E**licitation on **N**ucleotides (GASMEN). GASMEN employs submotif indexing to partition the search space into smaller subspace, making it easier for the GA to reach optimality. Multiple-motif control and motif refinements are proposed to avoid redundant computation and improve motif quality efficiently. The details of GASMEN are presented as follows.

Submotif Indexing and Initial Population

Submotif Indexing: The relaxed generic spaced motifs impose a huge pattern space compared with the previous method [103]. Although direct optimization using GA is possible, it is more probable for GA to achieve optimality through partitioning the space into smaller sub-space. Given certain w and d, all submotifs $M^{(i)} \in \{A, C, G, T\}^w$ are enumerated and the input sequences are scanned and indexed (with sequence numbers and positions) for each $M^{(i)}$, where for any substring $I^{(i)(j)}$ from the indexed set $I^{(i)}$ of $M^{(i)}$, $H(I^{(i)(j)}[1, 2, ..., w], M^{(i)}[1, 2, ..., w]) \leq d$. Suppose $w = 4$ and $d = 1$, all substring occurrences with Hamming distance $H \leq 1$ from submotif AAAA are indexed accordingly, e.g. substrings starting with A<u>T</u>AA, AAA<u>C</u>, <u>G</u>AAA, etc. Then the procedure is repeated for AAAC, AAAG, AAAT, ..., TTTG, TTTT. For a particular index I_i, GA is applied and it only needs to optimize spaced motifs with all possible occurrences indexed by I_i.

Initial Population: To generate candidate motifs for the GA population, two initialization methods are hybridized to cater for both monad and spaced motifs. One half of the population is generated using the **monad approach**: given submotif $M^{(i)}$, a substring indexed in $I^{(i)}$ is selected randomly, with width w' randomly chosen from $[w, W]$. Note that in the example AACAGTACCA, only the substring within $[w+1, w']$ is used, because the current index $I^{(i)}$ is fixed for submotif AAAA, and motifs starting with AACA will be handled in the other index. The other half of the population is generated using the **spaced approach**: given submotif $M^{(i)}$, the following part beyond $M^{(i)}$ (from $w+1$ to W) of a candidate motif is initialized with "n", and then is assigned with w segments randomly with probability $1/2 * c/w$, where c is the minimal non-n coverage defined previously. Then for each conserved segment (maximal substring of non-n characters), we randomly select a substring indexed in $I^{(i)}$ and fill the segment with the corresponding part of the substring. Thus the candidate motif is with conserved segments (guaranteed to be valid $\geq w$) from different possible occurrences rather than from a single occurrence. Genetic operators will add further variations to both monad and spaced candidate motifs to cover more complete pattern space. Figure 5.1 shows the population initialization approaches.

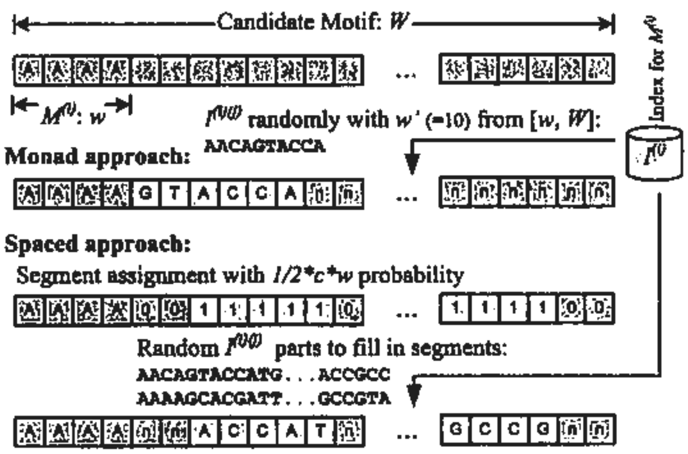


Figure 5.1: Population initialization: monad and spaced motif approaches

Genetic Operators

To generate offspring from the current generation, mutations and crossovers are applied. The genetic operators have to be designed such that they will not produce any invalid candidate motifs, in particular for the conserved segments ($\geq w$). They are illustrated in Figure 5.2 and detailed below.

Mutation: A mutation point p is selected randomly from $[w+1, W]$ (submotif index $M^{(i)}$ is not affected because such indices are enumerated for optimization; see Table 5.1), if p is within a conserved segment, a character is selected randomly from $\{A, C, G, T\}$ to change the motif. If p is within a gap, its nearest next conserved segment is obtained, and we change the segment end to be "n" if the segment $> w$, otherwise do the same mutation within the segment.

Crossover: A crossover point p is selected randomly for both the candidate motifs $P1$ and $P2$ as parents, if p is within a gap for both $P1$ and $P2$, they can be swapped for the parts split by p without violating the definitions. Otherwise the segment where p is located for either $P1$ or $P2$ is obtained and the whole segment is copied to the other parent (in such a case the parent offering the segment is not changed).

Probabilistic Refinement (memetic operator): To directly improve individual fitness for efficiency, probability refinements are applied every 10 generations, adopting the idea of combining consensus-based and matrix-based representations [19]. In the refinement, a PWM (Position Weight Matrix) is generated from the occurrences of each candidate motif, and for each position we change each non-n character, or each "n" neighboring a conserved segment, with probability of its frequency in the PWM, and accepts the variation if the resultant motif is evaluated to be fitter than the original one. The operator is designed for the situation that, because w-submotif has the flexibility of Hamming distance d from the w-segments of the

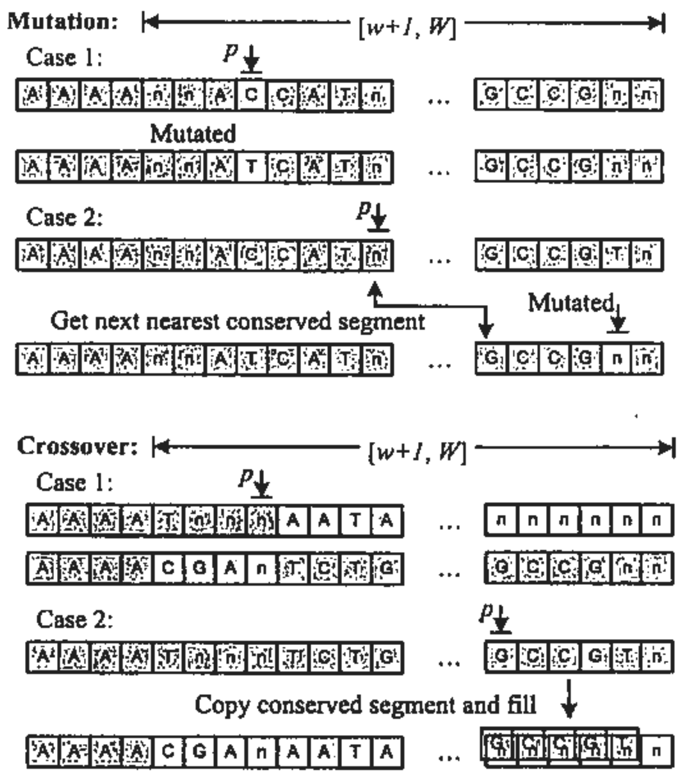


Figure 5.2: Genetic operators: mutation and crossover

occurrences, it may happen that all occurrences are conserved to certain nucleotide (e.g. 90% G) at one position while the submotif gets the wrong nucleotide, e.g. A. In such cases, the probabilistic refinement operator is able to revise the submotif in accordance with its occurrence probabilities (frequencies).

Evaluation Function

The evaluation function is based on the scoring techniques to compute the significance of candidate motifs in Weeder [79] and SPACE [103]. The basic concept is 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 the background and (2) the pattern is either very conserved or occurs in quite a number of the input sequences [103]. As a result, two scores β and σ are computed for the two purposes respectively.

(1) Let M be a candidate motif, and $Occ(M, e)$ be the total occurrences of M as defined in Section 5.2.1, where e is the largest Hamming distance of the occurrences from M. Define N as the total number of characters (nucleotides from $\{A, C, G, T\}$) in the input sequences. The frequency of M in the input sequences is thus $Occ(M, e)/N$. Let $E(M, e)$ (calculation shown later) be the expected frequency of M with at most Hamming distance e from a set of background sequences. $\beta(M)$ is defined as the log relative frequency ratio between M and the background:

$$\beta(M) = \log \frac{Occ(M, e)}{E(M, e) * N}. \tag{5.1}$$

(2) Assuming the input sequences ($\{S_i\}$) are independent, for a candidate motif M, we consider the most conserved occurrence of M in each sequence, and let e_i be the Hamming distance of this best occurrence. Naturally $e_i \leq e$. Thus $1/N(S_i)$ represents the frequency of the best occurrence in sequence S_i, where

$N_(S_i)$ is its total count of characters. Thus the log relative frequency ratio $\sigma(M)$ between all best occurrences of M and the background is defined as:

$$\sigma(M) = \sum_i \log \frac{1}{E(M, e_i) * N(S_i)}. \quad (5.2)$$

If the pattern is very conserved and/or occurs in many sequences, $\sigma(M)$ is large. The final evaluation function is thus $f = \beta(M) + \sigma(M)$. Note that the evaluation function is suitable for both monad and space motifs.

$E(M, e)$ from the background is originally computed by summing the expected frequency $E(M')$ of M' in the background sequences for all M' with at most Hamming distance e from M [103]. However, since the nucleotides for core bindings are specific in conserved segments of real biological motifs, the mismatches of motif occurrences are likely to be restricted in a few positions rather than every possible position in a conserved segment. To capture this property, in the summing procedure we only consider all M' having the same possible error positions as the occurrences from a motif M. Thus the calculation can capture the motif conservation more accurately. Similar to previous methods [79,103], when M' contains gaps, $E(M')$ equals the sum of $E(M'')$ among all possible M'' with all the "n" replaced by A, C, G, and T.

For the background statistics, we adopt the same pre-computed k-mer ($k = 8$) background expected frequencies ($E(M')$) of various species as used in both Weeder [79] and SPACE [103]. When M' is of width longer than k, we calculate $E(M')$ using a $k-1$th order Markov chain. Suppose $M' = p_1 p_2 ... p_{k'}$ with $k' > k$,

$$E(M') = E(p_1 p_2 ... p_k) \prod_{i=k+1}^{k'} P(p_i | p_{i-k+1} ... p_{i-1}) \quad (5.3)$$

where the conditional probability is

$$P(p_i|p_{i-k+1}...p_{i-1}) = \frac{P(p_{i-k+1}...p_i)}{P(p_{i-k+1}...p_{i-1}n)}. \tag{5.4}$$

The GA Procedure with Multiple-Motif Control

Probabilistic crowding [66] is employed in GASMEN to maintain diversity. Crowding has been employed and demonstrated to be more helpful than canonical selection methods in previous work [19], because the optimal motifs lie in a huge and complicated search space. In each generation, individuals are randomly paired to form parent couples P, and each couple competes locally with its own offspring C generated with genetic operators applied, according to higher similarity and better fitness. In GASMEN, a parent and its offspring are paired if they have smaller Hamming distance $H(P, C)$ than the other possible pairing. The competing individual survives with probability proportional to its fitness.

In the problem of motif discovery, multiple candidate motif outputs are desired for practical verifications. We employ multiple-motif control mechanism similar to that used in [20]. Because multiple motifs are considered different from each other in a certain degree, suppose n is the number of output motifs (solutions), a user-defined parameter α is set to control the difference percentage threshold between various candidate motifs. In GASMEN, n solutions are allocated for each w and each $M^{(i)}$. In each generation, every individual tries to get in one of the corresponding n solutions, subject to two criteria: (1) it is different ($> \alpha$) from all existing solutions and its fitness is better than the worst one in the n solutions; or (2) it is similar ($\leq \alpha$) to certain solution(s) and its fitness is better than all of them. In the latter case, all other similar solutions will be eliminated to make sure all n solutions are different with percentage $> \alpha$. To test the difference/similarity of two motifs, we employ the Hamming distance H again, but this time the two motif patterns will be aligned without gaps to check whether one is a shifted version of the other. If in an alignment $H/W' \leq \alpha$, where W' is the shorter effective width between the two motifs, they are considered similar and vice versa. Note that all characters including "n" are at $H = 1$ from empty positions made by shifting. With multiple-motif control, multiple and diverse potential motifs are well preserved through generations, with various submotifs and different w.

In the whole GA procedure, there are several w values, and a number of submotifs $M^{(i)}$ given each w, we use them as prefixes to denote solutions at a certain hierarchy, e.g. w-$M^{(i)}$-solutions. When all w-solutions are obtained, a cross-linking procedure is applied. Each w-solution is assigned different w's and the new w will be accepted if the fitness increases, provided the motif is still valid. Cross-linking prevents sub-optimal solutions with an inappropriate w for the same motif pattern. The whole GASMEN approach is illustrated in Table 5.1.

