Discovery of Flexible Gap Patterns from Sequences

by

En Hui Zhuang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2014

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Human genome contains abundant motifs bound by particular biomolecules. These motifs are involved in the complex regulatory mechanisms of gene expressions. The dominant mechanism behind the intriguing gene expression patterns is known as combinatorial regulation, achieved by multiple cooperating biomolecules binding in a nearby genomic region to provide a specific regulatory behavior. To decipher the complicated combinatorial regulation mechanism at work in the cellular processes, there is a pressing need to identify co-binding motifs for these cooperating biomolecules in genomic sequences. The great flexibility of the interaction distance between nearby cooperating biomolecules leads to the presence of flexible gaps in between component motifs of a co-binding motif.

Many existing motif discovery methods cannot handle co-binding motifs with flexible gaps. Existing co-binding motif discovery methods are ineffective in dealing with the following problems: (1) co-binding motifs may not appear in a large fraction of the input sequences, (2) the lengths of component motifs are unknown and (3) the maximum range of the flexible gap can be large. As a result, the probabilistic approach is easily trapped into a local optimal solution. Though deterministic approach may resolve these problems by allowing a relaxed motif template, it encounters the challenges of exploring an enormous pattern space and handling a huge output.

This thesis presents an effective and scalable method called DFGP which stands for “Discovery of Flexible Gap Patterns” for identifying co-binding motifs in massive datasets. DFGP follows the deterministic approach that uses flexible gap pattern to model co-binding motif. A flexible gap pattern is composed of a number of boxes with a flexible gap in between consecutive boxes where each box is a consensus pattern representing a component motif. To address the computational challenge and the need to effectively process the large output under a relaxed motif template, DFGP incorporates two redundancy reduction methods as well as an effective statistical significance measure for ranking patterns. The first reduction method is achieved by the proposed concept of representative patterns, which aims at reducing the large set of consensus patterns used as boxes in existing deterministic methods into a much smaller yet informative set. The second method is attained by the proposed concept of delegate occurrences aiming at reducing the redundancy among occurrences of a flexible gap pattern.

Extensive experiment results showed that (1) DFGP outperforms existing co-binding discovery methods significantly in terms of both the capability of identifying co-binding motifs and the runtime, (2) co-binding motifs found by DFGP in datasets reveal biological insights previously unknown, (3) the two redundancy reduction methods via the proposed concepts of representative patterns and delegate occurrences are indeed effective in significantly reducing the computational burden without sacrificing output quality, (4) the proposed statistical significance measures are robust and useful in ranking patterns and (5) DFGP allows a large maximum distance for flexible gap between component motifs and it is scalable to massive datasets.

Acknowledgements

First and foremost, my deepest gratitude and highest praise to my Lord and Savior Jesus Christ for his unconditional love, faithfulness and amazing grace, walking with me and helping me through many difficult struggles in life both inside and outside of the academy.

My deep gratitude also goes to my supervisor, Professor Andrew K.C. Wong, for his guidance, encouragement and heartfelt support during the whole journey of the Ph. D. research program. His persistence and passion in tackling research problems and in pursuing the academic excellence inspire me to take and press through challenges. My sincere gratitude also goes to my co-supervisor, Professor Daniel Stashuk, for his support on my Ph. D. study and his constructive feedback to my thesis.

I would like to express my sincere thanks to Xiaohua Hu, my external examiner from Drexel University and other examining committee members: Professor Paul Fieguth, Professor Hamid Tizhoosh, Professor Kumaraswamy Ponnambalam of Systems Design Engineering, and Professor Forbes Burkowski for their support on my Ph. D. and helpful comments for improving my thesis. Especially, I would like to once again thank Professor Forbes Burkowski for his support to provide references to my OGS applications and his detailed and careful read of my thesis to correct typos and grammatical errors.

I would like to greatly thank my good friend and brother in Christ, Gary Li. He introduced my supervisor Dr. Wong to me, opening an avenue to my Ph. D. study. He gave me valuable research advice, inspired me through thoughtful discussions and helped me revising research papers and this thesis. He also recommended me to my current company KFL capital management that provided me full support and resources for my thesis research. My thanks also go to the CEO Dave Sanderson and the head of technology Harsch Khandelwal in KFL.

I would like to thank my brothers and sisters in KWCAC fellowship: Alice Sze, Andy Huang, Angela Fan, Athena Tam, Charles Fung, Crystal Wen, Rocky Cui, Eric Chan, Erwin Chang, Fanny Luk, Florence Ngan, Franki Kung, Hammond Lo, Hannah Kwan, Hansen Lau, Jacqueline Chan, Jeffery Ng, Jessey Lin, Jimmy Chow, Johathan Ku, Justin Wen, Karen Xian, Kelvin Leung, Kelvin Poon, Kenneth Chan, Kenneth Ng, Kexin Ji, Koko Lung, Larissa Wong, Long Shun Cheng, Matthew Wong, Michael Ngelo, Miyuki Tsukimoto, Rebecca Kwan, Ricky Ngan, Robert Chan, Rosita Kwan, Sabrina Ngai, Shuman Ip, Simon Chan, Solomon Xu, Synergy Shum, Szeching Chang, Taoran Lin,

Thomson Wong, Tommy Cheng, Tony Ho, Tony Ip, Turbo Lee, Virginia Wong, Winnie Chan, Xin Wen and Yolanda Tang. Thank you for your friendship, encouragement and prayers. I really enjoyed with you, the fellowship program in each Friday, the moments of fellowship outing and retreat, the joyful moments of playing board games and computer games, watching movies and dining out, the moments of academic and theological discussions, the moments of bible studies and the moments of serving our God together. I would also like to thank Pastor Vincent Chan and my fellowship counselors Ardy and Geraldine Chan for their love, care and spiritual guidance during my Ph. D. study.

My parents deserve my special and deep gratitude. I am so grateful to have such loving parents. They have continued to provide care, encouragement, guidance and advice to me along my life. I would also like to thank my siblings for their warm support, care and constant love over the years.

Finally, my very sincere thanks go to my love, Jessica Wong. I am thankful that God brings her into my life. She accompanied me and provided continuous support during many nights of my thesis writing and in my preparation of the Ph. D. oral defense. She always draws me to God when I face challenges and difficulties, and loves me as who I am. I am so grateful and enjoy every moments with her.

Dedication

To My Lord

Table of Contents

List of Figures

List of Tables

List of Definitions

Chapter 1
Introduction

1.1 An Overview

Many whole-genome assemblies for both simple and complex organisms have been sequenced, and comparative genome analyses for them revealed that the increase of the number of genes in the genome does not substantially accounted for the increase of organism complexity as one might expect [92]. For example, the human genome is estimated to contain only around 25,000 protein coding genes, which is about the same as that for corn and is nearly twice more than that for fruit fly, but Homo Sapiens exhibit far more complex gene expression patterns in the developmental processes and during responses to the stimulus from external environments. So, what are the mechanisms at work behind these intriguing gene expression patterns in complex organisms, making a eukaryote different from a prokaryote, or a human from a yeast? The answer lies in the non-coding genomic regions. Indeed, the coding regions consists of only 2% of the human genome. The remaining 98% of the genome harbors a vast number of regulatory motifs for the delicate and precise control of gene expressions.

One of the most important genomic regulatory motifs in both prokaryotic and eukaryotic cells [93] is the transcription factor binding site (TFBS). Transcription factors (TF) are regulatory proteins to activate or inhibit the transcription of genes from the genome. The transcription process begins with the binding of RNA polymerase complex to an upstream region of the gene known as the promoter and then is followed by the elongation of the transcript. In prokaryote, one single activator or repressor together with a general specificity factor called sigma factor are often sufficient for transcriptional regulation of a gene [94]. Sigma factors help RNA polymerase to achieve high affinity in binding to the specific promoter regions. In eukaryote, the transcription process is much more complicated, and requires coordinated interactions of multiple proteins, the phenomenon known as combination regulation. More specifically, the transcription initialization involves binding of multiple transcription factors in a series of interactions to the promoter region to ultimately form a transcription complex that facilitates binding and transcription by RNA polymerase.

The existence of distal regulatory sequences outside of the promoter region of a gene such as enhancers, silencers, insulators [95], also adds great flexibility for regulating the transcription process in more complex organisms. For example, there is no known enhancers found in yeast genome while many human genes are greatly influenced by their associated enhancers. These distal regulatory motifs can be far from the transcription start site, sometimes thousands kilo bases away from a promoter, and they can be in the downstream or upstream or introns. Though they might be very far from the gene they regulates in terms of nucleotides, due to the structure of the chromatin complex of DNA, they can be spatially in close proximity to the promoter and the gene, where trans-acting transcription factors can bind to them to either activate or repress the transcription process.

In recent years, it is also found that non-coding RNAs (ncRNA) does not perform merely as a messenger between gene and protein but could play a significant regulatory role in the cellular developmental and differential processes [96] [97]. For example, an important class of these regulatory ncRNAs is called large ncRNA (lncRNA) [98]. As suggested by Wang et al. [17], an lncRNA can act a guide that can bind to proteins and direct their location to target genes either in cis or in trans by binding to the target DNA sequences.

The addition layer of the regulation comes from the accessibility of the chromatin state of the DNA. Chromatin are made up of histone proteins into which DNA is packaged. DNA methylation and Histone modification through methylation and acetylation can serve as epigenetic markers for transcriptional activeness or silencing [99]. Through epigenetic markers, gene expression changes can be achieved without changes in DNA sequences. For example, DNA methylation of CpGs in the promoters of genes [100] [101], the tri-methylation of lysine 27 and lysine 9 of histone H3 (H3K27me3 and H3K9me3) [102] are often associated with transcriptional silencing. In general, Epigenetic markers can be maintained for genes after cell division. Hence, they are important in controlling the cell identity. How these epigenetic markers are established during epigenetic reprogramming in germ-cell development and how they are modified during cellular differentiation are still partially understood. However, recent researches suggest that regulatory motifs in sequences around these markers might influence their modifications. For example, as suggested in [103] [104], sequence features around CpG islands contribute to the determination of their methylation state.

Pervasive regulatory motifs in the DNA sequences, such as TFBSs, enhancers, silencers, insulators, ncRNA binding sites and recruiters for epigenetic marker modifications, form an enormously complex regulatory network, enabling complex organisms to control and maintain precise patterns of gene expression in different cells under different external environments. The identification of regulatory motifs in DNA sequences serves as an important step towards deciphering the regulatory programme encoded in the genome, the blueprint of life, and better understanding the pathology and cancer mechanisms. A recent science paper [1] conducted a large scale study of 88 cancer genomes and identified 98 non-coding candidate drivers in causing cancers, in which 58 of them break transcription factor (TF) binding motifs, causing the loss of the binding function.

Existing experimental techniques such as Electrophoretic Mobility Shift Assay (EMSA) and DNA Pull-Down Assay are essentially impractical to identify these regulatory motifs in DNA sequences [6] [7], whose length is around 6 to 15 base pairs (bp) in midst of a great amount of non-coding genomic sequences. Conversely, high-throughput sequencing and experimental techniques have been becoming a source of great value for identifying regulatory motifs by producing massive sequence data with potential binding sites of target biomolecules. Up-to-date, broad genomic regions with length typically varying from 100 to 500 bp, potentially bound by target biomolecules (i.e. protein, protein complex and non-coding RNA), can be obtained. For example, Chromatin Immunoprecipitation sequencing (ChIP-seq) was used to extract genomic regions (referred to as peaks) likely bound by TFs in [2] [3] [4]. Chromatin Isolation by RNA Purification sequencing (ChIRP-seq) was used to detect genomic regions likely bound by a non-coding RNA (ncRNA) in [5]. Next-generation sequencing can also produce genomic maps of DNA methylation (ME-DIPseq), chromatin accessibility to TFs [105], or other epigenetic factors involved in repressing and activating gene transcription. Accordingly, motif discovery, the computational approach, has gradually emerged as a crucial tool for exploring binding motifs of biomolecules from these rich data resources.

Most motif discovery methods [8] [9] focus on discovering simple motifs and rigid gapped motifs. Rigid gapped motifs is composed of spaced simple motifs of fixed distance. Simple motifs are often targeted by biomolecules with only one DNA binding domain. On the contrary, rigid gapped motifs are often targeted by a biomolecule complex with physically rigid constrained binding domains such as homodimers or heterodimers, resulting in a rigid distance between their corresponding simple motifs. These binding domains are indispensable for the biomolecule complex to attain its intended function. Discovery methods for identifying simple motifs and rigid gapped motifs have achieved a certain level of success and are widely used by biologists nowadays. For instance, Chu et al. [5] used MEME [10] to find the simple binding motif for an ncRNA called HOTAIR in exploring its relation with PRC2 (ploycomb repressive complex 2). Wei et al. [11] used GLAM [38] to obtain a rigid gapped motif for a homodimer formed by the oncogenic protein TP53. Kunarso et al. [4] applied MDmodule [63] to ChIP-seq datasets for TF OCT4 and found a rigid gapped motif for a heterodimer formed by OCT4 and its binding partner SOX2, which are critical in maintaining the pluripotent state in stem cells.

The fundamental assumption behind these simple motif and rigid gapped motif discovery methods is that biomolecules responsible for these motifs bind independently to their target genomic sites and do not interact with other biomolecules binding nearby [12]. Under such assumption, it was observed that many predicted motif sites are false positives and therefore not functional in vivo [9] [13]. Biologists now agree to a more accurate picture that biomolecules seldom act alone but cooperatively through binding to a nearby genomic region to achieve a specific regulatory behavior.

The cooperative behavior of biomolecules may repress their target gene expression while each biomolecule alone induces gene expression [14], illustrating an intriguing aspect of combinatorial regulation. The general consensus is that combinatorial regulation is the dominant mechanism behind the observed complex gene expression patterns [15] and is one of the major contributors in the cellular developmental and differentiation processes [16]. Furthermore, cooperating biomolecules are not restricted to only proteins. The discovery of significant regulatory roles of ncRNAs adds one more layer of complexity in the combinatory regulation network [17]. To achieve specific regulatory role, cooperating biomolecules can form homodimers or heterodimers, or be parts of an even larger regulatory machinery through protein-protein and protein-RNA interactions.

To decipher the intriguingly complex combinatorial regulation at work in the cellular processes, there is a pressing need to identify co-binding motifs for these cooperating biomolecules in genomic sequences. In recent years, the field of motif discovery has therefore shifted to co-binding motif discovery. It attempts to uncover the combinatorial regulatory codes which are otherwise unattainable by simple motif and rigid gapped motif discovery [18] [19] [20] [21]. However, despite the considerable amounts of efforts that has been devoted to tackle this problem, co-binding motif discovery in large datasets (i.e., ChIP-seq datasets) still remains very challenging.

This thesis aims at developing an effective and scalable computational method to discover co-binding motifs from large DNA sequence datasets. A co-binding motif as shown in Figure 1 is composed of a number of simple component motifs with a flexible gap in between adjacent components. As opposed to rigid gapped motifs, the gap in co-binding motifs is flexible, reflecting the great flexibility of the interaction distance between cooperating biomolecules. In other words, the interaction distance is allowed to be within a maximum threshold but is not fixed.

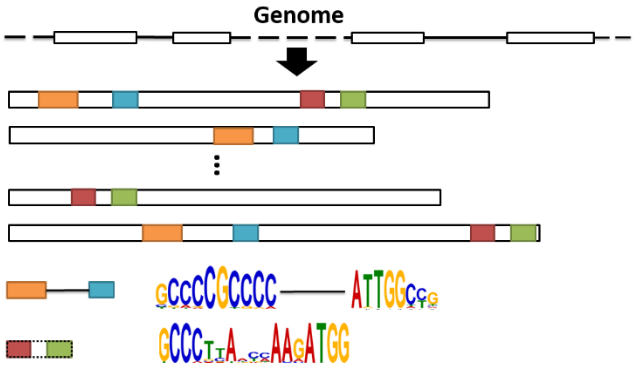


Figure 1. A co-binding motif in DNA sequences. A white box is a genomic region of interest. A colored box represents a simple binding motif of a biomolecule. The two boxes connected by a line is a co-binding motif, the line in between simple motifs represents a flexible gap. The two boxes in the box with dashed line represents a rigid gapped motif.

1.2 Limitations of Existing Methods

Existing co-binding motif discovery methods can be mainly divided into two approaches. One is the probabilistic approach [64] [65] [66] and the other is the deterministic approach [70] [71] [72]. However, both approaches are ineffective to deal with the following difficulties encountered in co-binding motif discovery due to the fact that (1) co-binding motifs may not appear in a large fraction of the input sequences, (2) the lengths of component motifs are unknown and (3) the maximum range of the flexible gap can be large.