5.3 Experimental Results

In this section, the experiment settings and comparisons on real datasets are reported. GASMEN is first compared with SPACE on 2 representative spaced motif datasets, and then compared with Weeder and SPACE on 8 real benchmark datasets for general motif discovery.

5.3.1 Experiment Settings

In all the comparison experiments, GASMEN was set with $W = 25$, $w = 4, 5$, $d = 1$, $n = 5$ and $\alpha = 0.2$. For GA, population size was 100, mutation rate was 0.5 (to push for more explo-

Table 5.1: The pseudo-code of GASMEN

Motif width W, submotif width w, distance d,
motif number n, difference threshold α

```
for each w {
  Submotif indexing
  for each submotif M^(i) {
    Population Initialization of all w-M^(i) candidate motifs
    Evaluation (f = β + σ) on the population
    for each generation g {
      Perform Probabilistic Refinement if g%10==0
      Random Pairing to form parent couples
      for each parent couple {
        Generate offspring C from P with Crossovers
        Mutations on C based on mutation rate=0.5)
        Pair P and C based on minimal H(P,C)
        Probabilistic Selection between the competing P and C
      }
      Fill in w-M^(i)-solutions with Multiple-Motif Control
      Check Convergence
    }
  }
  Fill in w-solutions with w-M^(i)-solutions (Multiple-Motif Control)
}
Refine all w-solutions with Cross-Linking different w
Fill in the n final solutions with all w-solutions (Multiple-Motif Control)
```

ration in the huge search space), generation number $g = 100$, and convergence count was 10. The constants were $q = 4$ (minimal occurrence number) and $c = 4$ (minimal non-n character number) respectively. As a result, GASMEN searched a very wide width range of 4-25, which covers most possible biological motifs on nucleotides.

GASMEN was compared with two representative algorithms, SPACE [103] and Weeder [79], which are state-of-the-art algorithms for spaced motif discovery and consensus-based monad motif discovery respectively. All three methods are able to search wide width ranges rather than requiring specified widths [19, 56] or small width ranges [20, 100, 101]. They also share similar background models for clear comparisons on the performance. Both SPACE and Weeder are designed to run with multiple settings (e.g. different W, q, c) and vote for the final output motifs. We employed the "large" mode of Weeder to cover the widest supported width ranges of 6-12 (unfortunately Weeder cannot support longer widths). SPACE was run with both default ($w = 5$, $c = 0.5, 0.8$, $q = 0.5, 1.0$, $W = 8, 15, 20$) and the paper [103] settings ($w = 5$, $c = 0.5, 0.8$, $q = 0.5, 0.9$, $W = 10, 15$). For each dataset, all three algorithms were run with the corresponding species background. Other parameters were kept default.

5.3.2 Comparisons on Spaced Motifs

In this section, we compare GASMEN with the state-of-the-art method, SPACE [103], for generic spaced motif discovery preliminarily. The known representative LexA ($W = 20$) [22] and PurR ($W = 16$) [16] motifs from *E. coli* are collected, where both of them have the characteristics of spaced motifs. LexA is the very example used in the Sequence Logo website [22]. Both GASMEN and SPACE (both default and the paper [103] settings) were ran on the corresponding datasets extracted from [37]. LexA has 9 sequences with sequence lengths from 80 to 580, and PurR has 12 sequences with sequence lengths from 100 to 600, respectively. Sequence logos were generated for the top ranked output motifs from the two algorithms, and were compared with the known motif logos. The results are shown in Figures 5.3 and 5.4.

In the LexA dataset, both GASMEN and SPACE found spaced motifs that are similar to the true LexA motif. Note that the problem is challenging because GASMEN had to search from a wide range of 4-25 and SPACE from 5-20 to find an optimal width for the motif. From Figure 5.3 we can see that GASMEN is successful to achieve the optimal width $W = 16$ with respect to conservation by removing the poorly conserved nucleotides at the two ends. GASMEN also retrieved a motif closer to the true LexA one than SPACE, where SPACE failed to find the correct

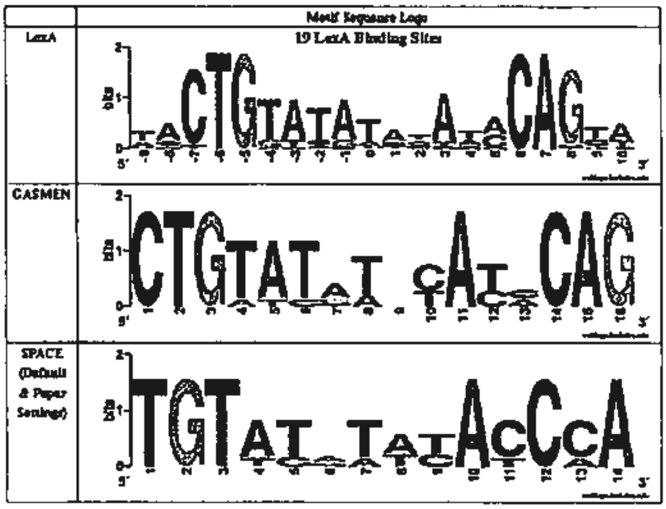


Figure 5.3: The comparisons of the motifs found on LexA dataset

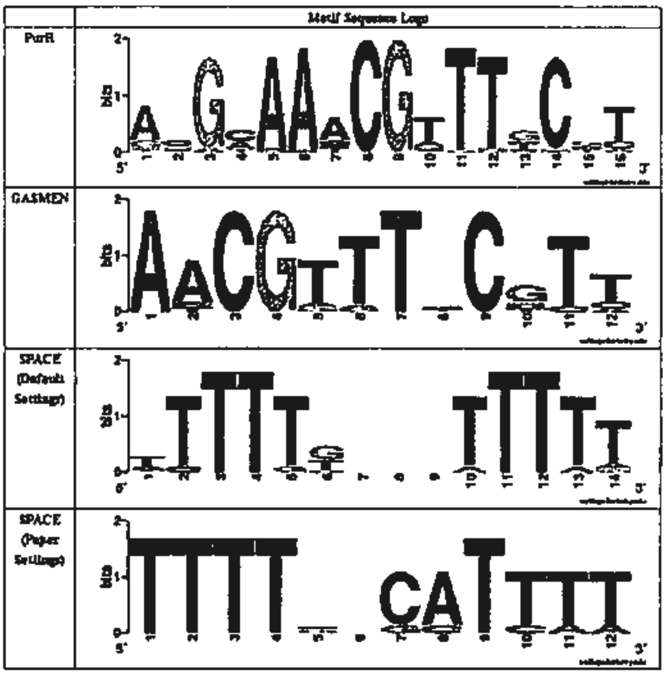


Figure 5.4: The comparisons of the motifs found on PurR dataset

submotifs of the second conserved segment.

In the PurR dataset, SPACE, with both default and paper settings, failed to find the correct motif logo. On the other hand, GASMEN found a motif which is close to the major part (6-16) of the true PurR motif. Because the first 4 nucleotides of PurR are overall too weakly conserved, GASMEN did not retrieved the degenerate part although G and A are well conserved. In this preliminary study with comparisons on two representative real spaced motifs, GASMEN outperforms SPACE with respect to finding the accurate motif logos and choosing the optimal widths from a wide possible range.

5.3.3 Quantitative Comparisons on 8 Real Datasets

Although GASMEN is designed for finding generic spaced motifs, it does not mean it is not capable of discovering general motifs. Moreover, in practice, no one can tell in advance whether a dataset has monad or spaced motifs. As a result, it is desirable to test the performance of GASMEN on general real datasets for motif discovery.

In this part, 8 real benchmark datasets [101] for testing monad motif discovery [19,100] were employed to test GASMEN, Weeder and SPACE. The 8 datasets cover different motif properties, with species ranging from prokaryotic (*E. coli*) to eukaryotic (*homo sapiens*), width from 6 to 22, sequence lengths from 105 to over 300, and total sequences numbers varying from 17 to 95. Among the 8 datasets, the CRP (cyclic AMP receptor protein) binding site motif in *E. coli* is a spaced motif with width 22, which contains two weakly conserved monad motifs separated by a gap [96]. The ERE dataset contains binding sites called estrogen response elements (EREs) with high affinity and activates gene expression in response to estradiol [47]. The E2F family [46] binding sites are from mammalian sequences. The five additional datasets for the TFs of CREB, MEF2, MYOD, SRF and TBP are from the ABS eukaryotic database [13]. The binding sites are labeled for the datasets such that quantitative comparisons can be performed. The datasets have been well studied where the chance to have some unknown TFBSs is small [100], and thus facilitate quantitative comparisons on performance.

We employ the following representative performance measures: the positive prediction value (precision) PPV, and sensitivity (recall) Sn, which are defined as follows respectively:

$$PPV = \frac{TP}{TP + FP} \tag{5.5}$$

$$Sn = \frac{TP}{TP + FN} \tag{5.6}$$

where TP is true positive, FN is false negative, and FP is false positive. F-score and the performance coefficient (PC) serve for similar purposes to reflect the balanced performance of PPV and Sn respectively as follows:

$$F = \frac{2 * PPV * Sn}{PPV + Sn} \tag{5.7}$$

$$PC = \frac{TP}{TP + FP + FN}. \tag{5.8}$$

If $TP = 0$ ($PPV = Sn = 0$), F is set to 0. All the measures are defined on both site (prefix s, and a predicted site has to overlap with at least 1/4 of the true one to be a TP) and nucleotide (prefix n) levels.

The performance of GASMEN, Weeder and SPACE on GAME is shown in Table 5.2. Note that except for CRP which has long width 22, the other datasets in general have short widths ranging from 6 to 13, and thus Weeder is favored for it supports and searches only widths 6-12. The test is tougher for GASMEN and SPACE because they search through a wide width range. As a result, SPACE only gives poor performance on those datasets with short motifs (we have chosen the best results from the top 10 outputs with both default and paper settings, if the top results are 0 in F and PC). On the other hand, with algorithm design catering for both monad and spaced motifs, GASMEN achieves competitive performance even compared with Weeder. In 6 out of the 8 datasets GASMEN has best performance in terms of both F-score and performance coefficient PC on both site (s) and nucleotide (n) levels. Weeder outperforms GASMEN in TBP dataset probably because TBP is a monad motif and has a very short width 6, which represents the best scenario Weeder is designed for. The experiments demonstrate the robust and competitive performance of GASMEN even for general monad motif discovery problems.

For the CRP dataset included, which is in fact a spaced motif, GASMEN outperforms SPACE in both PC and F on both site and nucleotide levels, indicating that GASMEN is still more promising for spaced motif discovery when compared quantitatively.

5.3.4 Quantitative Comparisons on the eukaryotic benchmark

We further compare GASMEN with GALF-G, MEME and Weeder on the improved eukaryotic benchmark [87]. There are 3 suites: 2 algorithm benchmarks and 1 model benchmark, all with real TFBS motifs extracted from TRANSFAC and includes representative eukaryotic species. The algorithm benchmark suite contains motifs that are supposed to be with certain conservation and the patterns can be learned by training based methods (the TFBS motifs have distinguishing power against the background). The model benchmark is the greatest challenge on existing motif models and methods because there is no explicit

Table 5.2: The comparisons of GASMEN, Weeder and SPACE on the 8 real datasets. n: nucleotide level; s: site level.

	GASMEN				Weeder				SPACE			
	Sn	*PPV*	*F*	*PC*	*Sn*	*PPV*	*F*	*PC*	*Sn*	*PPV*	*F*	*PC*
CREB												
n	0.41	0.66	0.51	**0.34**	0.40	0.41	0.41	0.26	0.00	0.00	0.00	0.00
s	0.68	0.65	0.67	**0.50**	0.79	0.42	0.55	0.38	0.00	0.00	0.00	0.00
CRP												
n	0.38	0.83	0.52	**0.35**	0.18	0.39	0.25	0.14	0.26	0.96	0.41	0.26
s	0.58	0.88	0.70	**0.54**	0.63	0.37	0.46	0.30	0.38	1.00	0.55	0.38
E2F												
n	0.42	0.28	0.33	**0.20**	0.48	0.22	0.31	0.18	0.06	0.09	0.07	0.04
s	0.78	0.28	0.41	**0.26**	0.89	0.22	0.36	0.22	0.11	0.19	0.14	0.08
ERE												
n	0.70	0.76	0.73	**0.57**	0.26	0.25	0.26	0.15	0.37	0.79	0.51	0.34
s	0.76	0.76	0.76	**0.61**	0.56	0.25	0.35	0.21	0.44	0.79	0.56	0.39
MEF2												
n	0.00	0.00	0.00	0.00	0.02	0.01	0.01	0.01	0.03	0.04	0.03	**0.02**
s	0.00	0.00	0.00	0.00	0.06	0.01	0.02	**0.01**	0.00	0.00	0.00	0.00
MYOD												
n	0.14	0.14	0.14	**0.08**	0.02	0.01	0.01	0.01	0.10	0.10	0.10	0.05
s	0.14	0.50	0.22	**0.13**	0.00	0.00	0.00	0.00	0.10	0.22	0.13	0.07
SRF												
n	0.41	0.63	0.50	**0.33**	0.26	0.47	0.34	0.20	0.14	0.51	0.22	0.12
s	0.51	0.69	0.59	**0.42**	0.63	0.54	0.58	0.41	0.17	0.60	0.27	0.15
TBP												
n	0.72	0.43	0.54	0.37	0.74	0.52	0.61	**0.44**	0.05	0.05	0.05	0.03
s	0.86	0.45	0.59	0.42	0.90	0.56	0.69	**0.52**	0.05	0.10	0.07	0.04

Table 5.3: Average performances (nPC and nCC) of GASMEN, GALF-G, MEME and Weeder on the eukaryotic benchmark.