The probabilistic approach represents the co-binding motif to be found using a probabilistic model and attempts to obtain parameter values maximizing the model likelihood through Gibbs sampling. However, its nature of being easily trapped into a local optimal solution makes it incapable of handling the above difficulties. It may work properly only if the co-binding motif to be found is highly enriched in the dataset with the lengths of its component motifs and the flexible gap range roughly known. In addition, it can model co-binding motifs consisting of only two components.

Conversely, the deterministic approach models a co-binding motif through a flexible gap pattern, which consists of a number of boxes with a flexible gap between every two consecutive boxes. A box represents a consensus pattern, which is essentially a string but with a certain number of mismatches allowed when matching the string in the input sequences to find its occurrences. The allowed mismatches increases the elasticity of each box for capturing a simple motif. The deterministic approach then exhaustively enumerates all flexible gap patterns occurring in at least a fraction of sequences above a frequency threshold specified. After the pattern extraction phase, patterns are ranked based on their statistical significance. The top ranked patterns are considered as candidate co-binding motifs.

The exhaustive manner of the deterministic approach guarantees that co-binding motifs in the dataset are unlikely to be missed in the output of flexible gap patterns. However, this guarantee requires the condition of allowing a low frequency threshold, a relaxed range of box length and a large range for the flexible gap. Using such a relaxed motif template, the above mentioned difficulties (1)-(3) can be addressed. However, the pattern extraction phase can take considerable time even by the best method RISOTTO [70]. Furthermore, the Monte Carlo method [68] used for computing pattern statistical significance would take an unmanageable amount of time. The huge computation time for assessing pattern statistical significance is due to the gigantic output resulting from the combinations of boxes.

Consider, for example, the case of discovering flexible gap patterns of two boxes in a dataset of a moderate size (500 sequences and 150546 bp in total). One mismatch is allowed for each box. Setting each box length to 6, the maximum distance for the flexible gap to 30 and the threshold to 10% of sequences, RISOTTO outputs 1976770 patterns in 71 seconds and the ranking of these patterns took 2.72 days. Under a more relaxed setting that allows box length to be from 4 to 12 and decreases the threshold to 5%. RISOTTO took 1.75 hours to generate 59730851 patterns, letting alone the time for computing statistical significances for ranking patterns.

It is clear that the deterministic approach suffers from either a need of exploring an enormous pattern space as well as handling a huge output or an unsatisfactory performance when the specific knowledge of the co-binding motifs is not available. In addition, it is not obvious that the top ranked patterns rendered by the Monte Carlo method is effective to capture co-binding motifs in the case of huge output size. Hence, the effectiveness of existing deterministic methods is greatly impaired by the computational burden and the lack of an effective method for handling a huge output.

1.3 Contributions of the Thesis

This thesis presents an effective and scalable method DFGP (Discovery of Flexible Gap Patterns) for identifying co-binding motifs in massive datasets, meeting the need for such a method in the research area of motif discovery. DFGP follows the deterministic approach to generate flexible gap patterns but resolves the tradeoff between the need to overcome the inevitable difficulties in the co-binding motif discovery and the need to mitigate the great computational burden for pattern extraction and ranking.

To overcome the difficulties posed, DFGP uses a relaxed motif template, imposing no restriction over the length of component motifs, as well as allowing large maximum distance for the flexible gap and low frequency threshold. To handle the tremendous computational burden and the huge output due to such relaxed template, this thesis proposes two redundancy reduction methods for drastically reducing the enormous pattern space to search while maintaining the output quality. One is the method for obtaining representative patterns (Chapter 3) and the other is the method for extracting delegate occurrences (Chapter 4).

The concept of representative patterns aims at reducing the large set of consensus patterns used as boxes for generating flexible gap patterns into a much smaller yet informative set. Such reduction leads to the exclusion of a large portion of pattern space to search but reduces the risk of missing important consensus patterns that are part of a co-binding motif.

The concept of delegate occurrences aims at reducing the positional redundancy among occurrences of a flexible gap pattern. It is based upon the observation that the presence of the flexible gap creates many occurrences that encompass other occurrences within themselves. The exclusion of these encompassing occurrences leads to a subset of occurrences called delegate occurrences. Counting only delegate occurrences further improves the pattern discovery process and makes the runtime of DFGP independent of the maximum allowed distance and the number of boxes in a flexible gap pattern. The extraction of only delegate occurrences during the construction of a flexible gap pattern enables DFGP to explore long-range interactions among multiple biomolecules.

Existing deterministic methods use the Monte Carlo method to estimate the expected number of sequences for a given flexible gap pattern. It generates a set of randomly shuffled sequence datasets of the same input size to achieve the estimation. This is impractical when the number of output patterns is large. To address this issue, this thesis develops two statistical significance measures (Chapter 5) for ranking patterns. The computation of the two measures does not add additional complexity into the pattern discovery process. The first measure accounts for the flexible gap pattern with a complete set of occurrences and the second one accounts for delegate occurrences. Both measures are effective in ranking patterns.

The effectiveness and scalability of DFGP are achieved by integrating the method for extracting representative patterns; the method for identifying delegate occurrences in the pattern discovery process; and the statistical significance measure introduced. Extensive experiment results have shown that (1) DFGP outperforms existing co-binding discovery methods significantly in terms of both the capability of identifying co-binding motifs and the runtime; (2) co-binding motifs found by DFGP in datasets reveal previously unknown biological insights; (3) the two proposed redundancy reduction methods for obtaining representative patterns and delegate occurrences respectively are effective in greatly reducing the computational burden without sacrificing output quality; (4) the proposed statistical significance measures are useful in ranking patterns and (5) DFGP allows large maximum distance for the flexible gap between component motifs and (6) DFGP is scalable to massive datasets.

1.4 Organization of the Thesis

There are seven chapters in this thesis including this introduction.

Chapter 2 provides a brief review of existing ideas relevant to motif discovery in biological sequences. Discussions of individual methods follow an overview of motif types and the presentation of two general approaches for motif discovery. The advantages and limitations of these methods pertaining to the focus of this thesis are examined, and state-of-the-art methods are identified for experimental comparisons. Preliminary definitions and terminologies that will be used throughout the rest of the thesis are presented.

Chapter 3 starts with a discussion of the problem concerning the overwhelming number of flexible gap patterns due to the exhaustive use of all available consensus patterns as boxes combinations. It then presents the rationale of using maximal solid patterns (a well-known pattern redundancy concept) instead of all consensus patterns as a starting point to narrow down the search. A linear time algorithm based upon suffix tree to extract these maximal patterns is developed. Furthermore, the insufficiency of maximal patterns to overcome this problem alone leads to the proposed redundancy reduction concept of representative patterns and the development of the method DRP (Discovery of Representative Patterns) for extracting these patterns. DRP achieves a much more compact yet informative set of consensus patterns to be used for constructing flexible gap patterns.

Chapter 4 first presents a straightforward method to obtain the complete set of occurrences for a flexible gap pattern. A formal analysis of the size of the complete set leads to the observation of the redundancy among occurrences of a flexible gap pattern. The concept of delegate occurrences is then proposed to address this redundancy issue and the method DOA (Delegate Occurrences Assembling) is developed to efficiently generate a flexible gap pattern while counting only delegate occurrences.

Chapter 5 introduces the two novel measures for assessing statistical significances of flexible gap patterns, and then presents DFGP that integrates DRP, DOA and the statistical significance measure. It ends with the discussion of the parameter settings and the runtime analysis for DFGP.

Chapter 6 presents the experiment results. The proposed method is applied to 68 ChIP-seq datasets, a subset of 457 ChIP-seq datasets on 119 human TFs generated by ENCODE Consortium. A computational pipeline developed in a recent genome research paper [2] proposed that each of the 68 datasets contains some co-binding motifs formed by the canonical motif for the ChIP'ed TF and a noncanonical motif for a cooperating partner TF. The experiments compare DFGP with the state-of-the-art methods RISOTTO, Bioprospector and GLAM2 for co-binding motif discovery as well as other well-known methods MEME, MEME-Chip, MDmoudle and Weeder for simple motifs as well as rigid gapped motifs discovery. The performance is evaluated in terms of the ability of finding co-binding motifs and also the computational runtime. Co-binding motifs found by DFGP are compared with those proposed by the computational pipeline. Experiments were also conducted to investigate the effect of the proposed two redundancy reduction methods. Experiment results for DFGP under different parameter settings and its scalability to massive datasets are also presented.

Chapter 2
Review of Related Works

2.1 An Overview of Motif Discovery

Methods for discovering regulatory motifs in DNA sequences have undergone tremendous developments over the last decade [8] [9]. A motif is the description for a particular functional unit [22]. An important type of motifs in the genomic sequences is the binding motif for regulatory proteins and ncRNAs such as TF binding sites (TFBS) [23]. Binding sites of a biomolecule often have variations yet still retain the same biological function [24]. De novo motif discovery is the task of identifying over-represented motifs in a set of sequences without using other additional information. There are also many methods that integrate other external evidences attempting to improve the motif discovery performance. For example, non-coding genomic regions well conserved across different species are likely to be functional regions that contain motifs and hence this information is utilized by the Phylogenetic footprinting approach [8] [23]. Some methods integrate the microarray gene expression data into motif discovery to find motifs that are likely to be major contributors to the expression of their regulated genes [63] [87]. There are also some motif search or localization methods [82] [88] that find the occurrences for a given motif. This thesis focuses on de novo motif discovery only. First, the definition of sequence is formally introduced.

Definition 1. Sequence

Let Σ be a set of distinct characters $\{e_1, e_2, \ldots, e_{|\Sigma|}\}$. Σ is called an alphabet and $|\Sigma|$ is its size. A sequence S over Σ is an ordered list of characters, denoted as $s_1 s_2 \ldots s_n$, where each $s_i \in \Sigma$. n is the length of S. $S[i,j]$ is a substring of S where i and j are the first and last indices of the substring in the sequence.

The DNA alphabet contains four nucleotides $\{A, C, G, T\}$. In general, the input sequence data might come as multiple sequences $S_1, S_2, \ldots, S_N$ with lengths $n_1, n_2, \ldots, n_N$ respectively. Let L be the input size (the total length) of the input sequences. An example of input sequences is shown in Table 1. The input contains 3 DNA sequences of length 16, 13 and 15 respectively. $S_1[2,5] = \mathrm{TTCG}$ is a substring in sequence S_1. The total length L is 44.

Table 1. An example input of DNA sequences

S_1 : GTTCGCATGTATACGA
S_2 : GATGTTAATGAGC
S_3 : TTCCAGCGTATCACT

Motifs in genomic sequences can be mainly categorized as three types: simple, rigid gapped and co-binding. A simple or single motif [10] [47] is a short sequence of nucleotides targeted by a biomolecule with a single DNA binding domain. A rigid gapped motif or simply gapped motif [40] [58] consists of simple motifs that are separated by a fixed distance. A co-binding motif or structured motif [2] [65] [70] consists of simple motifs that are separated by flexible gaps of variable distance. In earlier years of motif discovery, the promoter sequences of co-regulated genes are the main sources of input data [8] [22]. Most of the earlier years' discovery methods aim at discovering simple and rigid gapped motifs that are over-represented in these promoter sequences. A survey [9] conducted a large scale comparison of 13 of these motif discovery methods and found that they all perform poorly on metazoan datasets. These datasets consists of a few sequences, each of which is 2000 bp long. The poor performance on these datasets is likely due to the two factors: (1) the target motifs are eclipsed by a large amount of statistical noise as well as other functional motifs in the datasets and (2) a large number of false positives [12] [23] are predicted for the target motifs. The first factor leads to the development of the experimental techniques such as ChIP-seq to obtain more specific regions that potentially are bound by target biomolecules. Many more motif discovery methods [61] [62] [64] were developed specifically for these ChIP-seq datasets where the binding motifs are easier to distinguish from the background sequences. The second factor leads to the development of more methods focusing on co-binding motifs or composite motifs [18] [19] [89] in recognition of the combinatorial regulation of biomolecules to provide more binding specificity. A composite motif is a cluster of simple motifs in a fixed window size. Though composite motifs aim to reduce false positives in predicting binding instances, they do not reveal the combinatorial regulatory codes for cooperating co-binding biomolecules whereas co-binding motifs provide direct evidence for biomolecules co-binding within a certain distance.

There are other indirect ways for inferring co-binding biomolecules. One approach is to design large scale experiments for directly measuring TF-TF interactions [16]. The other approach is to use known simple motifs from existing databases and infer motif pairs that can explain microarray gene expression data or that occur frequently together in the dataset. However, this approach needs known motifs to be provided along with other external data such as gene expression data [79] [84] [85] [86].

Depending on how to model a motif [25] [26] [27], motif discovery methods can be divided into two main categories: the probabilistic approach and the deterministic approach. The probabilistic approach models a motif using a probabilistic model with parameters. It often uses Expectation Maximization (EM) or Gibbs Sampling (GS) method [31] [32], attempting to obtain model parameters to maximize the likelihood of the probabilistic model. This optimization process often involves iterating two steps alternatively. One step is to find the best instances in the sequences for the current model. The other step is to estimate the model parameters based on current instances. It often outputs a single best model but would miss some other significant motifs. It is also easily trapped into a local optimal solution when the target motif does not have a rich concentration in the input sequences.

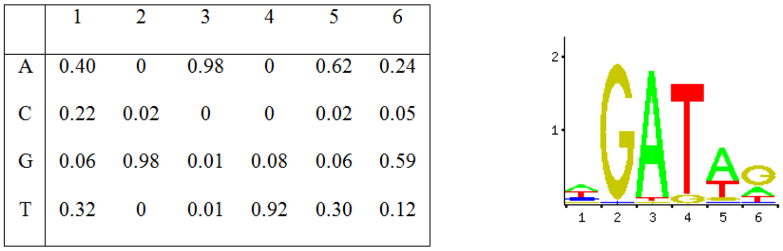
The deterministic approach models a motif by a sequence pattern. This approach then enumerates all sequence patterns satisfying certain constraints such as length and frequency requirements. The more flexible a pattern and the more relaxed the constraints, the greater the capability of this approach to capture subtle and complex motifs. However, such greater pattern flexibility and constraint relaxation could lead to an enormous pattern space and an overwhelming number of output patterns. Hence, this approach often encounters the challenges of computational efficiency as well as the need of an effective measure for ranking the output patterns.

In the following brief review, representations and methods for discovering simple motifs and rigid gapped motifs are first presented. The probabilistic approach for co-binding motif discovery is then introduced. The deterministic approach is discussed following the formal definition of flexible gap pattern discovery. The advantages and disadvantages of different methods are then discussed along the way and the state-of-the-art methods are identified for experimental comparison. The review ends with a summary of the differences between the method DFGP proposed in this thesis and the existing deterministic methods.

2.2 Discovery of Simple Motif and Rigid Gapped Motif

2.2.1 The Probabilistic Approach

In the probabilistic approach, simple motifs and rigid gapped motifs are often modeled using a position weight matrix (PWM) [12] [26]. A PWM $\mathbf{w}$ is a matrix representing a motif of l sites. Each entry $\mathbf{w}(c, j)$ of the matrix is the probability of observing the character c at the site j. Figure 2(a) shows an example PWM with 6 sites.



	1	2	3	4	5	6
A	0.40	0	0.98	0	0.62	0.24
C	0.22	0.02	0	0	0.02	0.05
G	0.06	0.98	0.01	0.08	0.06	0.59
T	0.32	0	0.01	0.92	0.30	0.12

(a)

(b)

Figure 2. PWM $\mathbf{w}$ and its sequence logo. (a) The PWM $\mathbf{w}$ for the binding motif of transcription factor GATA3 from JASPAR database [28] (b) The sequence logo of the PWM for visualization. The y-axis represents the relative entropy.

Given $f(c)$, the probability of observing character c in the background, the relative entropy R of a given PWM is

$$R = \sum_{j=0}^{l} R_j = \sum_{j=0}^{l} \sum_{c \in \Sigma} \mathbf{w}(c, j) \log_2 \frac{\mathbf{w}(c, j)}{f(c)}$$

where R_j is the relative entropy for the site j. Relative entropy is also called information gain. It provides a measure to indicate how a given motif deviates from the probability distribution of characters in the background. In a particular site, the more the distribution of the random variable deviates from the background distribution, the higher is the relative entropy for that site. Many probabilistic methods attempt to find a PWM with the highest relative entropy in the discovery process. It was shown that maximizing the information gain is equivalent to maximizing the log likelihood ratio that the observed data is generated by the PWM rather than by the background model [29] [90].

The sequence logo shown in Figure 2(b) is a visualization of a PWM. Each column of the sequence logo corresponds to a site. The height of each site is the relative entropy R_j when assuming an equal probable distribution of nucleotides in the background (i.e., 2 bits minus the Shannon entropy for the site j). The height of a character at a site is displayed in proportion to the observed probability of that character. The sequence logo visually reveals how conserved it is in a motif site. For example, the sites 2-4 of the binding motif for TF GATA3 are well conserved.

Using PWM as a motif representation, probabilistic methods often use either Expectation Maximization (EM) or Gibbs Sampling techniques, attempting to find model parameters that maximize the expected likelihood of the data given the model. Gibbs sampling can be viewed as a stochastic implementation of Expectation Maximization [30]. Here the basic EM and Gibbs sampling methods are shown below. They assume that each input sequence contains exactly one model instance.

The first EM method for motif discovery was proposed by [31]. It consists of two steps, E step and M step in the iteration. EM algorithm iterates between calculating the probability for each l-mer (subsequence of length l) generated by the current PWM and computing a new PWM based on the probabilities.

The basic EM method

Initialization step: Set the initial PWM randomly
Iteration step
E step: For each l-mer in the input sequences, the probability that it is generated by the current PWM, is computed. The sum of the probabilities of all l-mer in each sequence is normalized to one.
M step: The expected count of an observed character in a certain position of PWM is calculated as the sum of the probabilities of each l-mer whose corresponding position contains this character. The probability of observing a certain character in a given position (an entry) of PWM is then updated by normalizing the expected count of that character with respect to the total expected count (the sum of the probabilities of all l-mers).
Repeat iteration step, until the stop condition is met. (i.e. no further improvement)

Unlike EM algorithm, Gibbs Sampling [32] initially selects a l-mer from each sequence. PWM is directly computed from these aligned l-mers. Gibbs Sampling iterates between updating the selection of l-mers by the current PWM and computing a new PWM based upon the selected l-mers.

The Basic Gibbs Sampling Method

Initialization step:

The initial set of occurrences is formed by randomly selecting one l-mer from each sequence. Denote the occurrence in sequence i by o_i.

Iteration step:

(1) Pick randomly one sequence i;

(2) Compute PWM based on the set of occurrences except o_i;

(3) For each l-mer in sequence i, compute the probability that it is generated by the PWM, rather than by the background distribution;

(4) Choose randomly the new occurrence o_i' among all l-mers of sequence i according to their corresponding scores (it is more probable to be chosen with higher score);

(5) Replace o_i with o_i' in the set of occurrences.

Repeat iteration step, until the stop condition is met.

EM algorithm takes the weighted average across all l-mers whereas Gibbs Sampling takes a weighted sample from all l-mers. Given sufficient iterations, Gibbs sampling would efficiently sample the joint probability distribution of the likelihood of PWM. As EM, Gibbs sampling would also converge to a local maximum.

MEME [10] is an improved version of the basic EM algorithm [31]: (1) it removes the assumption that the probabilistic model has exactly one instance in each input sequence; (2) it can be forced to report several best PWMs instead of only one and (3) it increases the chance of finding the globally optimal PWM. Improbizer [33] adapts MEME for identifying cis-regulatory elements that activate expression within the pharyngeal gene clusters in C. elegans.

The research group of C. E. Lawrence continued to develop motif sampler [34], recursive sampler [35] and centroid sampler [36] that improves the first Gibbs sampling method [32] developed by them along this line. Motif sampler eliminates the one motif instance per sequence assumption, allowing 0 to maximum possible motif instances in one sequence. Recursive sampler implements an advanced sampling method that enables it to obtain simultaneously multiple PWMs with widths specified in advance, though more computationally intense than motif sampler. Centroid Sampler is a modification version of recursive sampler that does not pursue the optimal solution. It obtains a centroid solution consisting of a set of instances from the total sampled instances acquired during sampling period such that these instances achieve minimal cost in explaining other instances.

There are also many variants of Gibbs sampling based methods: ANN-spec [37], GLAM [38], AGLAM [39] and SeSiMCMC [40]. ANN-Spec combines an Artificial Neural Network with a PWM and uses a Gibbs sampling method to search for model parameters that maximizes the likelihood that a motif instance appears at least once in each input sequence compared to a background sequence set. GLAM modified the original Gibbs sampling method by including a procedure to automatically adjust the PWM width and modifying the way to choose a model instance in a sequence to escape from being trapped into a local maximum. AGLAM improves GLAM by incorporating positional information of model instance in assessing model quality. SeSiMCMC modified the Gibbs sampling method to better capture rigid gapped motifs of direct/inverted spaced repeat and to automatically determine the motif length for the PWM.

2.2.2 The Deterministic Approach

In the deterministic approach, sequence patterns are used to capture simple and rigid gapped motifs. Before discussing various methods for discovering sequence patterns, some basic sequence patterns, which will be used later for developing the proposed method DFGP, are introduced. The simplest sequence pattern is the solid pattern defined below.

Definition 2. Solid Pattern

A *solid pattern* P is a short sequence $p_1p_2 \ldots p_m$ over Σ where m is the pattern length. m should be at least 2.

The solid pattern is essentially a substring in the input sequences. It is rigid for modeling simple motifs since it does not allow any variation in a motif site. Sequence segments in the input must be exactly matched to the solid pattern to be occurrences of the pattern. Formally, the occurrences of a solid pattern and its counting statistics are defined as follows:

Definition 3. Solid Pattern Occurrence

A solid pattern P of length m occurs at position j in the sequence S_i if the pattern P matches a substring $S_i[j, k]$ in S_i where $k = j + m - 1$. Let $\rho = S_i[j, k]$ denote an *occurrence* of P. Then the list of all occurrences of P is represented as $L_P = \{\rho_1, \rho_2, \ldots, \rho_{|L_P|}\}$.

In compliance to Definition 3, let $l(\rho) = m$, $sid(\rho) = i$, $s(\rho) = j$ and $e(\rho) = j + m - 1$ be the length, the sequence ID, the starting position and the end position of an occurrence ρ for convenience. These notations will be used in the later definitions.

Definition 4. Number of occurrences of P

The *number of occurrences* of P denoted by k_P is the size of the occurrence list L_P.

Definition 5. Quorum of P

The *quorum* of P denoted by q_P is the number of sequences in which P occurs.

Though here the number of occurrences and quorum are defined for a solid pattern, they can be similarly extended to other sequence patterns since a sequence pattern alone determines its occurrences in the input sequences. They are often called frequency in the literature. Since a pattern with low frequency is unlikely to be a motif, deterministic methods aim at discovering frequent patterns only. A pattern is called *frequent* if its frequency is above a user defined threshold th_f.

Though solid pattern is inappropriate in directly modeling simple motifs since it does not allow motif site variations, it can be an integral part of other more complicated methods and serves as a basic unit for more complex sequence patterns. In fact, consensus pattern as defined below is a solid pattern with mismatches allowed.

Definition 6. Consensus Pattern

A *consensus pattern* P is a solid pattern but allows a certain number of mismatches up to a constant β in defining its occurrences in the input sequences. In other words, P occurs at position j in S_i if $H(P, S_i[j, j + m - 1]) \leq \beta$ where H is the Hamming distance measure.

Consensus pattern adds some flexibility to solid pattern. It is more suitable for modeling motifs by allowing motif site variations. Consensus patterns serve as boxes in a flexible gap pattern, the sequence pattern that is central to the research in this thesis. Chapter 3 focuses on how to select a representative set of consensus patterns as boxes for assembling flexible gap patterns instead of all available ones.

Example 1. Table 2 shows an example of a pattern $P =$ GTAT for the sequence dataset in Table 1. As a solid pattern, it has two exact matches in the input sequences and its quorum is 2. As a consensus pattern, it has an additional approximate match GCAT starting from position 5 in sequence 1. The quorum for this consensus pattern is 2 as well.

Table 2. An example solid pattern and consensus pattern for sequences shown in Table 1

Pattern	L_P
GTAT (solid)	$\{S_1[9,12], S_3[8,11]\}$
GTAT (consensus)	$\{S_1[5,8], S_1[9,12], S_3[8,11]\}$

The computational efficiency and the evaluation of pattern quality for capturing motifs are the two major themes in the pattern discovery approach. Consider the input sequences of the total length L, the theoretical size of solid patterns is $O(L^2)$. Hence, straightforward enumeration methods for generating solid patterns and consensus patterns can take $O(L^3)$ and $O(L^4)$ time respectively. These methods are impractical for large sequence datasets. More efficient methods were therefore developed for discovering solid patterns and consensus patterns.

Instead of considering all solid patterns, Verbumculus [41] discovers unusual solid patterns of size linear to the input size by considering only strings represented by the internal nodes of a suffix tree. The concepts of maximal solid patterns [42] [43] are proposed to extract a minimal size of solid patterns in the input sequences without information loss. A maximal pattern is the one that cannot be further extended at both ends by adding additional characters without reducing its number of occurrences. Non-maximal patterns are redundant since their positional information is completely captured by their corresponding maximal patterns. The number of maximal patterns is linear to the input size.

To discover consensus patterns in input sequences, WINNOWER [44] requires the input of the length of the pattern to be identified and the number of mismatches allowed. It finds from each sequence a pattern occurrence such that the Hamming distance between occurrences is at most 2 times the number of mismatches by transforming this problem to finding a clique of a certain size in a graph. The paper proposed by Sagot [45] takes a very different approach from WINNOWER to solve the problem. It builds a generalized suffix tree [46] for the input sequences. The method is based on the fact that the occurrence of a consensus pattern corresponds to a path spelled from root in the generalized suffix tree. It is similar to the exhaustive enumeration but with a clever pruning strategy for the suffix tree. Weeder [47] differs from Sagot's method [45] by redefining the valid occurrences of a consensus pattern. The valid occurrence of a pattern not only needs to be within a certain number of mismatches from the consensus but is more strictly defined as follows: Given an error ratio, any valid prefix of an occurrence cannot have more than a number of mismatches specified as the prefix length multiplying that error ratio. The unknown motif length problem makes consensus pattern finding methods infeasible. In practice, Weeder is confined to explore only patterns of length 6, 8 and 10 with respectively 1, 2 and 3 mismatches allowed to further reduce the huge computational cost.

The size of the set of consensus patterns can be as large as $O(L^2)$ or $O(|\Sigma|^l)$ where l is the maximum pattern length to be considered. The typical length of a simple motif can be up to 12. However, even considering only patterns of length 8, the number of consensus patterns is bounded by 4^8. Hence, the use of the set of consensus patterns as the assembling elements for flexible gap pattern generation is not feasible. In Chapter 3, a set of maximal patterns, which is a subset of consensus patterns, is used instead as a starting point to extract representative patterns. The rationale is that the well conserved part of a simple motif would be represented by a maximal pattern under low frequency threshold. This thesis develops a linear time algorithm based on the suffix tree to extract maximal patterns. However, the number of maximal patterns is still large. Hence, this thesis proposes the concept of representative patterns to further reduce the number of patterns to be used as boxes.

Solid patterns and consensus patterns are only suitable for modeling simple motifs. As for rigid gapped motifs, deterministic methods attempt to model them by using rigid patterns. A rigid pattern is a solid pattern defined over an extended alphabet Σ' that includes the original alphabet and degenerate characters from IUPAC nucleotide code. A degenerate character can match some nucleotides in the original alphabet, including a special character ‘N’ called a wild card character that can match any nucleotide. Each occurrence of a rigid pattern has the same length. Hence, rigid pattern is only appropriate for modeling rigid gapped motifs.

However, rigid pattern discovery encounters an exponential search space $O(2^L)$ due to the introduction of the wild card [43]. Thus, rigid pattern discovery methods often constrain the number of wild cards and their positions in the pattern and the quorum as well as employ pattern redundancy reduction to alleviate the huge computational burden. TEIRESIAS [48] defines a window length and requires at least a sufficient number of solid characters (not wild card character) in the window. MADMX [49] requires the ratio of the number of wild cards to the pattern length smaller than a threshold. SPLASH [50] requires that any substring of a certain length in a rigid pattern, starting with a solid character, has a sufficient number of solid characters. The wild card density constraint helps to narrow down the search space. These rigid pattern discovery methods also focus on discovering only maximal rigid patterns. A maximal rigid pattern cannot be further extended by adding any solid character to it or made more specific by replacing a wild card to a solid character without decreasing its frequency. The concept of irredundant patterns (a specific type of pattern redundancy) is proposed to further exclude exploring more patterns during the pattern generation process [51]. An irredundant pattern is a maximal rigid pattern which cannot be covered by its maximal superpatterns. A set of irredundant patterns thus form a motif base where all maximal patterns can be generated. Different motif bases are compared and discussed in [52] [53]. However, the size of the motif base is exponential in the minimum required quorum. In an extreme case, when the minimum required quorum is set to 2, the paper [54] shows that the size of the motif base is linear in the input size and can be extracted in linear time. However, a small quorum threshold would lead to a majority of patterns having a small number of occurrences close to the minimum quorum in the motif base. A pattern without sufficient quorum in input data is unlikely meaningful.

In fact, rigid pattern discovery methods are more suitable for discovering rigid gapped motifs from sequences of a protein family. In such case, the input dataset is relatively small and the minimum quorum can be set to a large number (i.e. half of the number of input sequences). On the contrary, DNA sequence datasets are often large and DNA motifs could have a relatively low quorum.

VARUN [55] extends rigid pattern to extensible pattern by incorporating flexible gaps into rigid patterns, and discovers maximal extensible patterns. A flexible gap in a maximal extensible pattern cannot be replaced by a fixed number of wild cards without decreasing its occurrences. However, allowing flexible gaps in the pattern further increases the computational cost that is already very demanding. Like those rigid pattern discovery methods, Varun is also specifically designed for motif discovery in protein sequences. In fact, Varun is confined to rigid mode when it was applied to DNA sequences in its original paper. The Varun program crashed for the smallest dataset in the experiment even with a small maximum distance for the flexible gap (i.e. 5). Similarly, subtle-varun [56], a follow-up method to detect subtle motifs in DNA sequences using Varun as a precursor, allows only a small maximum distance. Hence, subtle-varun is more suitable for discovering degenerated simple motifs due to small local insertions and deletions in their motif sites.

There are also some non-classical rigid pattern methods that tailor to a specific type of DNA motifs. For example, RSAT [57] defines a specific rigid pattern composed of two solid patterns of length 3 with a fixed number of wild cards in between them. RSAT was restricted to only discover dyads, a specific rigid gapped motif in Yeast. Similarly to RSAT, YMF [58] designs a specific rigid pattern that can contain 4 nucleotides as well as ambiguous characters {R,Y,W,S} from IUPAC codes but restrict the wild cards in the middle. The constraint imposed by YMF make it better to identify rigid gapped motifs composed of two component motifs, however, the computational cost can be high for large gap range. The maximum pattern length allowed is 23 and the maximum number of wild cards allowed is 13 in the YMF program. DREME [59] is specifically designed for finding the conserved regions of simple motifs in ChIP-seq datasets. It focuses on the rigid patterns defined over an IUPAC alphabet consisting of a regular DNA alphabet and 11 ambiguous characters. It limits the length of the pattern in the range of 4 to 8 bp and allows only limited number of ambiguous characters in the pattern. It first obtains solid patterns without ambiguous characters and calculates a p-value by Fisher's exact test, which indicates pattern enrichment in the input data relative to the control set. The control set of sequences can be constructed by randomly shuffling the set of input sequences if it is not given. It then heuristically tries to replace some characters in a solid pattern by ambiguous characters to form a new rigid pattern satisfying the condition that all solid patterns compatible with this rigid pattern are all significant. It outputs the best rigid pattern and erases all its occurrences in the input sequences by replacing them with a special symbol not in the alphabet. The whole process is repeated to obtain the next best pattern until the E-value of a pattern is below a particular threshold.

2.2.3 The Combined Approach

Some methods combine both probabilistic and deterministic approaches. For example, CisFinder [60] starts with solid patterns of length 8 based upon the empirical observation of TFBS length. It then creates 8 rigid patterns with specific gap configurations for each solid pattern by inserting wild cards. A position frequency matrix (PFM) associating to a rigid pattern is obtained as follows: The frequency of a nucleotide at a site in this PFM is obtained as the frequency of its associated rigid pattern with the corresponding site replaced by that nucleotide. A PFM for the control sequence set is obtained similarly. An intermediate PFM is obtained by the subtraction of these two matrices and its negative entries are replaced by 0. Finally, the PWM for a rigid pattern is obtained by normalizing the intermediate PFM. It groups similar PFMs into clusters to reduce redundancy. CisFinder can identify simple motifs as quickly as DREME but also has the ability to find rigid gapped motifs of specific gap configurations specified beforehand. MEME-Chip [61] combines the results from MEME and DREME. MEME in practice can identify only several simple or rigid gapped motifs. DREME complements MEME by identifying the well conserved regions of simple motifs quickly. MDscan [62] is specifically designed for ChIP-seq datasets which contain peaks ranked by their signal quality.

It utilizes the fact that top ranked sequences often contain strong TF binding motifs. It first enumerates solid patterns of a user defined length as seeds in the top 5 input sequences. All solid patterns with Hamming distance less than a threshold to the seed are used to form an initial PWM for that seed. A score is designed for evaluating the seeds. The 10-50 seeds with highest scores are retained and refined. MDmodule [63] further improves MDscan by reducing redundant PWMs. It is very efficient and hence allows the user to try different motif lengths.