Algorithms	Algo Markov		Algo Real		Model Real	
	nPC	nCC	nPC	nCC	nPC	nCC
GASMEN	0.091	0.116	0.112	0.167	0.045	0.090
GALF-G	0.102	0.138	0.095	0.126	0.045	0.070
MEME	0.077	0.097	0.063	0.083	0.020	0.029
Weeder	0.032	0.052	0.055	0.096	0.054	0.105

conservation nor motif in the set. There are 50 datasets with backgrounds generated by Markov models and 50 with real cis-regulatory region backgrounds (more realistic). The real benchmark contains 25 datasets with real cis-regulatory region backgrounds. The widths are not given in the benchmark. The additional evaluation measure corresponding to this benchmark is the nucleotide level correlation coefficient (nCC) [37,87,99].

The comparison results are shown in Table 5.3. With the maximal width W set to be 16, GASMEN has achieved the best performance in the algorithm benchmark with real backgrounds, and is better than GALF-G by 33% in nCC, although GASMEN is slightly outperformed in the aritifical Markov backgrounds (0.116 VS 0.138). The algorithm benchmark with real background shows the most practical scenarios in real data and GASMEN should be considered as the favorable choice (101% better than MEME and 74% better than Weeder in nCC). While GALF-G degenerates (0.070 in nCC) in the most challenging and difficult real benchmark, GASMEN still maintains competitive performance (0.090 VS 0.105 in nCC), as compared with the best Weeder which takes advantage of voting and searches on a smaller width range. Considering all 3 suites, GASMEN has the best balance of performance among them, and shows the best performance in the most practical algorithm benchmark.

5.4 Conclusions

In this chapter, we address the challenging problem of generic spaced motif discovery on nucleotides. To relax the previous constraints on spaced motifs, we have proposed Genetic Algorithm (GA) for Spaced Motifs Elicitation on Nucleotides (GASMEN), which searches from a wide range of possible widths (4-25) for both monad and spaced motifs. To the best of our knowledge, GASMEN is the first GA to address generic spaced motif discovery beyond monads and dyads, without stringent gap number and range constraints. GASMEN employs submotif indexing to partition the search space into smaller sub-space for GA, wherein it is easier to reach optimal motifs utilizing the schemata property of GA. Multiple-motif control has been proposed to avoid redundant computation, and is potentially useful to discover multiple motifs simultaneously. The probabilistic refinement memetic operator has also been developed to improve motif quality effectively and efficiently.

The experimental results, though still preliminary, on real representative spaced motifs of *E. coli* demonstrate the competitive and robust performance of GASMEN to find accurate motifs and optimal widths, compared with the state-of-the-art method SPACE. GASMEN is also capable of finding monad motifs, outperforming both Weeder and SPACE on most of the 8 real benchmark datasets, which contains both monad and spaced motif datasets from prokaryotic and eukaryotic species. GASMEN also shows the best balance of performance on the eukaryotic benchmark compared with GALF-G, MEME and Weeder.

Nevertheless there is still a lot of future work to do to improve generic spaced motif discovery. More real datasets are to be tested for comprehensive statistics to analyze the effectiveness and robustness of GASMEN for further improvement. The multiple-motif control has the potential to be extended to sup-

Table 5.4: Summary of GALF-P, GALF-G and GASMEN

	GALF-P	GALF-G	GASMEN
Motif Type	Monad	Monad (Generalized)	Spaced (Generic)
Motif Type Width (w)	Fixed (Known)	Range (Pior knowledge)	Any (No prior knowledge)
Scoring Function f	IC	Generalized Bayesian	Log likelihood ratio
Memetic Operator	Local Filtering	Local Filtering	Probabilistic Refinement
Motif No. (K)	Single	Multiple	Multiple
Instance Assumption	OOPS	OOPS/ZOOPS	ANOPS
Instance Adjustment	Post-processing	Post-processing	No need
Similarity Control	N	Y	Y

port multiple optimal spaced motifs effectively and efficiently. We will research into reducing the overheads of submotif indexing, because there are many similar and possibly redundant submotifs to be pruned. To exploit the sequence information for motif discovery, we are also interested in incorporating sequence bending properties such as curvature into conservation to capture more accurate motif properties. The GASMEN algorithm serves as a promising platform for the future work for improvement.

5.5 Summary

In this section, the three GA based motif discovery algorithms developed by us are summarized, namely GALF-P for optimization, GALF-G for modeling, GASMEN for spaced motifs, in Table 5.4. All of the proposed GA based algorithms have been extensively tested on comprehensive synthetic, real and benchmark datasets, and shown outstanding performances compared with state-of-the-art approaches. Our GA based algorithms also "evolve" to handle more and more relaxed cases, namely from fixed motif widths to most flexible widths, from single motifs to multiple motifs with overlapping control, from stringent motif instance assumption to very relaxed ones, and from contiguous motifs to generic spaced motifs with arbitrary spacers.

□ **End of chapter.**

Chapter 6

Discovering Approximate Associated Sequence Patterns for Protein-DNA Interactions

Summary

In this chapter, we further address the pattern discovery for TF-TFBS associated sequence patterns (rules), and make the first step to generalize the previous exact rules to approximate ones for both TFs and TFBSs.
Supplementary Data available at:
http://www.cse.cuhk.edu.hk/%7Etmchan/rules/

6.1 Introduction

In the previous chapters, the TFBS motif discovery problems we have addressed only consider one side of TF-TFBS binding, while discovering the binding patterns of both TF and TFBS can provide significantly better insight into protein-DNA interactions and further transcriptional regulation, as surveyed in the Background chapter.

In this chapter, we generalize the exact TF-TFBS associated sequence patterns to approximate ones on both sides. Many more informative rules are to be discovered compared with the exact ones, and they provide more detailed information to better understand protein-DNA binding mechanisms in the verification. The chapter layout is as follows: the proposed methods are detailed in the next section: **Materials AND Methods**; experimental results and verifications are reported in section **Results and Analysis**; and finally we have the **Discussion and Conclusion** section for the approximate approach.

6.2 Materials and Methods

In this section, we first present the data processed for investigations, and then elaborate the methodology of discovering approximate TF-TFBS associated sequence patterns.

6.2.1 Data Preparation

To perform the large-scale discovery on approximate TF-TFBS associated sequence patterns (or rules for short), we employ the updated version of TRANSFAC Professional 2009.4 (an older public version [72] is also available), which contains 13682 TF entries (7664 with protein sequences) and 1225 matrices of the TFBS nucleotide distributions (TFBS motif matrices). Each TF is associated with the set of TFBSs it binds to, and matrices are the aligned and refined profiles of the similar TFBSs bound by the same TFs, with the motif consensus represented with IUPAC codes, which can be considered as the approximate TFBS motifs.

Directly modeling (scoring) TF-TFBS associated sequence patterns as-a-whole is tempting, but it is computationally challenging. Alternatively, as the first study, we take advantage of the handy information of TFBS matrices (PWMs), in particular the TFBS motif consensuses, from TRANSFAC as part of the rules on the TFBS side. Note that the TFBS motif information is derived from TFBS sequence data using *de novo* motif discovery in TRANSFAC, so no extra information beyond typical TFBS motif discovery datasets is required if users want to discover the TFBS motifs themselves. The advantages of the available TFBS motifs include that: (i) the matrices are derived from datasets with better data integrity; (ii) TFBSs with varying widths from different experiments have been aligned based on Gibbs sampling [98], and a near-optimal width has been chosen for each TFBS motif; (iii) we can accelerate this first study for approximate rules based on the widely accepted representation and data.

For each TFBS matrix, we use the IUPAC consensus as the TFBS motif, and cut all leading and ending "N"s (poorly conserved and non-informative). Similar motif consensuses are grouped with 3 different hamming distance ratio threshold TY's: 0.0, 0.1 and 0.3, reflecting different levels of approximation criteria. In particular, for each motif consensus C of the 1225 matrices from TRANSFAC, we align it (and its reverse complement) with every other consensus C' for the best ungapped (substitution errors only) local pairwise alignment based on the hamming distance d. If d and the overlapping width w' between C and C' satisfy $d/w' \leq TY$, C' is grouped into C under threshold TY. Repeated consensuses are not processed again. For each TFBS consensus group, denoted by C, all the associated non-duplicate TF sequences are retrieved and then subject to CDHIT (with global sequence identify threshold 0.7) [54] to remove redundancy. Only non-redundant TF datasets with ≥ 5 sequences are kept. A summary of the TF datasets is shown in Table 6.1.

Table 6.1: The number of TF protein sequence datasets after preprocessing. Raw Group stands for the TF dataset number after TFBS consensus grouping; Redundancy Rm stands for the TF dataset number after CDHIT redundancy removal and with ≥ 5 protein sequences.

	TFBS TY		
TF Datasets	0.0	0.1	0.3
Raw Group	475	490	815
Redundancy Rm	75	99	506

6.2.2 Approximate TF Motif Discovery

Unlike the TFBS matrices and consensuses, there is no readily usable common motifs for the TF datasets retrieved by the preparation procedure. The core parts of TFBSs that closely interact with TFs are generally considered very short, so it is desirable to discover the short and conserved interacting amino acids from TFs. MEME, as one of the most widely used tools, did discover TF domain motifs which can be matched in verified conserved domains. However, the motifs were long (without specifying the widths) and degenerate with great variations of many possible matches, which are neither precise nor concise to be verified (shown in the experiments). Thus we have to design a customized algorithm for the task, and useful features such as the hydrophilic properties favoring binding can also be incorporated.

To best fit our objective, a simple customized algorithm is developed to discover short approximate TF motifs. The inputs are the TF data with n sequences $S = \{S_i\}$, $i = 1, ..., n$ corresponding to a TFBS group C, the specified motif width W and the maximal error E. The outputs are the top K (=10 in our experiments) TF motifs T_k ($k = 1, ..., K$) and their corresponding matches $\{t_{i,j}\}_k$ maximizing certain motif scoring function f. i is the sequence index of S_i, and $j = 0, 1$ is the match index, indicating at most one match per sequence ($j = 0$ means there is no match in S_i). Since the binding cores should be highly conserved, E is small in the expected target motifs. As a result, all W-substrings (W-mers) extracted by a sliding window on S are considered feasible to cover most of the probable motifs, without enumerating all 20^W possible W-mers. For each candidate motif T as a W-mer retrieved by the sliding window, all W-mers within hamming distance (substitution errors) E from T are retrieved as the candidate match set $\{tc_{i,j}\}$. i is the sequence index, and $j = 1, ..., q_i$, is the match index where q_i is the total number of matches in S_i. Exceptionally, $q_i = 0$ means no candidate match for S_i. The Blosum matrices are not used because they tend to favor complicated degenerate patterns (as existing tools do) while we aim at finding the the short and highly conserved motifs. To favor the residues that are likely to be on the surface for binding, a candidate motif T should have at least one hydrophilic amino acid with a scale < 0 (namely R, K, D, Q, N, E, H, S and T) from the normalized hydrophobic index [23].