2.2.4 Summary

Both the PWM model and sequence pattern models confine to a fixed motif length. No flexible gaps of large maximum distance are allowed. Hence, these methods are only able to discover simple motifs and rigid gapped motifs. They are not capable of finding co-binding motifs where there exists a flexible gap between adjacent component motifs.

Among the probabilistic methods, despite the fact that many methods are based on Gibbs sampling, MEME, an improved version of the EM method, remains one of the best in motif discovery. In practice, MEME, which scans motif length from 8 to 50 and is able to report several potential distinct PWMs, is effective and reasonably efficient to uncover simple and rigid gapped motifs in a large dataset, and is widely used by biologists in identifying motifs in ChIP-seq datasets.

For the deterministic methods, though there has been considerable effort in developing methods for finding rigid gapped motifs, it still remains a difficult task. Most of the newly developed ones are only applicable to certain restricted settings such as YMF. Comparatively, some probabilistic methods like MEME perform better. Nonetheless, some deterministic methods, such as Weeder, which discover consensus patterns, have been reported to achieve comparable performance in identifying simple motifs as the probabilistic methods.

The combined approach, which takes advantages of both the probabilistic and deterministic approaches, generally improves motif discovery performance. MEME-Chip and MDmodule are two such state-of-the-art methods in this approach. These two methods are specially tailored to handle ChIP-seq datasets. CisFinder has similar speed to MDmodule but is more or less confined to motifs with specific gap configurations.

Though these simple motif and rigid gapped motif discovery methods cannot directly discover co-binding motifs, they may provide discovered motifs as component motifs for building co-binding motifs. However, the issue is that most methods aim at finding one or several best motifs in a dataset, thus many other less dominant motifs would be missed while they can be parts of a strong co-binding motifs. In certain cases, some deterministic methods can output a large set of consensus patterns as boxes for building flexible gap patterns, but the sheer size of flexible gap patterns resulting from box combinations makes it infeasible to use all of them. Chapter 3 of this thesis is dedicated to the task of extracting a more compact set of consensus patterns for discovery of flexible gap patterns.

In this thesis, the state-of-the-art methods, MEME, Weeder, MEME-Chip and MDmoudle, are included for comparison in experiments later in Chapter 6. A final note is that since the occurrences of a pattern outputted by all these deterministic methods has the same length, these occurrences can be aligned together to form a PWM for that pattern as PWM is widely accepted as the motif representation for many well-known databases. In the experiments in Chapter 6, all discovered sequence patterns reported by a deterministic method including the one developed in this thesis DFGP would be converted to their PWM representations for comparison purpose.

2.3 Discovery of Co-binding Motif

Similar to simple motif and rigid gapped motif discovery, methods in co-binding motif discovery can be grouped into the probabilistic approach and the deterministic approach. The probabilistic approach models a co-binding motif by putting together two PWMs and with the constraint of a flexible gap in between. The PWMs are used to model simple motifs. The gap constraint is often enforced by a two stage Gibbs sampling process. It samples a sequence segment matching the first PWM and then samples another sequence segment matching the second PWM within the range of the flexible gap. The deterministic approach models a co-binding motif by a flexible gap pattern composed of a number of boxes. Each box is a consensus pattern for modeling a simple motif. It then devises algorithms and data structures in an attempt to enumerate all frequent flexible gap patterns more efficiently. There is also another approach that transforms the input sequences into a new sequence dataset. In the new dataset, each sequence corresponds to a possible sequence segment compatible to a co-binding motif template. The length of each sequence is the sum of the lengths of component motifs. It then utilizes the transformed dataset to find co-binding motifs. This approach is referred to as sample based approach in this thesis. In this section, the probabilistic approach is first presented. The deterministic approach is discussed following an introduction to flexible gap pattern discovery. The sample based approach is then presented. Finally, a summary of the uniqueness of the proposed method DFGP in comparison to existing deterministic methods is provided.

2.3.1 The Probabilistic Approach

Bioprospector [64], Co-Bind [65] and BiPad [66] are methods that model co-binding motifs through two PWMs with user-defined motif lengths. To incorporate the flexible gap constraint between two PWMs, they constrain the instances for the two PWMs to be within the flexible gap constraint whose range is specified by the user during the search process. Bioprospector seeks to maximize a score that combines the information content of the two PWMs adjusted by the number of model instances found. Co-Bind attempts to find PWMs that maximize the probability that both instances of the two PWMs simultaneously occupy in all input sequences. BiPad optimizes the objective function capturing both the information content of PWMs and the gap penalty by using a greedy search strategy. Co-Bind assumes one co-binding instance per sequence. BiPad can handle the case of one or zero instance per sequence. Bioprospector can handle the case of many instances or zero instance per sequence.

The foremost issue of this approach is that these methods are easily trapped into the local optimum solution. The main reason is that the instances of a co-binding motif may not appear very frequently in the input sequences unlike the case of finding simple and rigid gapped motifs where the primary motifs often have high enrichment in the dataset. This creates a huge difficulty for the two step sampling process. It was observed in the experimental results in this thesis that the first PWM obtained by Bioprospector often captures a primary motif in a ChIP-seq data yet the second PWM has very low information content, indicating that the second one is very poor in capturing motifs. In addition, these methods require specific knowledge of both the component motif lengths as well as the range of the flexible gap, further impairing its effectiveness in finding co-binding motifs. In practice, they may need to try a large amount of combinations of parameter values to render reasonable yet unguaranteed results. Finally as these methods seek to output only one best solution, they may miss many equally important co-binding motifs. Among these methods, only Bioprospector, which is the best of this kind, is included for comparison. Co-Bind is significantly slower than Bioprospector and there is no access to the BiPad program.

A related method is GLAM2 [67]. It uses one PWM to model motif but allows flexible gaps among motif sites by incorporating arbitrarily insertions and deletions during Gibbs sampling process. It relaxes the requirement that the lengths of sequence segments sampled be the same as the motif length for the PWM. This method suffers extensive search as it requires exploring an enormous space during sampling and editing. Thus, it is easily led to a solution far from optimum for large datasets. In addition, it is unclear that the arbitrary deletion and insertion scoring scheme specifically designed for protein sequences would work for DNA sequences. As pointed out in the original paper, discovering complex gapped DNA is a more speculative application of GLAM2. Nonetheless, it is also included for comparison in the thesis.

2.3.2 Introduction to Flexible Gap Pattern Discovery

Here flexible gap pattern discovery is formally introduced. A flexible gap pattern consists of a number of boxes with a flexible gap in between two consecutive boxes.

Definition 7. Flexible Gap Pattern

A *flexible gap pattern* $G = (P^i)_{1 \leq i \leq r}$ is a tuple of r consensus patterns co-occurring in sequential order with $r-1$ flexible gaps between them where P^i is a consensus pattern. The special symbol "$-$" is used to represent a flexible gap, which constrains P^{i+1} to occur within a predefined maximum distance d after P^i . G can be explicitly expressed as $P^1 - \ldots - P^i - P^{i+1} \ldots - P^r$.

An element P^i of a flexible gap pattern is called a box. The generic template for flexible gap patterns requires the two parameters r and d where r is the number of boxes and d is the maximum allowable distance between consecutive boxes. A flexible gap pattern models a co-binding motif in the way that each box corresponds to a component simple motif. Let $G(x) = (P^i)_{1 \leq i \leq x}$ be a prefix pattern with x boxes for G where $1 \leq x \leq r-1$. When $r = 1$, G contains only one box and is considered as 1-box flexible gap pattern, which has no prefix pattern. This special case is included just for illustrative purpose. In flexible gap pattern discovery, when asked to report flexible gap patterns of r boxes, all their prefix gap patterns $G(x)$ where $2 \leq x \leq r-1$ are also included in the output. Note that 1-box patterns are excluded. The occurrence of flexible gap pattern is defined below.

Definition 8. Flexible Gap Pattern Occurrence

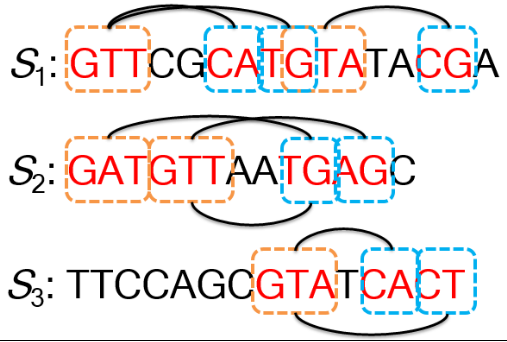
An occurrence of a flexible gap pattern G with r boxes ($r-1$ flexible gaps) is denoted by $\tau = (\rho^i)_{1 \leq i \leq r}$, which specifies the location of G in the input sequences where ρ^i represents an occurrence of box pattern P^i. The flexible gap constraint requires that ρ^{i+1} is within d distance from ρ^i. More specifically, it requires that the starting position of ρ^{i+1} is within d gaps after the end position of ρ^i, that is, $e(\rho^i) + 2 \leq s(\rho^{i+1}) \leq e(\rho^i) + 1 + d$ and $sid(\rho^i) = sid(\rho^{i+1})$ for

$i = 1, .., r-1$. τ can be explicitly expressed as $\rho^1 - \ldots - \rho^i - \rho^{i+1} \ldots - \rho^r$. All occurrences of G forms an occurrence list $L_G = \{\tau_1, \tau_2, \ldots, \tau_{|L_G|}\}$.

Let $sid(\tau) = sid(\rho^1)$, $s(\tau) = s(\rho^1)$, $e(\tau) = e(\rho^r)$ and $\alpha(\tau) = e(\tau) - s(\tau) + 1$ be the sequence ID, the starting position, the end position and the span of an occurrence τ respectively for convenience. Let $\tau(x) = (\rho^i)_{1 \leq i \leq x}$ be a prefix occurrence of τ where $1 \leq x \leq r-1$. Similarly, the number of occurrences k_G of G is the size of L_G and the quorum q_G of G is the number of sequences in which G occurs.

Example 2. Table 3 shows an example flexible gap pattern $G =$ GTT-CG of two boxes where each of which is a consensus pattern with the number of allowable mismatches being one. The maximum distance d is 5. An orange dashed box is an occurrence of the first box, the consensus pattern GTT. A blue dashed box is an occurrence of second box, the consensus pattern CG. This flexible gap pattern has 8 occurrences in the input sequences in Table 1 as shown by solid arcs. The first occurrence of G in the sequence S_1 is GTT..CA where the second box CA matching the consensus CG with one mismatch is at distance 2 from the first box GTT exactly matching the consensus GTT.

Table 3. An example flexible gap pattern G (an occurrence is shown by two Boxes linked by an arc) in the sample dataset in Table 1.

$G =$ GTT-CG
 S_1: GTTCGCATGTATACGA S_2: GATGTTAATGAGC S_3: TTCCAGCGTATCACT

Given parameters $\{r, d, th_f\}$, the task is to discover frequent flexible gap patterns from input sequences $S_1, S_2, \ldots, S_N$. Note that the generic template used for defining the flexible gap patterns is very relaxed. It does not impose constraints for the box length and the minimum distance between two consecutive boxes. In the next section, deterministic methods for discovering co-binding motifs will be discussed. The structured models they consider are exactly the same as the flexible gap pattern defined except that additional parameters are introduced to constrain the flexible gap patterns to be searched.

2.3.3 Deterministic Methods for Co-binding Motif Discovery

SMILE [68], RISO [69], RISOTTO [70] and EXMOTIF [71] discover flexible gap patterns for identifying co-binding motifs. The generic template used by SMILE requires all the boxes to have the same length and introduces the minimum distance for all flexible gaps. It utilizes the suffix tree to speed up enumerating those frequent flexible gap patterns compatible to the defined template. RISO further introduces parameters to the generic template allowing a specific length for each box and the range for each flexible gap. RISO significantly improves the speed of SMILE by developing a new data structure box-link and using the factor tree, a truncated suffix tree. It claims to have an exponential gain compared to SMILE. RISOTTO further improves the speed and is 2 times faster than RISO. EXMOTIF generates all candidate boxes first and builds flexible gap patterns using solid patterns instead of consensus patterns as boxes via positional join of the boxes. It then obtains correct quorum for those flexible gap patterns by allowing using consensus patterns. EXMOTIF could take large memory space and its program crashes when a box length is set to be greater than 6 for a dataset of moderate size. In fact, the original paper did not use box length greater than 6. To evaluate the statistical significance of the discovered flexible gap patterns, they all adopt the Monte Carlo method first used in SMILE. This method basically creates sets of shuffle sequences with the same size as the input sequences. It obtains the quorum of each pattern in every shuffled sequence dataset and calculates the quorum average and standard deviation. The z-score for each pattern, which measures how significantly the observed quorum in the input sequences deviates from the expected one, is then computed. However, it is extremely slow when the number of discovered flexible gap patterns is huge.

The problem of these deterministic methods is that they need a restricted template to narrow down the search space as well as to reduce the output size. Otherwise, they can take a considerable amount of runtime to extract patterns and unmanageable runtime to compute the pattern statistical significance. Consider the example of discovering flexible gap patterns of 2 boxes from a dataset of 500 sequences (total size 150546 bp), using a restricted template that requires each box length to be 6, the maximum distance for the flexible gap to be 30 and the minimum quorum threshold to be 10% of the number of sequences, their best method RISOTTO outputted 1,976,770 patterns in 71 seconds and took 2.72 days to compute pattern significance. Using a relaxed template that allows box length to range from 4 to 12 and the minimum quorum threshold set to 5%, RISOTTO took 1.75 hours to generate 59,730,851 patterns, not including the time for computing pattern significance. However, for motif discovery, the information about the motifs to be found is not available, and hence these methods either face an enormous space to search and huge output to handle or use a more restricted template but probably hampers their ability in identifying co-binding motifs in large datasets. RISOTTO is included for experimental comparison in Chapter 4.

2.3.4 Sample Based Approach

In the sample based approach, all possible instances compatible to a motif template are extracted from the input sequences to form a new sequence dataset. Further manipulation over the new dataset is performed to find co-binding motifs. The motif template used by MITRA [72] and MERMAID [73] specifies the length for each box and the range for the flexible gap. All possible substring combinations compatible to the template are extracted. Each substring pair is combined into a new sequence. Hence a sequence dataset containing sequences of length that is the sum of box lengths is constructed. MITRA considers only co-binding motifs of 2 components and uses its developed mismatch tree to find frequent consensus patterns (the mismatches allowed is the sum of the mismatches allowed for each box) in the newly constructed data. MERMAID uses each sequence in a new dataset to initialize a candidate PWM and carries out an iterative improvement search to optimize the quality of the PWM. L-SME [74] further modifies this approach as follows: It obtains all possible instances compatible to a motif template and further edits each instance into a set of transformed instances by carrying out mismatch, box swap and box skip operations. All these instances form a candidate set of flexible gap patterns. Instances are then associated with a unique sequence ID. L-SME outputs those candidate instances that have sufficient quorum and uses the same Monte Carlo method to compute pattern significance. These methods suffer from the same problem as those deterministic methods discussed in the previous section. Using a relaxed template that allows box length to range from 4 to 12 and the maximum distance for the flexible gap set to be 30, the size of all possible co-binding motif instances to be extracted in the dataset of size 150546 bp is approximately 365,826,780. Hence, this approach is impractical for large datasets.

2.3.5 Summary

The existing probabilistic methods are incapable of identifying co-binding motifs in realistic datasets of large size due to its nature of being easily trapped into local optimality. The deterministic methods encounter a huge pattern space to process by using a relaxed motif template in order not to miss co-binding motifs hidden in the dataset. The exhaustive enumeration of all flexible gap patterns compatible to the template brings great computational burden and results in an enormous output. The method that computes the pattern significance is unable to handle the output size.

This thesis focuses on reducing both the computational burden and the huge output in a principled way rather than restricting the motif template. It also provides the novel statistical significance measures for flexible gap patterns overcoming the hurdles of the time consuming Monte Carlo method to obtain pattern significance. The developed method DFGP, which incorporates two proposed redundancy reduction methods and the statistical significance assessment into the pattern discovery process, is effective in identifying co-binding motifs in large datasets. The first reduction aims at obtaining a compact and representative set of consensus patterns as boxes for flexible gap pattern generation. The second reduction aims at reducing redundancy among occurrences of a flexible gap pattern.

Unlike existing deterministic methods that use all frequent consensus patterns as boxes, DFGP manages to select a small subset that is non-redundant yet informative. The concept of representative patterns proposed to acquire such a subset is the focus of Chapter 3. Reducing the number of boxes to consider avoids exploring a large portion of the pattern space, thus drastically increasing the computational efficiency. The proposed reduction is novel and effective. The simple way to trim frequent consensus patterns according to their statistical significance using an arbitrary large threshold would miss those that are not highly significant if being considered alone.

A flexible gap pattern can have a large set of occurrences among which much positional information is shared due to the presence of the flexible gap. This problem is inadequately addressed in the current literature. The concept of delegate occurrences and the method for extracting these occurrences are proposed in Chapter 4 of this thesis to address this problem. It is based upon the observation that many occurrences have other nested occurrences within themselves. Such occurrences are termed encompassing occurrences in this thesis. Extracting only delegate occurrences avoid counting those encompassing occurrences. The concept of delegate occurrences differs from the concept of interleaving occurrences in DyVerb by Apostolico et al. [75] in two aspects: (1) it applies to only flexible gap pattern of two boxes, (2) it allows only solid patterns to be used as boxes and cannot handle consensus patterns. In addition, DyVerb can only deal with a small dataset and small maximum distance.

Chapter 3
Discovery of Representative Patterns

3.1 An Overview

As shown in the previous chapter, the use of all frequent consensus patterns for building flexible gap patterns under a relaxed template is one of the factors that lead to the great computational burden for existing deterministic co-binding motif discovery methods. Without restricting the length of consensus patterns, there could be a large number of them. The research question in this chapter is how a much smaller yet still informative set of consensus patterns can be extracted from the sequence data to reduce the combination of boxes to be considered in the generation of flexible gap patterns.

To obtain a compact set of consensus patterns, an obvious step is to start with a set of maximal solid patterns instead of all frequent consensus patterns. The set of maximal solid patterns captures frequent substrings in the input sequences and faithfully represents potential simple motifs with low frequency threshold. Furthermore, the set of consensus patterns considered in those deterministic methods may contain some patterns that are not even a substring of the input sequences as observed in L-SME [74]. However, the set of maximal patterns itself is still an unmanageably large set (i.e., thousands when a small frequency threshold is used) though it is relatively smaller than the original set of consensus patterns. To further reduce its size significantly, a simple strategy is to trim these patterns according to their statistical significance using an arbitrary large threshold. However, this would result in missing many of those patterns that are not highly significant by themselves alone [69]. To reduce these maximal patterns not in an ad hoc manner but in a systematic and rigorous way, the concept of representative patterns is proposed and the method DRP is developed to select such a representative subset of maximal patterns. The concept of representative patterns deals with the redundancy relations among patterns. The extracted set is much smaller yet representative. After such a set of representative patterns is obtained, they are converted to consensus patterns by allowing mismatches, and used as the boxes to construct flexible gap patterns.

In the section 3.2, a linear time algorithm based on the generalized suffix tree is developed to extract a set of maximal solid patterns in the input sequences. The section 3.3 presents the concept of representative patterns and the method to extract them from a set of maximal patterns.

3.2 Discovery of Maximal Solid Patterns

In this section, the well-known data structure of generalized suffix tree and its construction is introduced. The relationship between a maximal pattern and the internal nodes of the suffix tree is then established. The algorithm DMP [76] for extracting maximal patterns is to identify internal nodes that correspond to maximal patterns.

3.2.1 Generalized Suffix Tree

The generalized suffix tree, an efficient data structure for representing input strings, is well recognized in providing linear-time solutions to many string problems.

Given a collection of multiple strings $S_1, S_2, \ldots, S_N$ over Σ, the generalized suffix tree T representing them is a rooted directed tree with the following properties.

(1) Each internal node has at least two outgoing edges each of which is labelled with a non-empty substring in the input strings. No two edges going out of a node can have the edge-label starting with the same character.

(2) Each leaf node is labelled by a position (i, j) indicating a suffix of a string S_i starting at the position j. The concatenation of the edge-labels on the path from the root to a leaf node exactly spells out the suffix of a string S_i starting at the position j.

Most often, a termination character $\$ \notin \Sigma$ is appended to the end of each string to ensure that the suffix tree T exists for the input strings. Figure 3 gives an example of T for two input strings.

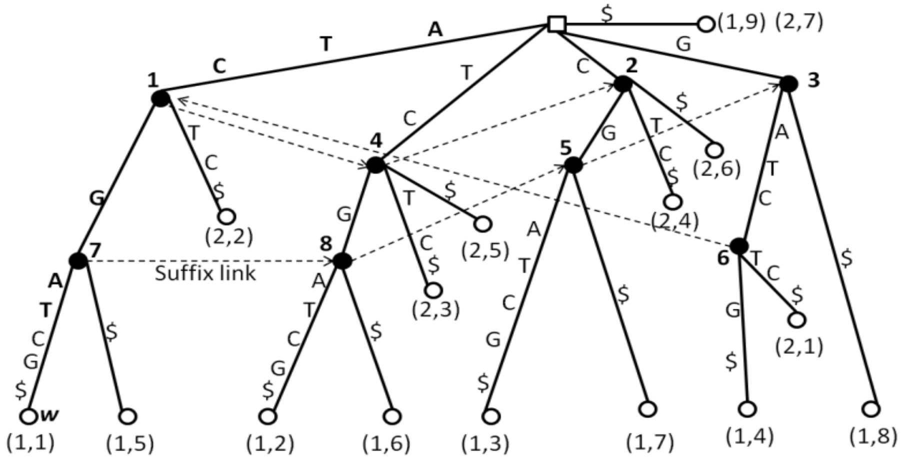


Figure 3. Generalized suffix tree T for multiple strings$\{S_1 = \mathrm{ATCGATCG\$}, S_2 = \mathrm{GATCTC\$}\}$. The square node is the root r, the solid circles are the internal nodes and the hollow circles denote the leaf nodes. The internal nodes are numbered and are denoted by v_j. w is a leaf node associated with a position $(1,1)$. An edge is denoted by the two nodes it connects and is labelled with a substring. The label of edge (v_3, v_6) is ATC. The concatenation of the edge-labels on the path from the root to the leaf node w is the suffix of a string S_1 starting at the position 1.

3.2.2 Generalized Suffix Tree Construction

We now describe a straightforward method for constructing the generalized suffix tree for the input strings $S_1, S_2, \ldots, S_N$. Let $S_{i,j}$ denote the j-th suffix of the input string i (e.g., $S_{1,1}$ is the first suffix and also the entire string of S_1). The algorithm sequentially inserts every suffixes of the input strings into the suffix tree. It begins with an initial tree $T_{1,1}$ consisting of a single edge labelled with the first suffix $S_{1,1}$. This edge connects the root and the leaf node labelled with position $(1,1)$; then it successively inserts the remaining suffixes into the growing tree. Let $T_{i,j}$ denote the tree after the suffix $S_{i,j}$ has been inserted.

The $T_{i,j+1}$ tree is constructed by inserting the suffix $S_{i,j+1}$ into $T_{i,j}$ as follows: Starting from the root of $T_{i,j}$, find the longest path label that matches the prefix of the suffix $S_{i,j+1}$ by comparing and matching characters in the prefix to characters along the path label, until no further matches are possible. At the point when no further matches are possible, the algorithm is either at a node w, or it is in the middle of an edge (u, v). If it is in the middle of an edge, it breaks the edge (u, v) into two edges (u, w) and (w, v) by inserting an internal node w just after the last matched character and just before the first mismatched character on the edge. Then (whether a new node w has been created or whether it already exists at the point of mismatch), the algorithm creates a leaf node x labelled with the position $(i, j+1)$ and an edge (w, x) labelled with the unmatched part of the suffix $S_{i,j+1}$. The tree now has the unique path from the root to the leaf node x, and this path label is the suffix $S_{i,j+1}$.

The example for constructing the generalized suffix tree T in Figure 4 using the straightforward algorithm is given below.

Position	1	2	3	4	5	6	7	8	9
S₁	A	T	C	G	A	T	C	G	$
S₂	G	A	T	C	T	C	$		

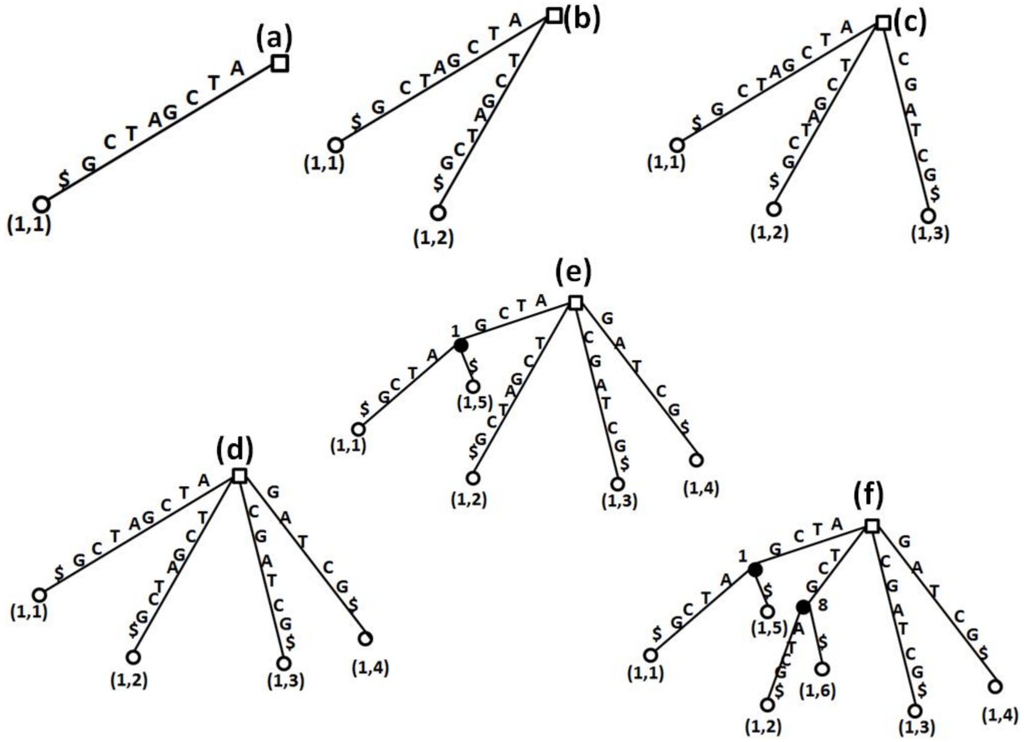


Figure 4. The first six growing trees during the construction of the generalized suffix tree construction T: (a) $T_{1,1}$; (b) $T_{1,2}$; (c) $T_{1,3}$; (d) $T_{1,4}$; (e) $T_{1,5}$; (f) $T_{1,6}$

The straightforward generalized suffix tree construction method takes $O(L^2)$ time for input strings whose total length is L and hence it is not efficient for large input. However, a generalized suffix tree T can be constructed in $O(L)$ time and space. The details of the suffix tree and its linear time and space construction algorithms can be found in [46].

3.2.3 Extraction of Maximal Solid Patterns from Suffix Tree

The generalized suffix tree is utilized in this thesis to develop the linear time algorithm DMP for discovering maximal solid patterns. A solid pattern is just a substring found in the input strings. The suffix tree in Figure 3 is used to illustrate the examples given in this section.

A solid pattern in the input string is represented by the label of a path starting from the root of T. A path label is the concatenated edge labels along the path. To find the occurrence list L_P for a pattern P, characters in P are matched against characters along a path label. At the point when all characters are matched, the matching ends at or above a node x. All starting positions of P in S can be found by listing all indices of the leaf nodes below x. The positions of a pattern can be found because they correspond to certain suffixes in the string. For example, the path label of the pattern ATC ends at the node v_1, its starting positions $\{(1,1),(1,5),(2,2)\}$ can be obtained by traversing the leaf nodes under v_1 because these leaf nodes, which correspond to the 1st and 5th suffixes of the string S_1, and the 2nd suffix of the string S_2, record the starting positions of this pattern. With the starting positions, L_P can be easily constructed with the length of P. By annotating each node x with the number of leaf nodes $k(x)$ found below x, the number of occurrences k_P can be obtained by finding the corresponding node for P. Hence, a frequent pattern P is represented by a path label ending at or above a node x with $k(x) \geq th_f$.

Recall that a maximal pattern is the one that cannot be further extended at both ends by adding additional characters without reducing its number of occurrences. Hence, a potential maximal pattern is represented by a path label ending at a node instead of above a node. A pattern with path label ending above a node cannot be maximal since it certainly has a superpattern with the same number of occurrences down the path. This superpattern is represented by the node where the path label of the pattern ends. For example, the pattern AT has the path label ending above v_1 and hence it is not maximal. It has ATC with the same number of occurrences ending at v_1.

So far it is shown that each node corresponds to a potential maximal pattern. Note that a pattern is maximal if and only if it cannot be extended at either the left or the right end. Since the case of extending the right end of a pattern has been dealt with, it remains to handle the case of extending a pattern by adding a character to its left end. If the pattern has the same number of occurrences as the extended pattern, it cannot be maximal. With this observation, a special kind of structure element called a suffix link in T is used.

Let v be an internal node in T and its path label be $pl(v) = c\alpha$ where c is a character and α is a string. Let u be another internal node with $pl(u) = \alpha$. A pointer from v to u is called a *suffix link* denoted by $SL(v) = u$. For example, in Figure 3, $pl(v_7) = \mathrm{ATCG}$ and $pl(v_8) = \mathrm{TCG}$, then $SL(v_7) = v_8$.

Since the suffix link connects a pattern $P' = \alpha$ and its superpattern $P = c\alpha$ with c appended to the left of α, the suffix link can be used to check whether or not a node corresponds to a closed pattern. A node u does not correspond to a maximal pattern if there is a node v with $SL(v) = u$ and $k(v) = k(u)$. For example, the pattern CG, though having the path label ending at v_5, is not closed since we have $SL(v_8) = v_5$ and $k(v_8) = k(v_5) = 2$. Hence, there is a one-to-one correspondence between a maximal pattern and a node in the suffix tree.

To effectively use the suffix link to identify nodes corresponding to closed patterns, the nodes with longer path labels are processed first. After maximal patterns are obtained, simple repeat patterns such as AAAAA are removed as they are unlikely to be part of simple motifs. Since there are abundant repetitive elements interspersed in the human genome, it is helpful that the input sequences are masked by repeat masking tools such as RepeatMasker [77] before discovering patterns. If otherwise, those repetitive elements are very likely to show up as strong over-represented patterns in the output, which hinders the motif discovery program's ability to find real regulatory motifs. If the repetitive elements are masked in the input DNA sequences, there would be often a character such as 'N' other than the four normal nucleotide codes in the masked sequences. Hence the patterns that contain characters not from the DNA alphabet are removed. Algorithm 3.1 DMP (Discovery of Maximal Patterns) for discovering maximal patterns from the input sequences is shown below.

Algorithm 3.1. DMP

```
Input: sequence data $\{S_1, S_2, \ldots, S_N\}$, $th_f$
Output:  a set of maximal solid patterns $\mathbb{M}$
1. Construct generalized suffix tree $T$ for $\{S_1, S_2, \ldots, S_N\}$
2. Traverse $T$ to obtain all nodes with $k(x) \geq th_f$ and initially mark them as maximal.
3. Sort the nodes in descending order according to the length of their path labels.
4. For each node $v$
        If $v$ is maximal, then output its corresponding pattern.
        Set $u = SL(v)$ to be non-maximal if $k(v) = k(u)$.
   End
5. Remove maximal patterns that have simple repeats or contain a special mask character.
```

3.3 Discovery of Representative Patterns

The concept of obtaining representative patterns is to extract a compact subset of maximal patterns such that the patterns excluded have their representatives within the selected subset. In order to explain the concept of a pattern being represented by the other one, a similarity measure between two solid patterns is provided in definition 9. Since the pattern considered is essentially a string, the length of the longest common subsequence (LCS) [78] between two strings is used to define pattern similarity. Given two strings S_1 and S_2, their LCS is the string resulting from the deletion of certain characters from both strings such that the edited strings completely match each other and have the longest length. For example, the LCS for two strings ATC and AGC is AC and its length is 2.

Definition 9. Pattern Similarity

Given two patterns P_i and P_j, their similarity measure is defined as follows:

$$sim(P_i, P_j) = \frac{2 \cdot |LCS(P_i, P_j)|}{|P_i| + |P_j|}$$

When P_i and P_j are identical, $sim(P_i, P_j) = 1$; when they are completely different (i.e. do not have common subsequence), $sim(P_i, P_j) = 0$.

A pattern can be represented by another pattern if they are similar and all patterns similar to it form a pattern neighborhood as defined below.

Definition 10. Neighborhood of Pattern P

A pattern neighborhood $NH(P)$ for P consists of a set of patterns that are similar to P. A pattern is similar to another pattern if their similarity is above a threshold th_s. Formally, $NH(P) = \{P' \in \mathbb{M} | P' \neq P, sim(P', P) \geq th_s\}$ where $\mathbb{M}$ is the set of maximal patterns.

The neighbor patterns for a pattern are in composition close to that pattern in the sense that the pattern can undergo a limited number of edits to restore each of them. If this pattern is included in the selected set of representative patterns, excluding its neighbor patterns is probably not too risky since they are represented by the selected pattern. Suppose an excluded pattern captures a simple motif, its representative pattern is likely to capture this motif as well.