There can be several approximate matches to the same motif T from $\{tc_{i,j}\}$, but only the best match (one actual TF interacting core for one given TFBS core) should be chosen for each sequence. This is important but seldom considered by current pattern based algorithms. Given the candidate set $\{tc_{i,j}\}$, we employ the Bayesian scoring function [40] used for TFBS motif discovery to choose the most probable set of matches $\{t_{i,j}\}$, $j = 0, 1$, from $\{tc_{i,j}\}$. A customized iterative refinement approach is proposed. Firstly all the first candidate matches, if any, are selected as the initial instance set $\{t'_{i,j}\} \leftarrow \{tc_{i,1}\}$ to build the initial position weight matrix (PWM) Θ of the amino acid distributions, where $\Theta_{a,b}$ represents the frequency of amino acid $b \in \Sigma$ at column $a \in [1, W]$. The background frequency of amino acid b, $\Theta_{0,b}$, can be calculated from input S. Then the

Bayesian scoring function [40] to be maximized is as follows:

$$f = |\{t'_{i,j}\}|(\sum_{a=1}^{w}\sum_{b\in\Sigma}\Theta_{a,b}\log\frac{\Theta_{a,b}}{\Theta_{0,b}} + \log\frac{p}{1-p} - 1) \qquad (6.1)$$

where $p = |\{t'_{i,j}\}|/|S|$ is the abundance ratio defined as the number of the matches, $|\{t'_{i,j}\}|$, over the dataset size $|S|$. The score reflects log posterior probability of having Θ and $\{t'_{i,j}\}$ with a non-informative prior. f can capture the over-representation and conservation concept of motifs with probability better than the simple supports (i.e. counts) [52], which could be large by chance only.

The algorithm iteratively (maximal 20 iterations) tries the other candidates $tc_{i,j'}$ one by one at each S_i, and accepts the change if the new Θ improves f. If there is no change after trying all the matches from $\{tc_{i,j}\}$. The algorithm stops and outputs the top K best T associated with $\{t_{i,j}\}$. The algorithm converges very fast in experiments because there are only a few near-optimal matches to be chosen from each S_i with a small E set. To speed up, for each TF dataset, only the motifs with matches for $\geq n/2$ sequences are eligible to be processed to reduce computational time. Repeating motifs will not be doubly-processed.

6.2.3 Approximate TF-TFBS Associated Sequence Patterns

Pairing the TFBS (approximate) consensus C ready in TRANSFAC and each of the best TF approximate motifs T discovered by the customized algorithm, we have the approximate TF-TFBS associated sequences patterns as T-C for further evaluation. The whole procedure is shown in Figure 6.1.

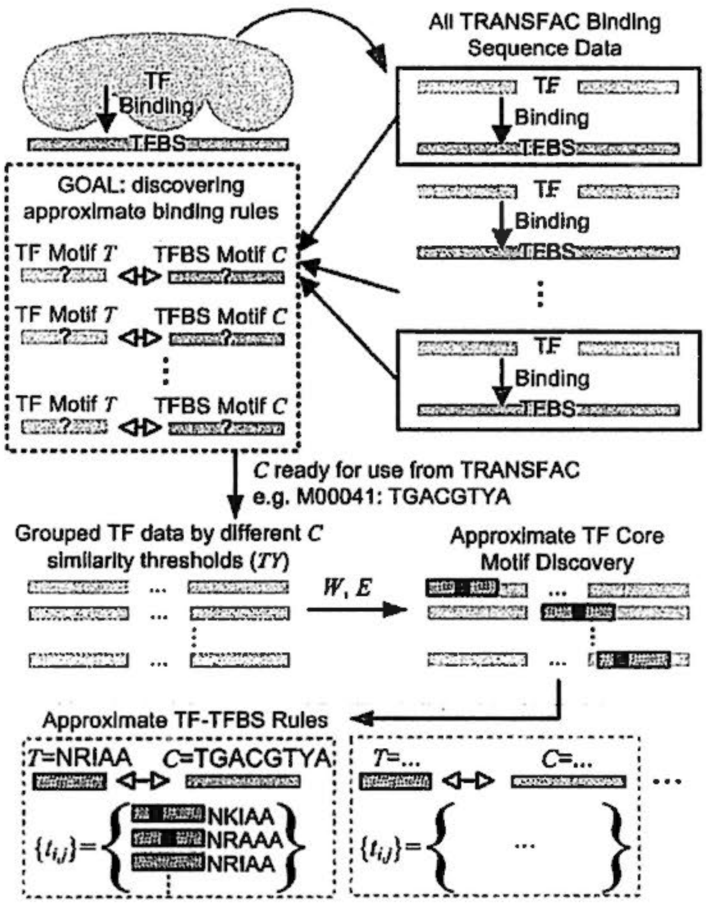


Figure 6.1: The whole procedure of discovering approximate TF-TFBS associated sequence patterns.

6.3 Results and Analysis

In this section, the discovered rules from experiments are reported, followed by detailed analysis and independent verification.

6.3.1 Experimental Settings

With the 3 TY threshold settings of TFBS consensus grouping, different settings of $W = 5, 6$ and $E = 0, 1$ were used to run the TF motif discovery to generate different approximate TF-TFBS associated sequences patterns (referred simply as rules later on) from the extracted TRANSFAC data.

To evaluate the discovered rules based only on TF-TFBS sequences, the 3D protein-DNA complex structures from Protein Data Bank (PDB) were employed as the verification evidences. In particular, we downloaded 2457 PDB entries labeled with prot-nuc (protein-nucleotides) with redundancy removal at 90% sequence identity (same as the previoius study [52]). We then removed entries without DNA chains (509 RNA entries), resulting in 1948 entries.

For each downloaded PDB entry, the distances between each amino acid on each protein chain and each nucleotide on each DNA chain were computed. If the respective residues (amino acid and nucleotide) have atoms that are close enough to be considered binding ($\leq$ 3.5 angstrom following [1, 2, 52]), the sequence pair P-D composed of the protein W'-mer P and DNA W'-mer D surrounding the particular close residues in the center was output, where W' is chosen as $2 * W - 1$. Thus if a W'-mer contains a W-mer from the discovered rules, the W-mer is guaranteed to contain the close (binding) residue pair. Thus $W' = 9, 11$ for $W = 5, 6$ settings respectively. These TF-TFBS W'-mer binding pairs (P-D pairs) were collected and compiled for the verifications (see Figure 6.2). The summary is shown in

Table 6.2.

Table 6.2: The summary of PDB binding data (P-D pairs) with different binding W' settings.

	Binding pair W'	
	9 (for W=5)	11 (for W=6)
PDB Entries	1290	1177
Protein Chains	2558	2348
DNA Chains	2989	2630
P-D pairs	40222	31530

For each rule T-C specified by W (width only for TF, because C is retrieved from TRANSFAC) and error E with the TF instance set (optimal matches) $\{t_{i,j}\}$, there are two levels of verification for the PDB binding data, **TF**: verified on the TF side by protein (P) evidences, and **TF-TFBS**: verified on both sides by protein-DNA (P-D) evidences. To be consistent with the previous study for comparisons, only rules with ≥ 7 instances are evaluated.

TF side: Since both the motif T and the instance set $\{t_{i,j}\}$ are obtained, one can directly compare each instance $t_{i,j}$ with protein substring P from the PDB binding data for their presence. The instance set $\{t_{i,j}\}$ for verification has the advantage of being more stringent and concise, as compared to using pattern (T, E) which may generate non-existing approximate instances. The verification approach is supported by the statistical significance shown later. A TF instance $t_{i,j}$ is verified on P if the W-mer $t_{i,j}$ is present in certain TF $W' = 2 * W - 1$-mer(s) of P from the PDB P-D pairs, e.g. $t_{i,j} = NRAAA$ present in $P = FLERNRAAA$. The **TF verification ratio** R_{TF} for a rule with TF motif T is defined as the number of verified TF instances over the total number of instances $|\{t_{i,j}\}|$. Thus if $E = 0$, R_{TF} is either 0 or 1 because all instances are the same as the TF motif T.

TF-TFBS sides: A TFBS motif consensus C from TRANS-

FAC is verified if there exist an W-mer in C, or its reverse complement, with at most E error from a present W-mer of D in the PDB P-D pairs. Note that since IUPAC code is employed in C, an ambiguity nucleotide can match any of its inclusive nucleotides (e.g. S matches C/G). For example with $W = 5$ and $E = 1$, $C = TGACGTYA$ is verified with $D = TCGATGACG$ because $TGACG$ (reverse complement $CGTCA$) matches the last W-mer of D. The TFBS verification is slightly more flexible than TF one, according to the higher variability TFBSs exhibit in TF-TFBS binding [64].

Thus an approximate (W, E) TF-TFBS rule instance $t_{i,j} - C$ is verified if both the TF instance $t_{i,j}$ and the TFBS motif C can be verified on P-D PDB pairs. The **TF-TFBS verification ratio** $R_{TF-TFBS}$ for a rule T-C is defined as the number of verified $t_{i,j} - C$ over the total number of rule instances (determined on the T side, i.e. $|\{t_{i,j}\}|$). Thus $R_{TF-TFBS} \leq R_{TF}$. If $R_{TF} = 0$ (not verified on TF side), $R_{TF-TFBS} = 0$ (impossible to be further verified). The verification procedure is illustrated in Figure 6.2.

6.3.2 Rule Results

Table 6.3 shows the verification ratios, R_{TF} on the TF side and $R_{TF-TFBS}$ on both sides, on the corresponding PDB binding data, with respect to all TFBS consensus grouping TY, width W and error E settings. All detailed results of the rules are available in the Supplementary Data.

To compare with the previous study with exact TF-TFBS rules [52], the results for $W = 5, 6$ (all rules with TF width W and TFBS width $\geq W$ are merged as one W setting for consistency) are collected and evaluated with the same verification procedures described above. The most exact setting from the approximate rules is $E = 0$ for $TY = 0.0$. Note that approx-

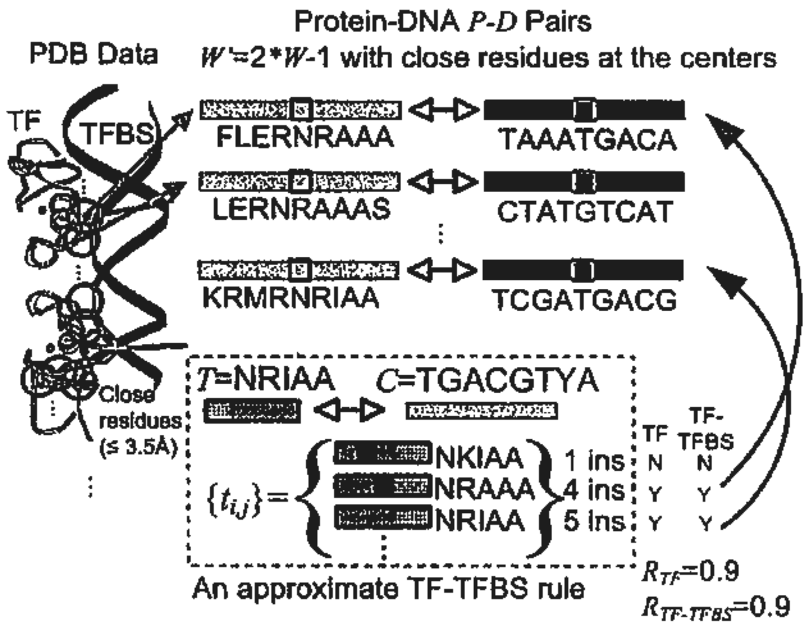


Figure 6.2: An illustrative example of generating P-D pairs from PDB and verifying the approximate TF-TFBS rules for $W = 5$, $E = 1$ ($W' = 9$).

imate information is already implicitly included even for this setting because of the IUPAC TFBS motifs from TRANSFAC.

The approximate rules have uniformly better average verified ratios (AVG R_*), e.g., better R_{TF} by 29% (0.74 VS 0.57, $W = 5$) and 300% (0.71 VS 0.18, $W = 6$) respectively, even when exact TF motifs are expected ($E = 0$). Similar improvements on AVG $R_{TF-TFBS}$ are observed, with 46% (0.64 VS 0.44, $W = 5$) and 226% (0.58 VS 0.18, $W = 6$) respectively. The improved performance indicates the advantage of grouping approximate TFBS consensuses and discovering hydrophilic and probable TF motifs, over the exact counts (supports) [52]. Furthermore, with the approximate extensions, many more informative rules (rules with $R_* > 0$) than exact ones are found ($W = 5$: 110 VS 76 and $W = 6$: 88 VS 6, on $R_{TF-TFBS} > 0$), while maintaining competitive informative rule ratios ($R_* > 0$ ratio). The previous exact rules [52] become less appealing when W increases because there are fewer exact rules reaching the support threshold. Note that AVG R_* is equal to $R_* > 0$ Ratio when $E = 0$ because all instances $t_{i,j}$ are the same and they are either "all verified" ($R_* = 1$) or "none verified" ($R_* = 0$) for a rule T-C.