Hence, the neighbor patterns of a pattern are considered as redundant with respect to this pattern. As a result of removing redundant patterns, the retained patterns become more distinct and the pattern set size is greatly reduced. Here the method DRP is developed to obtain a compact set of representative patterns such that they represent all maximal patterns. The set of representative patterns is formally defined in Definition 11.

Definition 11. A set of representative Pattern $\mathbb{R}$

A set of representative patterns $\mathbb{R}$ is a subset of $\mathbb{M}$ such that for each pattern $P \in \mathbb{R}$, $NH(P) \not\subset \mathbb{R}$ and $\mathbb{M}$ is the union of $\mathbb{R}$ with all neighborhoods of the patterns in $\mathbb{R}$.

The method DRP given in Algorithm 3.2 selects the representative patterns as follows: When a solid pattern P is chosen as a *representative pattern*, its neighbor patterns are excluded. First, it selects the one having the largest neighborhood size and excludes those patterns within its neighborhood. It then selects the next pattern with the second largest neighborhood size and so forth until all maximal patterns are examined. The reason to select representative patterns in this manner is that a representative pattern having large support from its neighbor patterns could be more important in capturing simple motifs. Suppose that a representative pattern captures a simple motif, then its neighbor patterns could be formed by a degenerate motif or a partially overlapped motif of the simple motif. Hence, the enrichment of neighborhood patterns for a pattern provides some indications that it might be important and representative with biological relevancy.

Another way to choose representative patterns is to model patterns as vertices in a graph. An edge between two vertices is formed if their corresponding patterns are similar. One might want to obtain a set of representative patterns by finding the minimal vertex cover, producing an even smaller set. However, the patterns, obtained by finding the minimal vertex cover, are not suitable for the definition of representative patterns since a pattern and some of its neighbor patterns can both be included in the output set. The proposed selection process is similar to a greedy algorithm for vertex cover problem that repeatedly chooses vertices incident to the largest number of currently uncovered edges but differs in two aspects: (1) the vertices with edges incident to the chosen vertices are not considered and (2) edges are not removed during the search process.

Algorithm 3.2. DRP

```
Input: sequence data $\{S_1, S_2, \ldots, S_N\}$, $th_f$, $th_s$
Output:  a set of representative consensus patterns $\mathbb{R}$
1. Use Algorithm 3.1 DMP to extract the set of maximal patterns $\mathbb{M}$
3. Calculate the similarity between patterns
4. Assign the neighborhood size for each pattern and sort the patterns
   according to the neighborhood size. Initially, all patterns in $\mathbb{M}$ are marked as unchecked
5. For each pattern $P$ in $\mathbb{M}$
        If $P$ is unchecked
            Add $P$ to $\mathbb{R}$
            Mark patterns in $NH(P)$ to checked
        End
   End
6. For each solid pattern in $\mathbb{R}$, convert it to a consensus pattern by allowing $\beta$ mismatches
```

In this thesis, the number of mismatches β for consensus patterns is fixed to 1. Fixing β equal to 1 is based on the observation that the core region of a simple motif is relatively better conserved. This phenomenon is also mentioned by the authors of DREME[59] and CisFinder [60]. Hence, a consensus pattern does not need too many variations when modeling this core motif region. Next, how to determine the parameter values is discussed.

The frequency threshold th_f cannot be set to a large value since this threshold is used to obtain frequent solid patterns. Otherwise, the simple motifs less conserved but important would be excluded from the set of representative patterns. Here th_f is set to $max(0.05N, 5)$, requiring that a pattern occurs a small number of times relative to the number of sequences but not less than 5 times to reduce the risk of missing important simple motifs. This means that when $N \geq 100$, a pattern is expected to appear in 5% of input sequences. When $N < 100$, a pattern needs to occur at least 5 times to be considered. A low frequency threshold is used here to exclude those patterns unlikely to correspond to motifs in the dataset but allow many slightly frequent patterns to be considered. A low frequency threshold is important for finding co-binding motifs since they might not have high enrichment in a dataset. The low enrichment of co-binding motifs is one of the critical reasons why probabilistic methods for co-binding motif discovery perform poorly in many datasets.

The similarity threshold th_s is an important parameter as it controls what is to be considered as a similar pattern. Setting it too high would result in a large number of representative patterns being retained while a low value would perceive two arbitrary patterns as similar, making the pattern neighborhood meaningless. By using LCS to measure the similarity between two patterns of length l and l' (assuming $l \leq l'$), the maximum similarity between them can be achieved is $\frac{2l}{l+l'}$. Under a specific threshold th_s, these two pattern are similar if $\frac{2l}{l+l'} \geq th_s$. By rearranging the inequality, $\frac{l'-l}{l} \leq 2(\frac{1}{th_s} - 1)$ is obtained. The threshold th_s controls the ratio of the length difference to the length of the shorter pattern. Setting th_s to 0.7, 0.8 and 0.9 results in 0.85, 0.5 and 0.2 for the value of this ratio respectively. Here $th_s = 0.8$ is used. This requires that the length difference between two similar patterns would not be overly large relative to the length of the shorter pattern (i.e. the length difference is at most half of the length of the shorter pattern).

The reduction from the set of maximal patterns to the set of representative patterns is one of the two reductions in the proposed method DFGP. This significantly reduces the computational burden in discovering flexible gap patterns. Here the reduction ratio is estimated as follows: With $th_s = 0.8$, consider a pattern P of length $l+1$ and its subpattern P' of length l, then $sim(P, P') = \frac{2l}{2l+1} \geq 0.8$ and hence P and P' are neighbors. Theoretically, for a given pattern P, there could be 10 such neighbor patterns. Two of them are its subpatterns with one character less. They are obtained by removing a character from either end. Eight of them are its superpatterns with one character more, obtained by appending a character (one out of four nucleotides) to either end. Choosing P as a representative pattern would exclude these 10 patterns. Hence, approximately, $|\mathbb{M}| \approx 11|\mathbb{R}|$ and the reduction ratio $\alpha = 0.09$, meaning the size of the representative patterns could be only 9% of the original maximal patterns, possibly leading to the exclusion of exploring 99% of the box combinations during generation of flexible gap patterns of 2 boxes.

Example 3. Figure 5 shows the set of representative patterns discovered by DRP from the input sequences in Table 1 using $th_s = 2$ (note that this is just a toy example and so the minimum required threshold of 5 occurrences does not apply here). Under $th_s = 2$, a maximal solid pattern needs to occur at least twice. Figure 5(a) shows the set of maximal patterns obtained by the method DMP. There are 15 maximal patterns in total. The pattern ATG has 3 occurrences in the input sequences. This pattern is maximal since it cannot be further extended by adding a character to either end without decreasing the number of occurrences. Figure 5(b) shows the pattern neighborhoods for the maximal patterns. Patterns AT and ATGT are the neighbor patterns of the pattern ATG since both of them are similar to ATG. The pattern TA does not have any neighbor pattern since it is not similar to any other maximal pattern. After the pattern neighborhoods are established. DRP selects the pattern with the largest pattern neighborhood and skips all its neighbor patterns. In this case, DRP selects the pattern ATG and excludes its neighbor patterns AT and ATGT. It then selects the next pattern GTT and so forth. As a result, a set of representative patterns is selected. Patterns in shaded circles are representative patterns. DRP reduces the set of 15 maximal patterns to the set of 8 representative patterns.

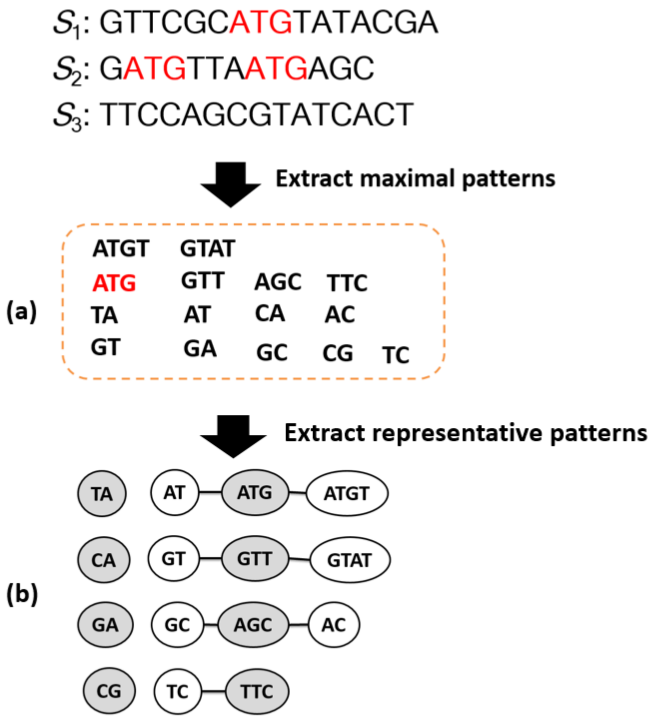


Figure 5. An example for extracting the set of representative patterns for input sequences in Table 1 with $th_f = 2$. (a) The set of maximal patterns. The occurrences of the pattern ATG colored in red are shown in the input sequences. (b) The set of representative patterns. All maximal patterns are represented by the circles and similar patterns share an edge between them. The neighbor patterns for ATG are AT and ATGT as shown by the connected edges. Some patterns such as TA do not have neighbor patterns since they are not similar to any other pattern. The shaded circles are representative patterns.

Chapter 4

Construction of a Flexible Gap Pattern and its Delegate Occurrences

4.1 An Overview

This chapter considers the construction of a flexible gap pattern G of k boxes by appending a box P to a flexible gap pattern G' of $k-1$ boxes. Section 4.2 presents the method COA (Complete Occurrences Assembling) that identifies the complete set of occurrences of G during pattern construction. Analysis of this method shows that the size of the complete set of occurrences increases greatly as the maximum distance d and the number of boxes k increases. This poses a greater computational burden for larger d and k. The concept of delegate occurrences is proposed in section 4.3 to exclude counting encompassing occurrences that have nested occurrences within themselves. The obtained set of delegate occurrences is more compact and less redundant. In contrast to the size of the complete set of occurrences, the size of the set of delegate occurrences does not depend upon either d or k. The method DOA (Delegate Occurrences Assembling), which efficiently identifies delegate occurrences during the flexible gap pattern construction, is thus developed.

4.2 Construction of a Flexible Gap Pattern with Complete Occurrences

A flexible gap pattern G of k boxes is constructed by adding a box P to G' while checking whether the occurrences of the new pattern G meets the frequency threshold th_f.

A straightforward method can pair up each occurrence from the occurrence list $L_{G'}$ and L_P to form a potential occurrence for G and check whether the newly formed occurrence is valid according to the conditions specified in definition 8 of flexible gap pattern occurrence.

However, this would take $O(|L_{G'}| \cdot |L_P|)$ time. Hence, it is quite time consuming. A better method, known as COA, is presented as follows: Consider two lists of sorted occurrences $L^i_{G'}$ and L^i_P from sequence i extracted from $L_{G'}$ and L_P respectively. This method as illustrated in Figure 6 goes through each occurrence in $L^i_{G'}$ and checks whether there exist occurrences from L^i_P that are within distance d after it. The occurrences in L^i_P before it are skipped in the next checking step. For example, in Figure 6, after the occurrence of G' marked by the arrow is checked, the occurrences of P before the arrow can be ignored when checking the next occurrence of G'.

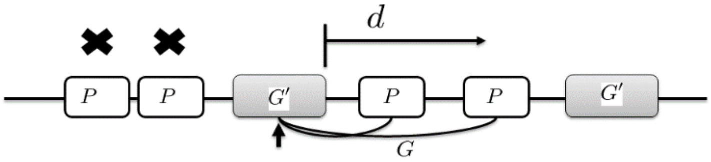


Figure 6. The Identification of occurrences of flexible gap pattern G during its construction from G' and P. A gray box represents an occurrence of G' and the white box an occurrences of P. The boxes connected with an arc form an occurrence of G. For every occurrence of G', simply checking whether there are occurrences of P within d distance.

COA identifies the complete occurrences of a flexible gap. However, it takes $O(|L_{G'}| \cdot d)$ time as an occurrence of G' can be associated with at most d occurrences of P. This also indicates that the size of the set of complete occurrences is $O(|L_G|) = O(|L_{G'}|d)$. Since the number of occurrences for a flexible gap pattern of one box (i.e., $k = 1$) can be asymptotic to the input size L, the size of the occurrence list L_G of a gap pattern of k boxes can be $O(Ld^{k-1})$. Hence $|L_{G'}| = O(Ld^{k-2})$, and the time to extract the occurrence list of G of k boxes from $L_{G'}$ and L_P is $O(Ld^{k-1})$. Thus, the computational time and space complexity increases considerably as the maximum number of gaps d and the number of boxes r increase.

4.3 Construction of a Flexible Gap Pattern with Delegate Occurrences

4.3.1 The Concept of Delegate Occurrences

It is observed that many occurrences of a flexible gap pattern have nested occurrences within themselves. Figure 7 shows the complete occurrences of a frequent flexible gap pattern $\mathrm{GTT-CG-GA}$ in the input sequences in Table 1 when $th_f = 2$ and $d = 5$ are used. There are 7 occurrences in total. Occurrence 2 has Occurrence 1 nesting within it. Occurrence 3 has Occurrences 1 and 2 nesting within it. These occurrences are referred to as encompassing occurrences. The formal definition of encompassing occurrences is given in Definition 12.

Definition 12. Encompassing occurrence

An occurrence τ of a flexible gap pattern G is referred to as an *encompassing occurrence* if there exists an occurrence τ' such that $sid(\tau') = sid(\tau)$, $s(\tau') \geq s(\tau)$, $e(\tau') \leq e(\tau)$ and the span $\alpha(\tau') < \alpha(\tau)$. Hence an encompassing occurrence strictly contains another occurrence within it.

Example 4. Occurrences 2 and 3 in Figure 7 are encompassing occurrences since each of them contains another occurrence (like Occurrence 2 contains Occurrence 1 as indicated by their spans $\alpha(\tau_2) > \alpha(\tau_1)$). Note that Occurrences 3 and 4 do not contain each other since they have the same span. However, the prefix occurrences of Occurrences 4 to 6 are encompassing occurrences since these prefix occurrences all contain the prefix occurrence of Occurrence 1.

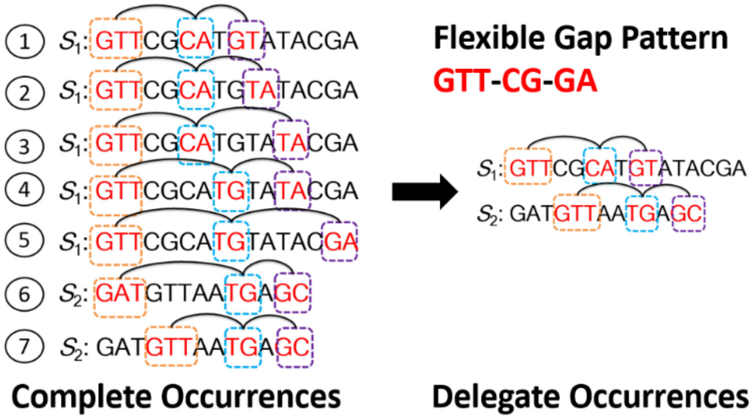


Figure 7. Complete occurrences and delegate occurrences of the flexible gap pattern $\mathrm{GTT-CG-GA}$ in sequences in Table 1 with $th_f = 2$ and $d = 5$. Each dashed box is an occurrence of the consensus patterns GTT, CG and GA with 1 mismatch allowed. A flexible gap pattern occurrence is represented by dashed boxes linked by arcs and is numbered (from 1 to 7). Occurrences 2 and 3 are encompassing occurrences since Occurrence 2 has Occurrence 1 nesting within it, and Occurrence 3 has Occurrences 1 and 2 nesting within it. Both of them are hence not delegate occurrences. Occurrences 4, 5 and 6 are not delegate occurrences as well since their prefix occurrences are encompassing occurrences. Only Occurrences 1 and 7 are delegate occurrences for the pattern since they contain no nested occurrence.

Encompassing occurrences are likely to form as a result of the flexible gap. Their number can be large with even a small value of d especially in the case when consensus patterns instead of solid patterns are used as boxes for building flexible gap patterns. Here the concept of delegate occurrences is proposed to avoid counting encompassing occurrences, which greatly improves the computational efficiency. The formal definition is given below.

Definition 13. Delegate occurrence

An occurrence τ of a flexible gap pattern G of r boxes is a *delegate occurrence* if it and all its prefix occurrences $\tau(x)$ for $2 \leq x \leq r-1$ are not encompassing occurrences. Note that when $r = 1$, every occurrence is considered a delegate occurrence. The delegate occurrence list of G is denoted as D_G. The number of occurrences and the quorum of G counting only delegate occurrence is denoted as $\hat{k}_G$ and $\hat{q}_G$ respectively.

Example 5. As shown in Figure 7, Occurrences 1 and 7 are delegate occurrences since they and their prefix occurrences are not encompassing occurrences. Occurrences 2 and 3 are not delegate occurrences since they are encompassing occurrences. Occurrences 4, 5 and 6 are not delegate occurrences as well since their prefix occurrences are encompassing occurrences.

Delegate occurrences are a subset of complete occurrences and its set size is shown to be $O(L)$ linear to the input size. It is a more compact set that captures representative information about the complete set. Counting only delegate occurrences also avoids exploring artificial frequent occurrences especially when a low frequency threshold t_f is used. It is clear that increasing d alone would make some flexible gap patterns become frequent. Experimental results presented in Chapter 6 will show that counting only the delegate occurrences instead of the complete occurrences does not impair the ability of DFGP for identifying co-binding motifs, indicating that delegate occurrences encapsulate critical information from the complete set.

4.3.2 The Identification of Delegate Occurrences

Next, the method for finding delegate occurrences of G during its construction from G' and P is presented (note that G' is a prefix pattern of $k-1$ boxes for G and P is its last box). A delegate occurrence of the synthesized G can be obtained as the pair formed by a delegate occurrence of G' and its closest occurrence of P. For example, in Figure 8, the delegate occurrence τ'_2 of G' and the occurrence ρ_2 form a delegate occurrence of G while τ'_1 cannot pair with ρ_2 to create a delegate occurrence of G since τ'_1 is not the one closest to ρ_2.

Hence, a delegate occurrence τ of G can be obtained by the closest pairing of a delegate occurrence τ' of G' with an occurrence ρ of P, denoted as $\tau = (\tau', \rho)$. The concept of closest pairing of two occurrences defined below is used to show how delegate occurrence could be obtained by synthesizing a delegate occurrence (of G') with another nearby occurrence (of box P) into the new delegate occurrence of G. Note that ρ should be within the maximum distance after τ' to be a valid occurrence τ for G. The closest pairing [91] requires that there is no other delegate occurrence τ'' after τ' and before ρ, and there is no other occurrence ρ' of box P before ρ and after τ'. Formally, this is stated in definition 14.

Definition 14. Closest pairing of τ' with ρ

The closest pairing of τ' with ρ, denoted as (τ', ρ), satisfies the following conditions: (1) the created occurrence $\tau = (\tau', \rho)$ meets the flexible gap pattern occurrence requirement, (2) there is no delegate occurrence τ'' such that $e(\tau') < e(\tau'') < s(\rho) - 1$ (i.e., in between τ' and ρ), and (3) there is no occurrence ρ' such that $e(\tau') + 1 < s(\rho') < s(\rho)$. For example, in Figure 8, (τ_1', ρ_2) is not a closest pairing since there is τ_2' in between τ_1' and ρ.

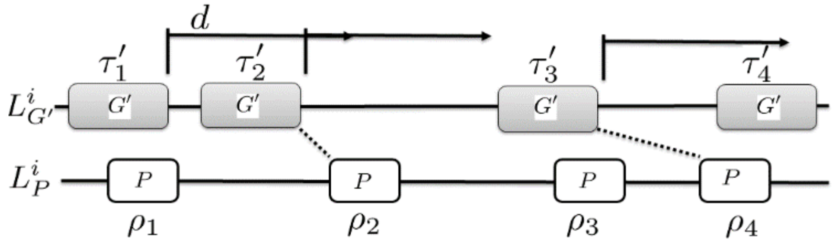


Figure 8. The identification of delegate occurrences of a flexible gap pattern G during its construction from G' and P in the same sequence S_i. The gray and white boxes along two lines represent the sorted delegate occurrences of G' and sorted occurrences of P in the sequence S_i respectively. Boxes linked by a dashed line represents a delegate occurrence of G after the synthesis process. The arrow line indicates the gap range allowed by d. In each case, like (τ_1', ρ_2) and (τ_3', ρ_4), the closest pairs are chosen.

The following proposition shows that the τ created by the closest pairing of τ' with ρ is a delegate occurrence.

Proposition 1. $\tau = (\tau', \rho)$ specified in Definition 14 is a delegate occurrence of G.

Proof. The prefix occurrence τ' is a delegate occurrence so all its prefix occurrences $\tau(x)$ for $2 \le x \le r - 1$ are not encompassing occurrences. Suppose τ is an encompassing occurrence, then it has a nested occurrence τ_α within it. Note that the end position of the prefix occurrence $\tau_\alpha(r-1)$ can either be the same as $e(\tau')$ or after $e(\tau')$. If this is not the case, τ' has a nested occurrence and hence it is not a delegate occurrence, which leads to a contradiction. Consider the case when $e(\tau_\alpha(r-1)) = e(\tau'),$ let the last box of τ_α be ρ_α, then $s(\rho_\alpha) < s(\rho)$, violating the closest pairing condition (3). In the case when $e(\tau_\alpha(r-1)) > e(\tau')$, it violates the closest pairing condition 2. Hence, τ is not an encompassing occurrence and hence it is a delegate occurrence of G.

To obtain the delegate occurrences for G during its construction from G' and P, it follows the same approach as that of the method COA but can immediately check the next delegate occurrence of G' when the current one and an occurrence of P forms a closest pair (a delegate occurrence) for G. Let $D^i_{G'}$ and L^i_P be two lists of sorted delegate occurrences from sequence i for $D_{G'}$ and L_P respectively. Two index trackers $I_{G'}$ and I_P are used to keep track of the elements in the sorted lists $L^i_{G'}$ and L^i_P respectively. To check whether an element τ' at $I_{G'}$ of $L^i_{G'}$ and an element ρ at I_P of L^i_P form a closest pairing, it suffices to check whether the starting position of ρ is in the valid range that can be derived from the end position of τ', the end position of τ'_n (the next occurrence of τ') and the maximum distance d. More specifically, the valid range is $[e(\tau') + 2, min(e(\tau') + d + 1, e(\tau'_n) + 1)]$. The valid range assures that τ' and the first ρ falling in this range forms a closest pair during the scanning of the occurrence list L^i_P. In such case, both trackers are increased by 1.

Figure 8 shows the process of identifying delegation occurrences during the pattern assemble step. The valid range for τ'_1 is $[e(\tau'_1) + 2, e(\tau'_2) + 1]$ and hence no occurrence of P falls into this range. The valid range for τ'_2 is $[e(\tau'_2) + 2, e(\tau'_2) + d + 1]$ and (τ'_2, ρ_2) is a delegate occurrence. The valid range for τ'_3 is $[e(\tau'_2) + 2, e(\tau'_3) + 1]$ and (τ'_3, ρ_4) is a delegate occurrence. This method referred to as *DOA* (delegate occurrence assembling) is presented in Algorithm 4.1. Since one delegate occurrence of G' can only pair with at most one occurrence of P, the number of delegate occurrences is $O(L)$ and DOA runs in $O(L)$ time.

Algorithm 4.1. DOA

```
Input: a flexible gap pattern $G'$, a box $P$ and $d$
Output: $G = (G', P)$ with $D_G = \{\cdots, D^i_G, \cdots\}$
1. Let $\{\cdots, D^i_{G'}, \cdots\}$ and $\{\cdots, L^i_P, \cdots\}$ be the set of sorted occurrence lists for $G'$ and $P$
2. For each pair of lists $D^i_{G'}$ and $L^i_P$ from same sequence
    Set index trackers $I_{G'}$ and $I_P$ to 1
    While $I_{G'} \leq |D^i_{G'}|$ and $I_P \leq |L^i_P|$
        Let $\tau'$ be the element at $I_{G'}$ of $D^i_{G'}$ and
        $\tau'_n$ be the element at $I_{G'} + 1$ of $D^i_{G'}$ and
        $\rho$ be the element at $I_P$ of $L^i_P$
        Obtain the valid range as
        $\mathbb{V} = [e(\tau') + 2, min(e(\tau') + d + 1, e(\tau'_n) + 1)]$
        If $\rho$ is before $\mathbb{V}$
            $I_P = I_P + 1$
        Else If $\rho$ is after $\mathbb{V}$
            $I_{G'} = I_{G'} + 1$
        Else
            Add $\tau = (\tau', \rho)$ to $D_G$
            Increase $I_{G'}$ and $I_P$ by 1
        End
    End
End
```

Chapter 5
Discovery of Flexible Gap Patterns

5.1 An Overview

In this chapter, the method DFGP (Discovery of Flexible Gap Patterns) that discovers flexible gap patterns for identifying co-binding motifs in the input sequences is presented. DFGP uses the representative consensus patterns obtained by the method DRP in Chapter 3 to build the flexible gap patterns. A flexible gap pattern is constructed from its prefix pattern and its last box using the method DOA provided in Chapter 4. The parameter settings and the runtime analysis are discussed in this chapter. The conversion of a flexible gap pattern to its PWM representation is then illustrated. Before presenting DFGP, Section 5.2 introduces the two statistical significance measures for ranking flexible gap patterns with complete occurrences and delegate occurrences respectively. Unlike the Monte Carlo method adopted by existing deterministic methods for co-binding motif discovery, these two measures are computationally efficient and do not add additional complexity into the pattern discovery process.

5.2 Statistical Significance Measures

Since the number of frequent flexible gap patterns discovered is often huge, it is exceedingly important for the discovery method to assess the importance of these patterns. Only after pattern ranking, a small number of top ranked patterns may then serve as guiding information for researchers for further study. Existing methods often use Monte Carlo simulation that generates a batch of randomly shuffled sequences of the same size as the input, and then estimates the expected quorum for a discovered pattern to compute the z-score, which measures the deviation of the observed quorum from the expected quorum. This approach is simple and straightforward. However, the sheer amount of patterns discovered makes this approach very time consuming. Here, two measures are proposed for evaluating the statistical significance of a flexible gap pattern with complete occurrences and with only delegate occurrences respectively. Both statistical measures do not add additional computational complexity into the pattern discovery process. Furthermore, these measures can provide a pruning strategy to exclude insignificant patterns to further expand into patterns with a higher number of boxes.

The measures for assessing the statistical significance of a flexible gap pattern G of r boxes are presented below. Its statistical significance is assessed in two different settings, namely, the case where complete occurrences are counted and the case where only delegate occurrences are counted. In both cases, the random model assumes that the prefix pattern $G(r-1)$ and the last box P^r occur uniformly at random in the input sequences.

In the case of complete occurrences, the probability of observing an occurrence of $G(r-1)$ in a position is estimated by its observed number of occurrences in the input sequence. Thus, $pr(G(r-1)) = \frac{k_{G(r-1)}}{L}$. The probability of observing an occurrence of P^r is estimated as $pr(P^r) = \frac{k_P}{L}$. The probability $pr(G)$ of observing an occurrence of G is equivalent to the probability of observing an occurrence of P^r within d distance after an occurrence of $G(r-1)$. The probability that P^r does not occur within distance d is $(1 - pr(P^r))^d$. Hence, $pr(G)$ is computed as $pr(G) = pr(G(r-1)) \cdot (1 - (1 - pr(P^r))^d)$.

In the case where only delegate occurrences are counted, the probability of observing a delegate occurrence of $G(r-1)$ in a position is estimated by its observed number of delegate occurrences in the input sequence. More specifically, $\hat{pr}(G(r-1)) = \frac{\hat{k}_{G(r-1)}}{L}$. The probability of observing an occurrence of P^r is estimated as $pr(P^r) = \frac{k_P}{L}$. Let the probability that $G(r-1)$ and P^r form a delegate occurrence of G with exactly distance w in between them be $p(w)$. It is the probability of observing an occurrence of P^r at exactly distance w after a delegate occurrence of $G(r-1)$ and no other occurrence of each pattern in between. Hence, $p(w) = \hat{pr}(G(r-1)) \cdot pr(P^r) \cdot \theta^{w-1}$, where $\theta = (1 - \hat{pr}(G(r-1))) \cdot (1 - pr(P^r))$ is the probability of not observing both a delegate occurrence of $G(r-1)$ and an occurrence of P^r. Since w can range from 1 to d, the probability that a delegate occurrence of G is formed by the closest pairing of $G(r-1)$ with P^r is $\hat{pr}(G) = \sum_{w=1}^{d} p(w)$. After expansion, $\hat{pr}(G) = \hat{pr}(G(r-1)) \cdot pr(P^r) \cdot \frac{1-\theta^d}{1-\theta}$.

The input data often comes as multiple sequences. Hence, motifs shared across many sequences are considered as more important than those having a large number of occurrences in a sequence. It is generally accepted that motifs shared across sequences better reflect functionality. The quorum of a pattern therefore provides a better clue about its biological relevance.

Here, Pearson's χ^2 test is used to calculate the statistical significance of the discrepancy between the observed quorum of a pattern and its expected quorum. Given the probability $pr(G)$ of observing an occurrence of G (or $\hat{pr}(G)$ for a delegate occurrence) at a position in a sequence, the expected quorum can be computed. The assessment of statistical significance presented below uses $pr(G)$ for the complete occurrences case. The statistical significance for the case of delegate occurrence can be obtained by replacing $pr(G)$ by $\hat{pr}(G)$ in the formulation.

As in [43], the number of occurrences in a sequence S_i is assumed to follow the Poisson distribution with its expected value $\mu_i = pr(G) \cdot |S_i|$. By the law of rare events, the probability that there is no occurrence in sequence S_i is $e^{-\mu_i}$. Hence, the probability $pr^{(i)}$ that an occurrence appears at least once in the sequence S_i can be computed as $pr^{(i)} = 1 - e^{-\mu_i}$. Accordingly, the expected number of sequences that contain at least one occurrence of G is $\overline{q}_G = \sum_{i=1}^{N} pr^{(i)} = N - \sum_{i=1}^{N} e^{-pr(G)\cdot|S_i|}$. The statistical significance is then defined as follows:

Definition 15. Statistical significance

The *statistical significance* $\chi^2(G)$ of a flexible gap pattern G, which measures the deviation of the observed quorum from the expected quorum, is given below.

$$\chi^2(G) = \frac{(q_G - \overline{q}_G)^2}{\overline{q}_G}$$

The statistical significance for the delegate occurrence case is similarly obtained as $\hat{\chi}^2(G) = \frac{(\hat{q}_G - \hat{\overline{q}}_G)^2}{\hat{\overline{q}}_G}$ where $\hat{\overline{q}}_G = N - \sum_{i=1}^{N} e^{-\hat{pr}(G)\cdot|S_i|}$. A pattern G is said to be significantly over-represented if $\chi^2(G) \geq 3.84$ (corresponding to a p-value of 0.05) and $q_G > \overline{q}_G$. The under-represented pattern where $q_G < \overline{q}_G$ is ignored. Equivalently, it is required that $\chi(G) = \frac{q_G - \overline{q}_G}{\sqrt{\overline{q}_G}} \geq 1.96$

.

5.3 Method for Flexible Gap Pattern Discovery

The efficient and scalable method DFGP for discovering flexible gap patterns is developed by incorporating (1) the use of representative consensus patterns as boxes for assembling flexible gap patterns, (2) the efficient method DOA that identifies delegate occurrences during the pattern assembling process, and (3) the pruning of statistically insignificant patterns to reduce the output size and prevent them from further expanding into patterns of more boxes. As shown in the experiments in Chapter 6, the top ranked flexible gap patterns are effective in identifying co-binding motifs in ChIP-seq datasets and DFGP can efficiently handle large datasets.

5.3.1 DFGP

The cross operation shown in Figure 9 pairs up a flexible gap pattern G' of $k-1$ boxes and a box $P \in \mathbb{R}$ (i.e. a representative pattern) and determines whether this pair (G', P) can form a frequent flexible gap pattern G of k boxes. DFGP implements this cross operation in a depth first search manner. It starts with each representative pattern and attempts to grow it to a pattern of more boxes using DOA until it reaches the maximum number of boxes, or it is not a frequent pattern, or it is not a statistically over-represented pattern. The cross operation implemented in a breadth first search manner needs to store occurrence information for each pattern of $k-1$ boxes before expanding them into patterns of k boxes. When $k \geq 3$, it demands a significantly large amount of computational space. In contrast, depth first search needs only to keep track of the occurrence information of less than r patterns. A stack is used to store patterns to be expanded. A counter is assigned to each pattern in the stack to check whether the expansion of this pattern is completed. The expansion is considered completed if this pattern has gone through pairing up with all consensus patterns. DFGP is summarized in Algorithm 5.1.