The approximate rules also superset the exact ones in general. By summarizing all $E = 0$ rules across different TY settings, the approximate rules for $W = 5, E = 0$ cover 79% of the $W = 5$ exact rules on TF sides, and 79% on both sides. $W = 5, E = 1$ rules further cover 85% TF and 82% TF-TFBS exact rules. The small portions of the non-overlapping rules are probably due to the different data collection methods used (exact: TF oriented and all TFBSs used [52]; ours: TFBS consensu groups oriented and some original TFBSs ignored). Approximate rules for $W = 6, E = 0$ also cover 88% TF and 85% TF-TFBS exact rules respectively. Examples verified by the exact rules [52] are also covered by the approximate rules. The exact rule GGTCA-CEGCK, representing the P-box within Bp-nhr-2 binding domain [73], is contained in 19 approximate rules (by matching the motifs) from all settings. 17 of the approximate rules are with both $R_{TF} = 1$ and $R_{TF-TFBS} = 1$, and 2 with $R_{TF} = 0.96$ and $R_{TF-TFBS} = 0.96$. The corresponding approximate rules have other verified TF instances ($t_{i,j}$) such as CEACK (PDB: 1LO1) and CESCK (PDB: 2A66, 2FF0), demonstrating the better generality to discover real TF-TFBS binding patterns. The exact rule AAACA-IRHNL is also contained by 12 approximate rules, with other verified $t_{i,j}$ such as VRHNL (PDB: 2A07, 2AS5).

6.3.3 Comparisons with MEME

As a representative tool we used, MEME was also run on the same TF datasets with the W, E and TY settings. MEME uses expectation maximization to discover TF/TFBS motifs in the PWM representation, by minimizing the chance of having random motifs with better information content (IC) [5]. Hence

Table 6.3: The verified rules on PDB binding data (P-D pairs) with different TY, W and E settings, compared with the corresponding $W = 5, 6$ exact rules in the previous study [52]

	$W = 5, E = 0$		$W = 5, E = 0$						$W = 5, E = 1$					
TY	Exact rules [52]		0.0		0.1		0.3		0.0		0.1		0.3	
R_*	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS
AVG R_*	0.57	0.44	0.74	0.64	0.78	0.70	0.82	0.73	0.57	0.56	0.63	0.62	0.69	0.68
$R_* > 0$	99	76	127	110	165	147	636	567	235	231	291	287	2101	2072
Rule No.	173	173	172	172	211	211	774	774	340	340	396	396	2559	2559
$R_* > 0$ Ratio	0.57	0.44	0.74	0.64	0.78	0.70	0.82	0.73	0.68	0.67	0.73	0.72	0.82	0.81

	$W = 6, E = 0$		$W = 6, E = 0$						$W = 6, E = 1$					
TY	Exact rules [52]		0.0		0.1		0.3		0.0		0.1		0.3	
	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS
AVG R_*	0.18	0.18	0.71	0.58	0.76	0.65	0.81	0.67	0.58	0.54	0.63	0.60	0.70	0.68
$R_* > 0$	6	6	108	88	143	121	448	370	181	169	234	222	1665	1618
Rule No.	34	34	153	153	187	187	555	555	271	271	319	319	1920	1920
$R_* > 0$ Ratio	0.18	0.18	0.71	0.58	0.76	0.65	0.81	0.67	0.67	0.62	0.73	0.70	0.87	0.84

MEME is likely to produce degenerate motifs (error E can be large) with respect to the consensus representation. MEME was set with fixed widths ($W = 5, 6$) and ZOOPS (zero or one (TF) instance per sequence) for consistency. AVG R_{TF}, AVG $R_{TF-TFBS}$ and $R_* > 0$ Ratio were measured and compared with our approach. There is no error E parameter for MEME, so the same set of results for a specific W were measured twice with $E = 0$ and $E = 1$, of which the same R_{TF} results are expected because the TF performance measurement is instance oriented (matching $\{t_{i,j}\}$). On the other hand, $R_{TF-TFBS}$ will increase from $E = 0$ to more relaxed $E = 1$. The comparison results are shown in Table 6.4. Our approach is 73% − 262% better in terms of AVG R_* than MEME for all different settings. MEME did find more rules in general because it tends to discover degenerate motifs. However, the verification ratios ($R_* > 0$ Ratios) on all settings of our approach are 33% − 79% better than MEME. The significant improvements indicate that our aim for highly conserved and short TF core motifs with hydrophilic constraints better achieves the goal of this specific problem than MEME which targets for general and degenerate motifs.

Table 6.4: MEME results on different TY, W and E settings and the improved ratios of our approach over MEME (**Ours better by** referring to Table 6.3).

MEME Results	$W=5, E=0$						$W=5, E=1$					
TY	0.0		0.1		0.3		0.0		0.1		0.3	
R_*	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS
AVG R_*	0.33	0.20	0.36	0.28	0.37	0.28	0.33	0.32	0.36	0.34	0.37	0.36
Ours better by	124%	144%	120%	146%	120%	100%	73%	74%	76%	70%	85%	91%
$R_* > 0$	143	123	179	151	1306	1071	143	142	179	175	1306	1262
Rule No.	298	298	342	342	2118	2118	298	298	342	342	2118	2118
$R_* > 0$ Ratio	0.48	0.41	0.52	0.44	0.62	0.51	0.48	0.48	0.52	0.51	0.62	0.60
Ours better by	54%	56%	49%	68%	33%	45%	42%	40%	40%	42%	33%	36%

MEME Results	$W=6, E=0$						$W=6, E=1$					
TY	0.0		0.1		0.3		0.0		0.1		0.3	
R_*	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS
AVG R_*	0.29	0.22	0.31	0.23	0.29	0.18	0.29	0.27	0.31	0.29	0.29	0.26
Ours better by	142%	163%	145%	181%	178%	262%	97%	96%	102%	104%	142%	157%
$R_* > 0$	127	96	163	121	1194	839	127	120	163	154	1194	1127
Rule No.	289	289	334	334	2170	2170	289	289	334	334	2170	2170
$R_* > 0$ Ratio	0.44	0.33	0.49	0.36	0.55	0.39	0.44	0.42	0.49	0.46	0.55	0.52
Ours better by	61%	73%	57%	79%	47%	72%	52%	50%	50%	51%	58%	62%

6.3.4 Statistical Significance

To test the statistical significances ($W = 5$ results for illustration) on R_{TF} and $R_{TF-TFBS}$, an empirical method is employed to simulate if the rules are randomly generated from the datasets. For each TY and E setting, each dataset corresponding to a TFBS consensus C is sampled equal times to output 10 TF motifs (denoted by T'), with m instances $t'_{i,j}$ generated with at most E from T', where m is randomly sampled to be valid for the above evaluation (i.e. ≥ 7 and $\geq n/2$, i.e. at least half of the sequence number). The sampling time for each C dataset is set such that there are $N \geq 10000$ datasets (e.g. $N = 134 * 75 = 10050$ for the 75 datasets with $TY = 0.0$ and $E = 0$) with totally $10{*}N$ rules generated. The empirical p-value of a rule is thus the proportion of random rules that has equal or better performance of R_* than it. The results for statistically significant rules (with p-values < 0.05) for $W = 5$ are summarized in Table 6.5. Note that for $E = 0$, each random rule is either $R_* = 0$ or $R_* = 1$, and the best achievable p-values on TF side (i.e. $p(R_{TF} \geq 1)$) are 0.0625 (TY=0.0), 0.0668 (TY=0.1) and 0.0602 (TY=0.3). In such cases the number of rules with the best achievable p-values are shown. It can be seen that the majority of the rules (0.64 – 0.79) are statistically significant for the TF-TFBS verification ratios $R_{TF-TFBS}$, indicating the competitive performances achieved by the approximate rules are not trivial.

Table 6.5: The statistically significant rules for W = 5. * indicates the number of rules with the best achievable P-values when they are > 0.05 (all < 0.07).

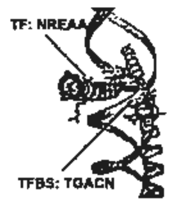
	$W=5, E=0$						$W=5, E=1$					
TY	0.0		0.1		0.3		0.0		0.1		0.3	
R_*	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS	TF	TF-TFBS
P-value < 0.05	0 (127*)	110	0 (165*)	147	0 (636*)	567	223	226	278	272	1974	2023
Rule No.	172	172	211	211	774	774	340	340	396	396	2559	2559
Significant Ratio	0 (0.74*)	0.64	0 (0.78*)	0.70	0 (0.82*)	0.73	0.64	0.65	0.70	0.69	0.77	0.79

6.3.5 Detailed Analysis

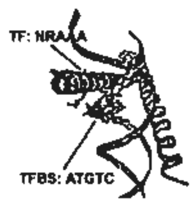
In this subsection, we investigate how the approximate rules generalize the exact ones with the verified PDB entries for illustration.

With the setting W = 5 and E = 1, we show how approximate rules generalize and retrieve informative verified evidences on both TF and TFBS sides. From the 231 verified ($R_{TF-TFBS} > 0$) TF-TFBS rules for TY = 0.0, there are 133 verified rules with $\geq$ 5 PDB entries (maximum number of verified entries: 23). An illustrative rule with 5 verified PDB entries is chosen for illustration. The rule is M00041: NRIAA-TGACGTYA (ID 1160), with maximal $E = 1$, the different TF instances (i.e. $\{t_{i,j}\}$) discovered by the customized algorithm are NKIAA, NRAAA, NREAA, and NRIAA. Except NKIAA, other instances have been verified with PDB entries, namely 1DH3, 1FOS, 1JNM, 1T2K, and 2H7H. The case with NKIAA, is shown to be within TF records of NCBI in next subsection. The results are shown using ProteinWorkshop in Figure 6.3. By allowing maximal 1 substitution error, we discover that the TF binding motif NR*AA summarized from our results is flexible with the middle amino acid, varying with E, A, and I. Such discoveries supported by the approximate rules give us more clues into the TF-TFBS binding mechanisms.

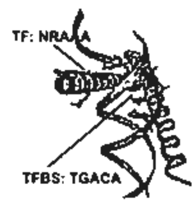
(a) 1DH3: NREAA-NGTCA



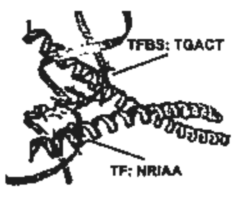
(b) 1DH3: NREAA-TGACN



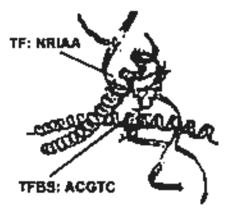
(c) 1T2K: NRAAA-ATGTC



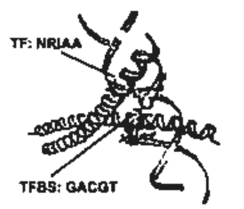
(d) 1T2K: NRAAA-TGACA



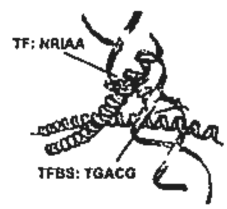
(e) 1FOS: NRIAA-TGACT



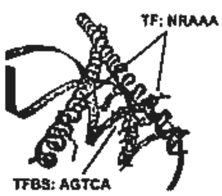
(f) 1JNM: NRIAA-ACGTC



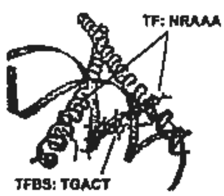
(g) 1JNM: NRIAA-GACGT



(h) 1JNM: NRIAA-TGACG



(i) 2H7H: NRIAA-AGTCA



(j) 2H7H: NRIAA-TGACT

Figure 6.3: PDB verifications for rule M00041: NRIAA(NKIAA; NRAAA; NREAA; NRIAA)-TGACGTYA for $W = 5$, $E = 1$, $TY = 0.0$ using ProteinWorkshop.

In order to investigate the case of NKIAA, a model was built based on the structure of 1JNM using homology modeling. As shown in Figure 6.4, the change of arginine (R) to lysine (K) does not introduce the steric effect and the basic property of the amino acid is retained (both are positive charge). NKIAA is also shown to be within TF records of NCBI [89] in the next subsection. Thus we believe that NKIAA should be a correct

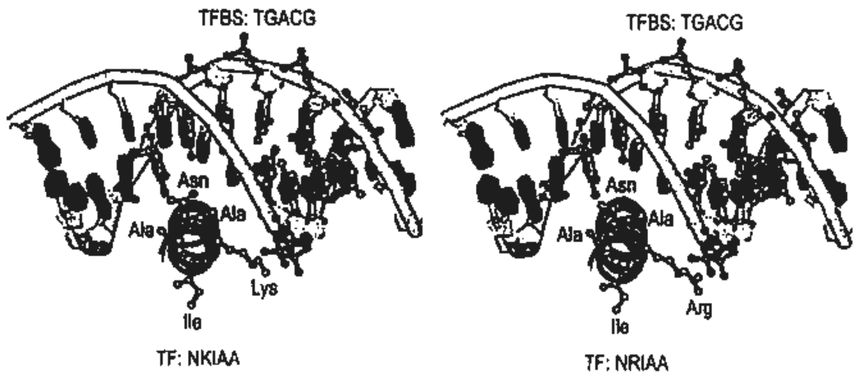


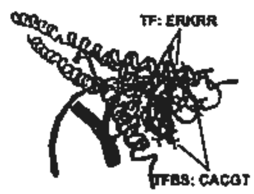
Figure 6.4: Homology modeling of NKIAA-TGACG which does not have PDB records, based on the verified NRIAA-TGACG pair. The model (left) was built based on and compared with the structure of 1JNM (right). The proteins are shown in ribbon diagram with the highlighted TF amino acids in ball and stick format. The TFBS sequences in the DNA are also highlighted in ball and stick format. The figures are generated using Discovery Studio Visualizer, Accelrys.

prediction.

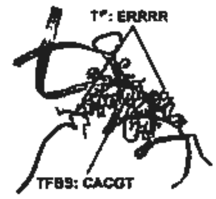
We further analyze the rule picked up from setting $W = 5$, $E = 1$ and $TY = 0.1$. The rule M00217: ERKRR-CACGTG has 3 different TF instances (i.e. $\{t_{i,j}\}$) ERKRR, ERQRR and ERRRR, and 5 verified PDB entries: 1AN2, 1AN4, 1HLO, 1NKP and 1NLW. The results are shown using ProteinWorkshop in Figure 6.5. This case further demonstrates the flexibility in specific positions for TF-TFBS binding. ER*RR has the variations of K, R and Q for the middle amino acid, and these variants can appear in the same TF-TFBS binding, for example, 1NKP (ERKRR and ERQRR) in Figure 6.5. The discovery prompts further investigation into the flexibility and specificity of protein-DNA interactions.

6.3.6 Conservation Verification on NCBI Protein Records

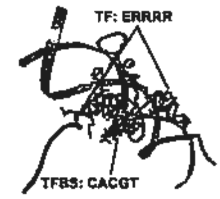
Besides the PDB entries, we further verified the approximate rules on NCBI [89] for conservation independently, namely check-



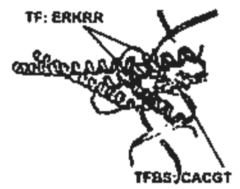
(a) 1AN2: ERKRR-CACGT



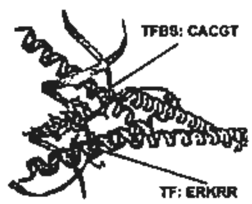
(b) 1AN4: ERRRR-CACGT



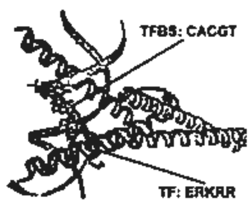
(c) 1AN4: ERRRR-CACGT (2nd match)



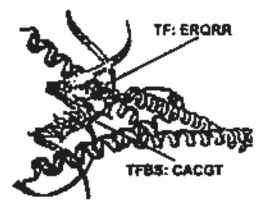
(d) 1NLW: ERKRR-CACGT



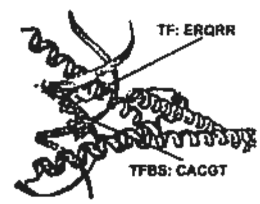
(e) 1NKP: ERKRR-CACGT



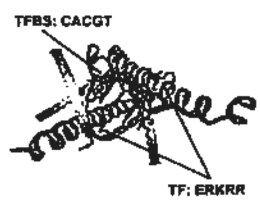
(f) 1NKP: ERKRR-CACGT (2nd match)



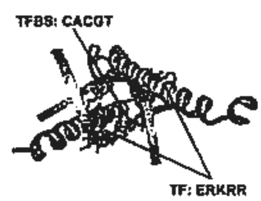
(g) 1NKP: ERQRR-CACGT



(h) 1NKP: ERQRR-CACGT (2nd match)



(i) 1HLO: ERKRR-CACGT



(j) 1HLO: ERKRR-CACGT (2nd match)

Figure 6.5: PDB verifications for rule M00217: ERKRR(ERKRR; ERQRR; ERRRR)-CACGTG for $W = 5$, $E = 1$, $TY = 0.1$ using ProteinWorkshop.

ing the occurrences of TF motif instances ($\{t_{i,j}\}$) with the related NCBI TFs (proteins) independently. The previous 134 rules with $R_{TF-TFBS} \geq 0.9$ ($W = 5, E = 1, TY = 0.1$) were compiled (grouped) according to their 39 different TFBS consensus C groups, and the first 10 groups were analyzed for illustration (because of the time-consuming manual inspection). For each C, the TF names FA and organisms OS of the related TFs were retrieved, and TF instances ($\{t_{i,j}\}$) found in the approximate rules were recorded. We then queried proteins in NCBI with FA, and check whether any instance in $\{t_{i,j}\}$ occurs in protein records of organisms **NOT included in** OS.

All the 10 groups are conserved within protein records in NCBI from organisms not recorded in the TRANSFAC data (see supplementary data for details). All of the TF instances are within the conserved domains (especially binding domains), except one case where the domain information is missing in NCBI, and overlap with the annotated DNA binding sites. For example, NREAA, NRAAA in the 1st, 7th and 10th groups are conserved among proteins (TFs) CREB1, ATF-1 in various organisms such as Danio rerio, Oncorhynchus mykiss and Saccharomyces cerevisiae, which are beyond the TRANSFAC data containing mainly higher mammals. NREAA and NRAAA, together with NKIAA and NRIAA in the 2nd group, are also conserved in proteins c-jun, ATF-3 from NCBI. VNEAF in the 3rd group is conserved among MyoD proteins of Sus scrofa and Meleagris gallopavo. None of these organisms are included in the corresponding TRANSFAC data used to discover the rules. There are also NCBI records with partial matches to the TF motifs we discovered, such as VNDAF (to VNEAF) and NRESA (to NREAA), implying that relaxing the approximation appropriately would further improve the results. Furthermore, the conserved TF instances are all within consistent conserved domains and overlapping with binding sites according to the NCBI annotations. For example, the conserved ERQRR and ERRRR from the 6th group are all within helix-loop-helix (HLH) domains in NCBI although they appear in various proteins such as USF, N-Myc and arnt. The conserved IRHNL in the 8th group is all within the forkhead (FH) domains.

NCBI serves as an independent annotation source for verification with proteins from organisms not included in the TRANSFAC data used for rule discovery. The confirmation of conservation of the discovered TF instances in NCBI records strongly indicates the approximate TF motifs are very likely to be real conserved binding cores across different organisms (especially when they are within consistent conserved domains and overlapping with DNA binding sites), thus demonstrating the accuracy and generality of the approximate rules for revealing real TF-TFBS interactions.

6.4 Discussion and Conslusion

Large-scale sequence patterns show great potentials for discovering TF-TFBS binding patterns for further understanding protein-DNA interactions. In this chapter, we have for the first time generalized the exact TF-TFBS associated sequence patterns [52] to approximate ones to discover more informative and intricate rules. We have taken advantage of the available TFBS motif consensuses C from TRANSFAC. Reliable datasets are ready for use through grouping the non-redundant TF sequences corresponding to similar TFBS consensuses C, which has greatly accelerated the study. A simple customized algorithm has been developed to help discover the short ($W = 5, 6$) and well conserved ($E = 0, 1$) TF motifs in an approximate manner. The algorithm better suits our need to have precise and concise rules, and significantly outperforms MEME by over 73%. Comprehensive measures, e.g. both TF and TF-TFBS verification ratios (R_*), verified rule ratios ($R_* > 0$ ratios), as well as statistical significances have been used to evaluate the discovered approximate TF-TFBS rules.

The discovered approximate TF-TFBS rules have demonstrated competitive performance with respect to verifications ratios (R_*) on both TF and TF-TFBS aspects. The approximate rules exhibit a strong edge over the previous exact ones on both average verification ratios (0.64 VS 0.44 for $W = 5$ and 0.58 VS 0.18 for $W = 6$ on AVG $R_{TF-TFBS}$) and number of informative rules (88 VS 6 for $W = 6$ on $R_{TF-TFBS} > 0$ rule number). The majority of the discovered rules are shown to be statistically significant (over 64% and up to 79% on $R_{TF-TFBS}$). With detailed analysis, the approximate rules are confirmed by the PDB binding structures visually and interatomic distances. The examples for various settings demonstrate the flexibility of specific positions TF-TFBS binding for both proteins and DNAs, reinforcing the need to extend exact rules to approximate ones to better discover TF-TFBS binding patterns. The approximate TF instances are conserved in binding domains and even binding sites according to the independent verification on NCBI records from organisms not included in TRANSFAC data used, and hense strongly support the biological significance of the discovered rules.

Compared with the previous study on exact rules, the proposed discovery of approximate TF-TFBS rules has demonstrated significantly better generalized capability of exploring more informative binding rules, and potential applications to predict protein-DNA interactions given either side for better decipher transcriptional regulation. Nevertheless, this study is just the first generalization step towards approximate TF-TFBS rule discovery. The revealed potentials drive us for more advanced models and algorithms. Future work includes introducing formal models to score the TF-TFBS rules as-a-whole, applying and/or developing novel search algorithms based on the new scoring functions, associating multiple short TF motifs, as well as handling uncertainty such as unknown widths. As the advanced computational facilities and techniques are being developed quickly, there will be numerous promising ways to further improve approximate TF-TFBS rule discovery greatly.

□ **End of chapter.**

Chapter 7

Conclusion

Summary

In this chapter, we conclude the thesis and provide further discussion on future work.

7.1 Conclusion

In this thesis, we have contributed to various aspects of pattern discovery for deciphering gene regulation, including extensive efforts on developing novel Genetic Algorithm (GA) based algorithms to discover Transcription Factor Binding Site patterns, i.e. TFBS motif discovery, with respect to optimization, modeling and generic spaced motifs. Moreover, we have developed approximate TF-TFBS associated patterns (TF-TFBS rules) discovery, which is very promising for better understand protein-DNA interactions for future applications.

On TFBS motif discovery, which is a very challenging problem with respect to both optimization and modeling, we have developed two novel Genetic Algorithm with Local Filtering (GALF) algorithms: GALF-P (post-processing) and GALF-G (generalized), as well as the extra Genetic Algorithm (GA) for

Spaced Motifs Elicitation on Nucleotides (GASMEN), to handle recent raised problems for generic and complicated spaced motif discovery. All of the proposed GA based algorithms have been extensively tested on comprehensive synthetic, real and benchmark datasets, and shown outstanding performances compared with the state-of-the-art approaches. Our algorithms also "evolve" to handle more and more generalized cases, namely from fixed motif widths to most flexible widths, from single motifs to multiple motifs with overlapping control, from stringent motif instance assumption to very relaxed ones, and from contiguous motifs to generic spaced motifs with arbitrary spacers.

We have further investigate the TF-TFBS binding pattern discovery in a generalized manner with approximation. The approximate TF-TFBS associated sequence patterns (rules) are essential to better understand and interpret protein-DNA interactions, which are fundamental for gene regulation. Based on a progressive approach taking advantage of existing TFBS motifs from TRANSFAC, a customized algorithm is developed to target at discovering the approximate TF core motifs, with significant improvement over existing MEME. The approximate rules discovered are evaluated comprehensively with experiment-verified Protein Data Bank (PDB) data and exhibit a significant edge over the exact ones, with many more verified rules discovered and significant better verification ratios. The majority of the rules are also shown statistically significant (p-values < 0.05). Detailed analysis on PDB cases and conservation verification on NCBI protein records from other organisms illustrate that the approximate rules are important to better reveal the flexible and specific protein-DNA interactions. The approximate TF-TFBS rule discovery demonstrates great generalized capability of exploring more informative binding rules, and potential applications to predict protein-DNA interactions given either side for better deciphering of transcriptional regulation.

7.2 Future Work

With the current extensive efforts on TFBS motif discovery based on sequences, the future work is to incorporate more comprehensive informative data to improve the predictive power for identifying TFBSs, including expression data, phylogenetic information as well as possible protein features. The approximate TF-TFBS associated sequence patterns discovery serves as one such extension step and can be further applied to deciphering gene regulation. The future work is summarized as follows:

In-depth study and learning on TFBS motif models: Despite numerous motif discovery algorithms, the TFBS motif models are not yet fully understood and current models mainly concentrate on "conservation" and "over-representation". With the experience on TFBS motif discovery and comprehensive data of TRANSFAC, we will perform large-scale study on the TFBS data and try to obtain the comprehensive statistics to learn the appropriate TFBS motif model(s). Learning is one promising direction, where we have done some preliminary results using genetic programming to learn the TFBS motif scoring functions [58].

Incorporation of informative data for motif discovery: As the TFBS motif models being better studied and developed, additional informative data can be incorporated for more powerful prediction. Expression data which are widely employed, can be incorporated with our novel generalized models and/or spaced motif discovery for identifying TFBSs more accurately and comprehensively, via multi-variate regressions. Knowledge driven learning wil be the future trend.

Formal approximate TF-TFBS rules modeling: as mentioned before, the generalization on approximate TF-TFBS rules using a progressive approach is just the first step towards revealing the great potential of predicting and understanding the in-depth mechanisms of TF-TFBS sequence patterns. Future work includes formal models to score the TF-TFBS bindings sequence pattern as-a-whole, more advanced search algorithms based on the new scoring functions, associating multiple short TF motifs, as well as handling uncertainty such as unknown widths on both TF and TFBS sides.

Transcriptional regulatory network inference: with the patterns concerning transcriptional regulation discovered by our novel methods, we can further apply these patterns to predict more TF and/or TFBS binding relationship, and construct the transcriptional regulatory network based on the putative relationships. With common regulatory network inference data incorporated, e.g. expressions, more reliable and insightful transcriptional regulatory networks are to be discovered together with pattern discovery results. Large-scale putative gene networks are expected to be generated.

As more and more accurate biological data are available and more advanced computational approaches are being proposed, pattern discovery for deciphering transcriptional regulation will remain its fundamental role in bioinformatics. The proposed pattern discovery paradigms and approaches in this thesis, will be consistently improved and extended, and generate more promising outcomes and show better applicability with further novel paradigms and approaches by us in the future.

□ **End of chapter.**

Bibliography

[1] S. Ahmad, M. M. Gromiha, and A. Sarai. Analysis and prediction of dna-binding proteins and their binding residues based on composition, sequence and structural information. *Bioinformatics*, 20(4):477–486, March 2004.

[2] S. Ahmad, O. Keskin, A. Sarai, and R. Nussinov. Protein-DNA interactions: structural, thermodynamic and clustering patterns of conserved residues in DNA-binding proteins. *Nucleic Acids Res.*, 36:5922–5932, Oct 2008.

[3] A. Aronheim, R. Shiran, A. Rosen, and M. D. Walker. Cell-specific expression of helix-loop-helix transcription factors encoded by the E2A gene. *Nucleic Acids Res.*, 21(7):1601–1606, 1993.

[4] G. Badis, M. F. Berger, A. A. Philippakis, S. Talukder, A. R. Gehrke, S. A. Jaeger, E. T. Chan, G. Metzler, A. Vedenko, X. Chen, H. Kuznetsov, C.-F. Wang, D. Coburn, D. E. Newburger, Q. Morris, T. R. Hughes, and M. L. Bulyk. Diversity and complexity in dna recognition by transcription factors. *Science*, 324:1720–1723, 2009.

[5] T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36, 1994.

[6] P. Baldi and S. Brunak, editors. *Bioinformatics: the machine learning approach.* Cambridge, Mass. : MIT Press, 2001.

[7] A. L. Barabasi and Z. N. Oltvai. Network biology: Understanding the cell's functional organization. *Nat. Rev. Genet.*, 5:101–113, 2004.

[8] A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. GrifRths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, D. J. Studholme, C. Yeats, and S. R. Eddy. The pfam protein families database. *Nucleic Acids Res*, 32:D138–141, 2004.

[9] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucl. Acids Res.*, 28(1):235–242, January 2000.

[10] P. Bieganski, J. Riedl, J. V. Carlis, and E. Retzel. Generalized suffix trees for biological sequence data: applications and implementations. In *Proc. of the 27th Hawaii Int. Conf. on Systems Sci.*, pages 35–44, 1994.

[11] T. K. Blackwell and H. Weintraub. Differences and similarities in DNA-binding preferences of MyoD and E2A protein complexes revealed by binding site selection. *Science*, 250(4984):1104–1110, 1990.

[12] A. Blais and B. D. Dynlacht. Constructing transcriptional regulatory networks. *Genes and Dev.*, 19:1499–1511, 2005.

[13] E. Blanco, D. Farre, M. M. Alba, X. Messeguer, and R. Guigo. ABS: a database of annotated regulatory binding sites from orthologous promoters. *Nucleic Acids Res.*, 34:D63–D67, 2006.

[14] J. M. Bower and H. Bolouri, editors. *Computational Modeling of Genetic and Biochemical Networks.* Cambridge, Mass. : MIT Press, 2001.

[15] J. Buhler and M. Tompa. Finding motifs using random projections. In *RECOMB*, pages 69–76, 2001.

[16] S. C. Carmack, L. A. Mccue, L. A. Newberg, and C. E. Lawrence. Phyloscan: Identification of transcription factor binding sites using cross-species evidence. *Algorithms for Molecular Biology*, 2:1+, January 2007.

[17] T.-M. Chan, K.-H. Lee, K.-S. Leung, and P. Lio'. Generic spaced DNA motif discovery using genetic algorithm. In *2010 IEEE Congress on Evolutionary Computation*, page to appear, 2010.

[18] T.-M. Chan, K.-S. Leung, and K.-H. Lee. TFBS identification by position- and consensus-led genetic algorithm with local filtering. In *GECCO '07: Proceedings of the 2007 conference on Genetic and evolutionary computation*, pages 377–384, 2007.

[19] T.-M. Chan, K.-S. Leung, and K.-H. Lee. TFBS identification based on genetic algorithm with combined representations and adaptive post-processing. *Bioinformatics*, 24(3):341–349, 2008.

[20] T. M. Chan, G. Li, K. S. Leung, and K. H. Lee. Discovering multiple realistic tfbs motifs based on a generalized model. *BMC Bioinformatics*, 10(1):321+, October 2009.

[21] D. Che, Y. Song, and K. Rasheed. MDGA: motif discovery using a genetic algorithm. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 447-452, 2005.

[22] G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner. Weblogo: A sequence logo generator. *Genome Research*, 14(6):1188–1190, June 2004.

[23] E. D. Three-dimensional structure of membrane and surface proteins. *Ann. Rev. Biochem.*, 53:595–623, July 1984.

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

[25] A. E. Eiben and J. E. Smith. *Introduction to evolutionary computing*. Berlin; New York: Springer, 2003.

[26] E. Eskin and P. A. Pevzner. Finding composite regulatory patterns in DNA sequences. *Bioinformatics*, 18 Suppl 1, 2002.

[27] G. B. Fogel. Computational intelligence approaches for pattern discovery in biological systems. *Briefings in bioinformatics*, 9(4):307–316, July 2008.

[28] G. B. Fogel and D. W. Corne, editors. *Evolutionary Computation in Bioinformatics*. San Francisco, CA : Morgan Kaufmann, 2003.

[29] G. B. Fogel, D. G. Weekes, G. Varga, E. R. Dow, H. B. Harlow, J. E. Onyia, and C. Su. Discovery of sequence motifs related to coexpression of genes using evolutionary computation. *Nucleic Acids Res.*, 32(13):3826–3835, 2004.

[30] D. J. Galas and A. Schmitz. DNAse footprinting: a simple method for the detection of protein-DNA binding specificity. *Nucleic Acids Res.*, 5(9):3157–3170, September 1987.

[31] M. M. Garner and A. Revzin. A gel electrophoresis method for quantifying the binding of proteins to specific DNA regions: application to components of the escherichia coli lactose operon regulatory system. *Nucleic Acids Res.*, 9(13):3047–3060, July 1981.

[32] C. W. Garviea and C. Wolberger. Recognition of specific DNA sequences. *Molecular Cell*, 8:937 946, 2001.

[33] D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Boston, MA: Kluwer Academic Publishers, 1989.

[34] N. Habib, T. Kaplan, H. Margalit, and N. Friedman. A novel Bayesian DNA motif comparison method for clustering and retrieval. *PLoS Comput. Biol.*, 4(2):e1000010, 2008.

[35] G. Hertz and G. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7-8):563–577, 1999.

[36] J. H. Holland. *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press, 1975.

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

[38] J. Hu, Y. D. Yang, and D. Kihara. EMD: an ensemble algorithm for discovering regulatory motifs in DNA sequences. *BMC Bioinformatics*, 7:e342, 2006.

[39] N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, B. A. Cuche, E. de Castro, C. Lachaize, P. S. Langendijk-Genevaux, and C. J. A. Sigrist. The 20 years of prosite. *Nucl. Acids Res.*, 36(suppl_1):D245–249, January 2008.

[40] S. T. Jensen and J. S. Liu. BioOptimizer: a Bayesian scoring function approach to motif discovery. *Bioinformatics*, 20:1557–1564, 2004.

[41] S. T. Jensen, X. S. Liu, Q. Zhou, and J. S. Liu. Computational discovery of gene regulatory binding motifs: a bayesian perspective. *Statistical Science*, 19(1):188–204, 2004.

[42] H. Ji, H. Jiang, W. Ma, D. S. Johnson, R. M. Myers, and W. H. Wong. An integrated software system for analyzing chip-chip and chip-seq data. *Nature Biotechnology*, 26(11):1293–1300, 2008.

[43] S. Jones, H. P. Shanahan, H. M. Berman, and J. M. Thornton. Using electrostatic potentials to predict dna-binding sites on dna-binding proteins. *Nucleic Acids Res*, 31(24):7189–7198, December 2003.

[44] S. Jones, P. van Heyningen, H. M. Berman, and J. M. Thornton. Protein-dna interactions: a structural analysis. *Journal of Molecular Biology*, 287(5):877–896, April 1999.

[45] A. E. Kel, E. Goessling, I. Reuter, E. Cheremushkin, O. V. Kel-Margoulis, and E. Wingender. MATCH: a tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res.*, 31(13):3576–3579, 2003.

[46] A. E. Kel, O. V. Kel-Margoulis, P. J. Farnham, S. M. Bartley, E. Wingender, and M. Q. Zhang. Computer-assisted identification of cell cycle-related genes: new targets for E2F transcription factors. *J. Mol. Biol.*, 309(1):99–120, 2001.

[47] C. M. Klinge. Estrogen receptor interaction with estrogen response elements. *Nucleic Acids Res.*, 29:2905–2919, 2001.

[48] S. S. Krishna, I. Majumdar, and N. V. Grishin. Structural classification of zinc fingers: survey and summary. *Nucleic acids research*, 31(2):532–550, January 2003.

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

[50] A. B. Lassara, R. L. Davisa, W. E. Wrightb, T. Kadeschc, C. Murred, A. Voronovad, D. Baltimored, and H. Weintraub. Functional activity of myogenic HLH proteins requires hetero-oligomerization with E12/E47-like proteins in vivo. *Cell*, 58:305–315, 1991.

[51] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wooton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262(8):208–214, October 1993.

[52] K.-S. Leung, K.-C. Wong, T.-M. Chan, M.-H. Wong, K.-H. Lee, C.-K. Lau, and S. K. W. Tsui. Discovering protein-dna binding sequence patterns using association rule mining. *Nucleic Acids Res*, doi:10.1093/nar/gkq500:1–14, 2010.

[53] M. Li, B. Ma, and L. Wang. Finding similar regions in many sequences. *Journal of Computer and System Sciences*, 65:73–96, 2002.

[54] W. Li and A. Godzik. CD-HIT: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22:1658–1659, 2006.

[55] F. F. M. Liu, J. J. P. Tsai, R. M. Chen, S. N. Chen, and S. H. Shih. FMGA: finding motifs by genetic algorithm. In *BIBE '04: Proceedings of the 4th IEEE Symposium on Bioinformatics and Bioengineering*, pages 459–466, 2004.

[56] X. Liu, D. L. Brutlag, and J. S. Liu. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. In *Pac. Symp. Biocomput.*, volume 6, pages 127–138, 2001.

[57] X. S. Liu, D. L. Brutlag, and J. S. Liu. An algorithm for finding protein–DNA binding sites with applications to chromatinimmunoprecipitation microarray experiments. *Nat. Biotechnol.*, 20:835–839, 2002.

[58] L.-Y. Lo, T.-M. Chan, K.-H. Lee, and K.-S. Leung. Challenges rising from learning motif evaluation functions using genetic programming. In *GECCO '10: Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 171–178, New York, NY, USA, 2010. ACM.

[59] M. A. Lones and A. M. Tyrrell. The evolutionary computation approach to motif discovery in biological sequences. In *GECCO '05: Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pages 1–11, 2005.

[60] M. A. Lones and A. M. Tyrrell. A co-evolutionary framework for regulatory motif discovery. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 3894–3901, 2007.

[61] M. A. Lones and A. M. Tyrrell. Regulatory motif discovery using a population clustering evolutionary algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(3):403–414, 2007.

[62] N. M. Luscombe, S. E. Austin, H. M. Berman, and J. M. Thornton. An overview of the structures of protein-dna complexes. *Genome Biol.*, 1(1):REVIEWS001, June 2000.

[63] N. M. Luscombe, R. A. Laskowski, and J. M. Thornton. Amino acid-base interactions: a three-dimensional analysis of protein-dna interactions at an atomic level. *Nucleic Acids Res*, 29(13):2860–2874, July 2001.

[64] N. M. Luscombe and J. M. Thornton. Protein-dna interactions: amino acid conservation and the effects of mutations on binding specificity. *J Mol Biol*, 320(5):991–1009, July 2002.

[65] K. D. MacIsaac and E. Fraenkel. Practical strategies for discovering regulatory dna sequence motifs. *PLoS Comput Biol.*, 2(4):e36, 2006.

[66] S. W. Mahfoud. Crowding and preselection revisited. In *Parallel problem solving from nature 2*, pages 27–36. North-Holland, 1992.

[67] S. Mahony and P. V. Benos. STAMP: a web tool for exploring DNA-binding motif similarities. *Nucleic Acids Res.*, 35:W253–W258, 2007.

[68] Y. Mandel-Gutfreund and H. Margalit. Quantitative parameters for amino acid-base interaction: implications for prediction of protein-dna binding sites. *Nucleic Acids Res*, 26(10):2306–2312, May 1998.

[69] Y. Mandel-Gutfreund, O. Schueler, and H. Margalit. Comprehensive analysis of hydrogen bonds in regulatory protein dna-complexes: in search of common principles. *J Mol Biol*, 253(2):370–382, October 1995.

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

[71] K. A. Martin, K. Walsh, and S. L. Mader. The mouse creatine kinase paired E-box element confers muscle-specific expression to a heterologous promoter. *Gene*, 142:275–278, 1994.

[72] V. Matys, O. V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, N. Voss, P. Stegmaier, B. Lewicki-Potapov, H. Saxel, A. E. Kel, and E. Wingender. Transfac and its module transcompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, 34:108–110, 2006.

[73] J. Moore and E. Devaney. Cloning and characterization of two nuclear receptors from the filarial nematode Brugia pahangi. *Biochem. J.*, 344 Pt 1:245–252, Nov 1999.

[74] A. V. Morozov and E. D. Siggia. Connecting protein structure with predictions of regulatory sites. *Proc. Natl. Acad. Sci. U.S.A.*, 104(17):7068–7073, 2007.

[75] Y. Ofran, V. Mysore, and B. Rost. Prediction of dna-binding residues from sequence. *Bioinformatics*, 23(13):i347–353, July 2007.

[76] S. K. Pal, S. Bandyopadhyay, and S. S. Ray. Evolutionary computation in bioinformatics: a review. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(5):601–615, 2006.

[77] T. K. Paul and H. Iba. Identification of weak motifs in multiple biological sequences using genetic algorithm. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 271–278, 2006.

[78] G. Pavesi, G. Mauri, and G. Pesole. An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics*, 17:S207–S214, 2001.

[79] 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 Res.*, 32:W199–W203, 2004.

[80] P. A. Pevzner and S.-H. Sze. Combinatorial approaches to finding subtle signals in DNA sequences. In *Proceedings International Conference on Intelligent Systems for Molecular Biology*, pages 269–278. AAAI Press, 2000.

[81] T. H. Pham, J. C. Clemente, K. Satou, and T. B. Ho. Computational discovery of transcriptional regulatory rules. *Bioinformatics*, 21(2):101–107, 2005.

[82] B. Raphael, L.-T. Liu, and G. Varghese. A uniform projection method for motif discovery in DNA sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(2):91–94, April-June 2004.

[83] Research Collaboratory For Structural Bioinformatics. RCSB PDB Annual Report July 2009. http://www.rcsb.org/pdb/, December 2009.

[84] F. Roth, J. Hughes, P. Estep, and G. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.*, 16:939–945, 1998.

[85] M. F. Sagot. Spelling approximate repeated or common motifs using a suffix tree. In *LATIN'98, LNCS 1380*, pages 374–390, 1998.

[86] A. Sandelin, W. Alkema, P. Engstrom, W. Wasserman, and B. Lenhard. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.*, 32:D91–94, Jan 2004.

[87] G. K. Sandve, O. Abul, V. Walseng, and F. Drablos. Improved benchmarks for computational motif discovery. *BMC Bioinformatics*, 8(1):193, 2007.

[88] A. Sarai and H. Kono. Protein-dna recognition patterns and predictions. *Annu Rev Biophys Biomol Struct*, 34:379–398, 2005.

[89] E. W. Sayers, T. Barrett, D. A. Benson, E. Bolton, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. Dicuccio, S. Federhen, M. Feolo, L. Y. Geer, W. Helmberg, Y. Kapustin, D. Landsman, D. J. Lipman, Z. Lu, T. L. Madden, T. Madej, D. R. Maglott, A. Marchler-Bauer, V. Miller, I. Mizrachi, J. Ostell, A. Panchenko, K. D. Pruitt, G. D. Schuler, E. Sequeira, S. T. Sherry, M. Shumway, K. Sirotkin, D. Slotta, A. Souvorov, G. Starchenko, T. A. Tatusova, L. Wagner, Y. Wang, W. John Wilbur, E. Yaschenko, and J. Ye. Database resources of the national center for biotechnology information. *Nucleic acids research*, 38(Database issue):D5–16, January 2010.

[90] W. M. Shaw, R. Burgin, and P. Howell. Performance standards and evaluations in ir test collections: cluster-based retrieval models. *Inf. Process. Manage.*, 33:144, 1997.

[91] R. Siddharthan, E. D. Siggia, and E. van Nimwegen. PhyloGibbs: a Gibbs sampling motif finder that incorporates phylogeny. *PLoS Comput. Biol.*, 1(7):e67, 2005.

[92] S. Sinha and M. Tompa. Ymf: a program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucl. Acids Res.*, 31(13):3586–3588, July 2003.

[93] A. D. Smith, P. Sumazin, D. Das, and M. Q. Zhang. Mining ChIP-chip data for transcription factor and cofactor binding sites. *Bioinformatics*, Suppl 1(20):i403–i412, 2005.

[94] M. Stine, D. Dasgupta, and S. Mukatira. Motif discovery in upstream sequences of coordinately expressed genes. In *CEC '03: Evolutionary Computation, The 2003 Congress on*, volume 3, pages 1596–1603, 2003.

[95] M. Stine, D. Dasgupta, and S. Mukatira. Motif discovery in upstream sequences of coordinately expressed genes. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, volume 3, pages 1596–1603 Vol.3, 2003.

[96] G. D. Stormo. Computer methods for analyzing sequence recognition of nucleic acids. *Annu. Rev. BioChem.*, 17:241–263, 1988.

[97] G. D. Stormo and G. W. Hartzell. Identifying proteinbinding sites from unaligned DNA fragments. *Proc. Natl. Acad. Sci. U.S.A.*, 86:1183–1187, 1989.

[98] G. Thijs, K. Marchal, M. Lescot, S. Rombauts, B. DeMoor, P. Rouze, and Y. Moreau. A gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *J. Comput. Biol.*, 9:447–464, 2002.

[99] M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. D. Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, V. J. Makeev, A. A. Mironov, W. S. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. van Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, , and Z. Zhu. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23(1):137–144, 2005.

[100] D. Wang and X. Li. GAPK: genetic algorithms with prior knowledge for motif discovery in DNA sequences. In *CEC'09: Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, pages 277–284, Piscataway, NJ, USA, 2009. IEEE Press.

[101] Z. Wei and S. T. Jensen. GAME: detecting cis-regulatory elements using a genetic algorithm. *Bioinformatics*, 22(13):1577–1584, 2006.

[102] B. L. Welch. The generalization of "student's" problem when several different population variances are involved. *Biometrika*, 34:28–35, 1947.

[103] E. Wijaya, K. Rajaraman, S.-M. Yiu, and W.-K. Sung. Detection of generic spaced motifs using submotif pattern mining. *Bioinformatics*, 23(12):1476–1485, 2007.

[104] E. Wijaya, S.-M. Yiu, N. T. Son, R. Kanagasabai, and W.-K. Sung. MotifVoter: a novel ensemble method for fine-grained integration of generic motif finders. *Bioinformatics*, 24(20):2288–2295, 2008.

[105] G. P. Zambetti, J. Bargonetti, K. Walker, C. Prives, and A. J. Levine. Wild-type p53 mediates positive regulation of gene expression through a specific DNA sequence element. *Genes Dev.*, 6:1143–1152, 1992.

[106] J. Zhao, F. I. Schmieg, D. T. Simmons, and G. R. Molloy. Mouse p53 represses the rat brain creatine kinase gene but activates the rat muscle creatine kinase gene. *Mol. Cell. Biol.*, 14(12):8483–8492, 1994.

[107] Q. Zhou and J. S. Liu. Extracting sequence features to predict protein-dna interactions: a comparative study. *Nucl. Acids Res.*, 36(12):4137–4148, July 2008.

Appendix A

Publications and Awards

Summary

Refereed publications including manuscripts in preparation, and research awards.

A.1 Refereed Publications:

- **Tak-Ming Chan**, Ka-Chun Wong, Kin-Hong Lee, Man-Hon Wong, Chi-Kong Lau, Stephen Kwok-Wing Tsui, Kwong-Sak Leung, Discovering Approximate Associated Sequence Patterns for Protein-DNA Interactions. Under review for Bioinformatics.

- Chi-Fai Lam, **Tak-Ming Chan**, Leung-Yau Lo, Kin-Hong Lee, Stephen Kwok-Wing Tsui, Kwong-Sak Leung, On Intra and Inter Disimilarities of TFBS Motifs. In preparation.

- Kwong-Sak Leung, Ka-Chun Wong, Tak-Ming Chan, Man-Hon Wong, Kin-Hong Lee, Chi-Kong Lau, Stephen Kwok-Wing Tsui, Discovering Protein-DNA Binding Sequence Patterns Using Association Rule Mining. Nucleic Acids Research (IF: 7.479), to appear.

- **Tak-Ming Chan**, Kin-Hong Lee, Kwong-Sak Leung, Pietro Lio', Generic Spaced DNA Motif Discovery Using Genetic Algorithm. In Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC 2010), pp. 2647-2654.

- Leung-Yau Lo, **Tak-Ming Chan**, Kin-Hong Lee, Kwong-Sak Leung, Challenges Rising from Learning Motif Evaluation Functions Using Genetic Programming. In GECCO 10: Proceedings of the 2010 conference on Genetic and evolutionary computation, pp. 171-178.

- **Tak-Ming Chan**, Gang Li, Kwong-Sak Leung, Kin-Hong Lee, Discovering multiple realistic TFBS motifs based on a generalized model. BMC Bioinformatics (IF: 3.428), 2009, 10: 321

- Gang Li, **Tak-Ming Chan**, Kwong-Sak Leung and Kin-Hong Lee, A Cluster Refinement Algorithm for Motif Discovery. IEEE/ACM Transaction on Computational Biology and Bioinformatics (IF: 2.246), 2010, in press.

- Gang Li, **Tak-Ming Chan**, Kwong-Sak Leung and Kin-Hong Lee, An Estimation of Distribution Algorithm for Motif Discovery. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC 2008), IEEE Press, 2008, pp. 2416-2423.

- **Tak-Ming Chan**, Kwong-Sak Leung, Kin-Hong Lee, TFBS identification based on genetic algorithm with combined representations and adaptive post-processing. Bioinformatics (IF: 4.926), 2008, 24(3), pp. 341-349.

- **Tak-Ming Chan**, Kwong-Sak Leung, Kin-Hong Lee, TFBS identification by position- and consensus-led genetic algorithm with local filtering. In GECCO 07: Proceedings of the 2007 conference on Genetic and evolutionary computation, pp. 377-384.

- **Tak-Ming Chan**, Junping Zhang, Jian Pu, Hua Huang, Neighbor Embedding Based Super-Resolution Algorithm Through Edge Detection and Feature Selection. Pattern Recognition Letters, 2009, 30(5), pp. 494-502. (Extension on the undergraduate thesis)
- **Tak-Ming Chan**, Junping Zhang, An Improved Super-Resolution with Manifold Learning and Histogram Matching. In Proceedings of IAPR International Conference on Biometric (ICB-2006), pp. 756-762. (Undergraduate research topic)

A.2 Research Awards

- Best Teaching Assistant Award, Department of Computer Science and Engineering, The Chinese University of Hong Kong, 2009-2010
- Best Poster Award (1st Place), ACM-HK Bioinformatics Symposium, 2010 (GALF-G)
- Sponsorship Scheme for I.T. Exchange Programmes 2008-09 (research visit to Computer Laboratory, University of Cambridge), Hong Kong Cyberport, 2009 (GASMEN)
- Merit Award, ACM-HK Student Research and Career Day, 2009 (Preliminary GALF-G)
- Champion (1st prize) of IEEE (HK) Computational Intelligence Chapter Postgraduate Paper Competition, 2008 (GALF-P)