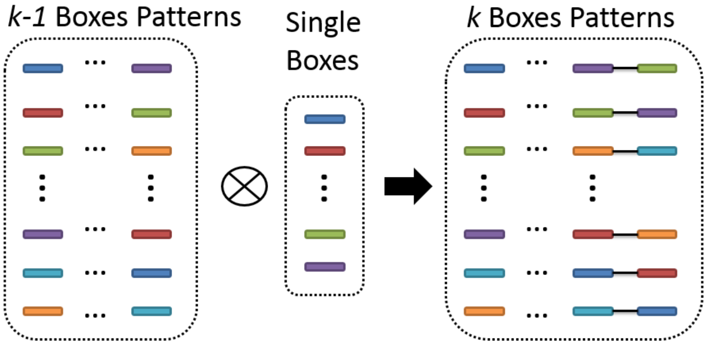


Figure 9. Cross Operation to construct flexible gap patterns of *k-1* boxes to *k* boxes patterns. Each solid rectangle represents a box.

Algorithm 5.1 DFGP

```
Input: multiple sequences $\{S_1, S_2, \ldots, S_N\}$, $th_f$, $r$, $d$
Output:  a set of flexible gap patterns $\mathbb{G}$
1. Obtain a list of representative patterns $\mathbb{R}$ by DRP (Algorithm 3.2)
2. Ensure that occurrences of each pattern in $\mathbb{R}$ are grouped by sequence id
   and sorted in the form of $\{\cdots, L_P^i, \cdots\}$
3. Put each pattern in $\mathbb{R}$ to a stack $\mathbb{T}$ and initialize the counter of each pattern to 1.
4. While $\mathbb{T}$ not empty
       Let the pattern at the top of $\mathbb{T}$ be $G'$, the counter of $G'$ be $x$, the number of boxes of $G'$ be $r'$
       If $x > |\mathbb{R}|$ or $r' = r$ (all patterns in $\mathbb{R}$ are considered or maximum number of boxes is reached)
           If $r' \geq 2$ (ignore patterns of one box)
               Add $G'$ to $\mathbb{G}$ and discard occurrences of $G'$
           End
           Pop $G'$ from $\mathbb{T}$
       Else
           Obtain $G$ by $DOA(G', R[x], d)$ and increase $x$ by 1 (Algorithm 4.1)
           Set the counter of $G$ to 1
           If $|D_G| \geq th_s$ and $\hat{\chi}(G) \geq 1.96$ (exclude expanding infrequent and insignificant patterns )
               Push $G$ to $\mathbb{T}$
           End
       End
End
```

After a set of flexible gap patterns $\mathbb{G}$ is obtained by DFGP, a simple post-processing method *ExtractTopPatterns* shown in Procedure 1 can be applied to extract the top ranked flexible gap patterns. This method goes over the ranked patterns and selects a top ranked pattern to include in the output if this pattern is not similar to the previous selected patterns according to the pattern similarity in definition 8. When measuring similarity between two flexible gap patterns G and G', they are represented in their explicit string form. They are similar if $sim(G, G') \geq 0.8$. This procedure aims to make output of top ranked patterns more distinct.

Procedure 1. ExtractTopPatterns

```
Input: a set of flexible gap patterns $\mathbb{G}$ and $z$ number of top ranked patterns to extract
Output: the set of top patterns $\mathbb{G}_z$
1. Sort the patterns in $G$ according to their statistical significance. Let $x$ be the index for $G$, initially set to $1$
2. While $|\mathbb{G}_z| < z$
       If $\mathbb{G}[x]$ is not similar to any pattern $G$ in $\mathbb{G}_z$
           Add $\mathbb{G}[x]$ to $\mathbb{G}_z$
       End
       Increase $x$ by 1
   End
```

5.3.2 Parameter Settings

As for setting the parameters, th_f is set to $max(0.05N, 5)$ as discussed in Chapter 3 for DRP. In general, the maximum number of boxes in a flexible gap pattern r is often set to 2 or 3 in DNA co-binding motif discovery as a good starting point since the co-binding of two or three TFs is more common. A flexible gap pattern of more boxes could have a very low quorum and becomes infrequent. Maximum distance d is relatively easy to specify since there is a physical constraint over the distance between interacting biomolecules. Setting it to 30 bp is often good enough to capture co-binding motifs as suggested by the paper [2] studying the sequence features around genomic regions bounded by TFs. As the runtime does not depend upon d, a large value can be used, in contrast to existing methods which often require a narrow range. However, too large a value for d would possibly decrease the statistical significance of discovered patterns. It is expected that a consensus pattern can occur frequently after another one within a certain distance if the maximum distance allowed is very large. Hence, if DFGP does not produce any significant pattern, it is likely that d is set too large.

5.3.3 Runtime Analysis of DFGP

Let the number of flexible gap patterns be $N_1, \cdots, N_r$ for different number of boxes. N_1 is the number of patterns of 1 box (representative patterns), so $N_1 = |\mathbb{R}|$. The generation of patterns of k boxes from those of $k-1$ boxes needs to go through N_{k-1} patterns in the expansion process, where each pattern is paired up with $|\mathbb{R}|$ boxes. Each pairing for generating one flexible gap pattern by DOA takes $O(L)$ time. Hence the total runtime for DFGP is $O(\sum_{k=1}^{r-1} N_k|\mathbb{R}|L)$. The space requirement is

$O(rL)$. The runtime for DRP to obtain the set of representative patterns is $O(L+|\mathbb{M}|^2)$, which is negligible comparing to the runtime for the flexible gap pattern generation process.

For $r=2$, the runtime is $O(|\mathbb{R}|^2L)$. For $r \geq 3$, without pruning insignificant patterns, $N_k = O(N_{k-1}|\mathbb{R}|)$, and hence the total runtime is $O(|\mathbb{R}|^rL)$. However, with pruning, $N_k = O(\gamma N_{k-1}|\mathbb{R}|)$, where γ is a trimming factor that reduces the number of patterns of k boxes. Experiments in Chapter 6 showed that γ could be a small ratio and hence its runtime does not increase much for $r=3$.

Let DFGP-M be the method of DFGP except that it uses the set of maximal patterns instead of representative patterns as boxes. The DFGP-M runtime is $O(|\mathbb{M}|^rL)$. It was shown in Chapter 3 for DRP that the set of representative patterns is a small subset of maximal patterns with a reduction ratio α such that $|\mathbb{R}| = \alpha|\mathbb{M}|$. The estimated α is approximately 0.09. For $r=2$, DFGP is much faster than DFGP-M, showing a drastic runtime reduction by using only representative patterns for flexible gap pattern discovery. Experiments in Chapter 6 support this suggestion.

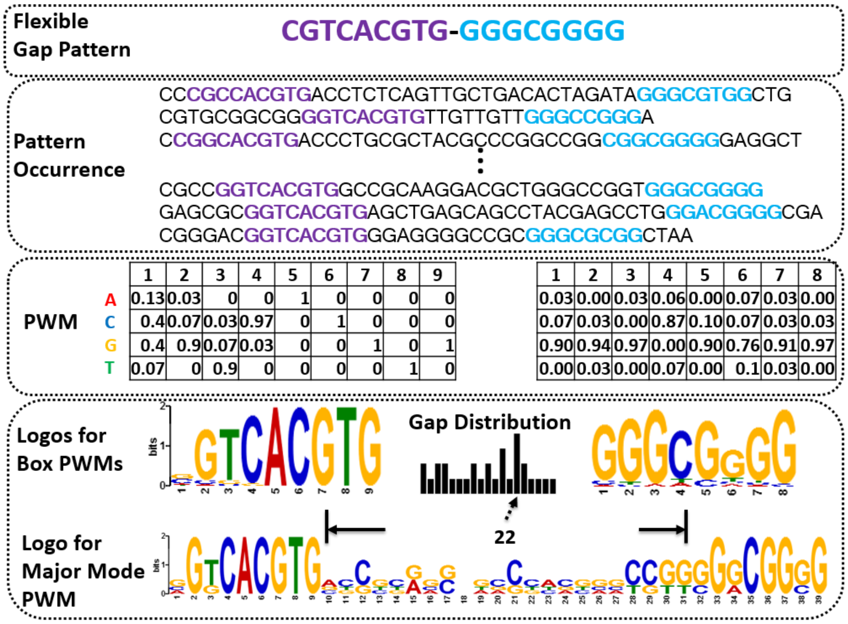
Let DFGP-CO be the method of DFGP except that it obtains a complete set of occurrences for a flexible gap pattern by the method COA. The runtime of DFGP-CO is $O(|\mathbb{R}|^rLd^{r-1})$. DFGP is $O(d)$ times faster than DFGP-CO for $r=2$.

Combining the use of representative patterns, delegate occurrences and the pruning of insignificant patterns, DFGP is much more efficient to handle large datasets and is able to allow large maximum distance.

5.3.4 Conversion of a Flexible Gap Pattern to its PWM Representations

Occurrences of a flexible gap pattern identify specific segments in the input sequences and hence can be aligned together to produce PWMs representing it.

A flexible gap pattern consists of a number of boxes allowing flexible gaps among them. By aligning the occurrences of each box in the flexible gap pattern, a PWM for that box can be obtained by counting the frequency of a character at a site. The gap distribution between two consecutive boxes can be obtained by counting the gaps between their occurrences. Let the major gap between two boxes be the one that appears most frequently. The major mode of a flexible gap pattern, which captures the majority binding mode of a co-binding motif, is obtained as follows: The boxes in the major mode are separated by a fixed distance. The occurrences for the major mode are selected as follow. The selection process starts with all occurrences. It then sequentially identifies the major gap for each flexible gap and excludes occurrences whose corresponding gap is not the major gap along the way. The remaining occurrences are the occurrences of the major mode. In these occurrences, the gap between two box occurrences is the same and hence the span of these selected occurrences is the same. The sequence segments extracted between the start position and the end position of these occurrences for the major mode can be aligned to form the major mode PWM for a flexible gap pattern. Figure 10 shows an example of the flexible gap pattern CGTCACGTG-GGGCGGGG and its corresponding PWMs. The pattern has a set of occurrences in the sequences. The occurrences of each box form a PWM. The gap distribution between box occurrences is obtained. The major gap is identified as 22, meaning that a substantial number of flexible gap pattern occurrences consist of two boxes separated by a fixed distance 22. The major mode PWM is formed by aligning those occurrences having a fixed distance of 22 between two boxes.



CCCGCCACGTGACCTCTCAGTTGCTGACACTAGATAGGGCGTGGCTG
CGTGCGGCGGGGTCACGTGTTGTTGTTGGGCCGGGA
CCGGCACGTGACCCTGCGCTACGCCCGGCCGGCGGCGGGGGAGGCT
⋮
CGCCGGTCACGTGGCCGCAAGGACGCTGGGCCGGTGGGCGGGG
GAGCGCGGTCACGTGAGCTGAGCAGCCTACGAGCCTGGGACGGGGCGA
CGGGACGGTCACGTGGGAGGGGCCGCGGGCGCGGCTAA

	1	2	3	4	5	6	7	8	9
A	0.13	0.03	0	0	1	0	0	0	0
C	0.4	0.07	0.03	0.97	0	1	0	0	0
G	0.4	0.9	0.07	0.03	0	0	1	0	1
T	0.07	0	0.9	0	0	0	0	1	0

1	2	3	4	5	6	7	8
0.03	0.00	0.03	0.06	0.00	0.07	0.03	0.00
0.07	0.03	0.00	0.87	0.10	0.07	0.03	0.03
0.90	0.94	0.97	0.00	0.90	0.76	0.91	0.97
0.00	0.03	0.00	0.07	0.00	0.1	0.03	0.00

Figure 10. The conversion of a flexible gap pattern to its PWM representations. The black dashed arrow points to the major gap in the gap distribution between two boxes of the flexible gap pattern.

Chapter 6
Experiments

6.1 An Overview

The experiments presented in this chapter were designed and conducted to evaluate the performance of DFGP for identifying co-binding motifs in ChIP-seq datasets. DFGP is compared with the existing state-of-the-art motif discovery methods including RISOTTO [70], Bioprospector [64], GLAM2 [67], MEME [10], MEME-Chip [61], MDmodule [63] and Weeder [47]. Among these methods, RISOTTO and Bioprospector are specifically designed for finding co-binding motifs. GLAM2 was shown in its original paper to be able to find structural motifs in protein sequences and was speculated to be a possible application to find complex gapped motifs. Though the primary focus of DFGP aims at finding co-binding motifs, these experiments also help demonstrate its usefulness for identifying rigid gapped motifs and simple motifs as well since rigid gapped motifs are a special type of co-binding motifs where the distance between component motifs are fixed, and simple motifs are just the component motifs in co-binding motifs. Hence, those methods MEME, MEME-Chip and MDmodule which are capable of finding both rigid gapped motifs and simple motifs, and Weeder that is only suitable for simple motif discovery, are included in the experiments. The 68 ChIP-seq datasets used are a subset of 457 ENCODE datasets for human TFs. These datasets contain proposed co-binding motifs for TFs by a computational pipeline (CP) built by Wang et al. in a recent genome research paper [2]. There are five sets of experiments in total to be presented.

The first set of experiments compares the performance of DFGP and other well-known motif discovery methods in terms of identifying the proposed co-binding motifs, canonical motifs and noncanonical motifs in the datasets as well as their runtimes. The second set compares the quality of the co-binding motifs found by DFGP and those proposed by the computational pipeline. The third set compares DFGP with DFGP-M which uses maximal patterns instead of representative patterns and DFGP-CO which counts complete occurrences instead of delegate occurrences to validate the two proposed redundancy reduction methods in the thesis. The fourth set demonstrates the performance of DFGP over varying parameter values. The fifth set evaluates the scalability of DFGP by using all peaks of the ChIP-seq datasets with an average size around 2.5 million bp.

These experiments demonstrate that DFGP outperforms the existing co-binding motif discovery methods in terms of the accuracy of finding the proposed co-binding motifs in the datasets and the runtime. DFGP achieves performance similar to the best simple motif and rigid gapped motif discovery method MEME-Chip in finding canonical motifs and noncanoncial motifs. Co-binding motifs found by DFGP in these datasets is shown to reveal novel biological insights previously unknown to authors of CP. The experiments also show that (1) the adaption of the two proposed redundancy reduction methods in DFGP drastically reduces the computational burden without sacrificing quality, (2) the effectiveness of the statistical significance measures for assessing flexible gap patterns and (3) DFGP is scalable to massive datasets.

6.2 ChIP-seq Datasets

Wang et al. developed the CP method and studied the sequence features around the genomic regions bounded by 119 human TFs. They analyzed 457 ChIP-seq datasets for these 119 TFs generated by ENCODE Consortium [79]. They used ChIP-seq datasets processed by the ENCODE uniform ChIP-seq processing pipeline as described in [80]. A ChIP-seq dataset contains coordinates of genomic regions of length approximately 100 to 500 called peaks that are likely to be bounded by a target TF in the ChIP-seq experiment. Signal quality can be assigned to peaks by the SPP algorithm [81]. Hence the peaks in a ChIP-seq dataset can be ranked according to their signal quality. The ChIP-seq datasets of SPP-based peaks can be downloaded from http://factorbook.org (or http://encodeproject.org/ENCODE/downloads.html under TFBS SPP-based Peaks). These datasets were from the January 2011 freeze pipeline. The genomic coordinates in these ChIP-seq datasets are for the NCBI human reference genome at version GRCH37/HG19. The March 2012 freeze pipeline (http://encodeproject.org/ENCODE/downloads.html under Transcription Factor ChIP-seq Uniform Peaks from ENCODE/Analysis) provides up-to-date ENCODE TF ChIP-seq datasets, increasing from 457 to 690 datasets. The experiments in this thesis used the same ChIP-seq datasets from the January 2011 freeze pipeline as the CP method for comparison purpose. The CP method identifies a canonical motif and several noncanonical motifs for each ChIP-seq dataset. They further identify co-binding motifs which are the canonical and noncanonical motif pairs in 68 out of 457 datasets. Hence these 68 ChIP-seq datasets are used as benchmark datasets for evaluating DFGP and other motif discovery methods. These datasets can be found in the supplementary Table S3 of the CP paper where the predicted interaction mode between the canonical motif and the noncanonical motif is co-binding. These datasets are also summarized in the Appendix A of this thesis. For each dataset, the top ranked 500 peaks are extracted and repetitive elements in these sequences are masked by using the webserver of RepeatMasker [77]. All parameters for RepeatMasker are default values except that the search engine was changed from Abblast to Rmblast to achieve better sensitivity. This ensures that motif discovery tools would not be biased towards discovering those known repetitive elements that are abundant and interspersed in the human genome. The masked sequences of each dataset serve as the input for motif discovery methods. Each input sequence dataset has 500 sequences and its average size is 127152 bp with standard deviation 39044 bp.

The CP built by Wang et al. is briefly summarized as follows: They used a motif discovery tool together with motif quality assessment filters to find TF binding motifs in 457 ChIP-seq datasets. More specifically, they obtained sequence data in the [-50 bp, +50 bp] window around the summits of the top 500 peaks for each Chip-seq dataset, and applied MEME, a de novo motif discovery method, to discover up to 5 motifs per dataset. Hence 2285 motifs were obtained. They designed two quality assessment filters to exclude low quality motifs. In the first filter, a testing set was constructed from peaks ranked 501 to 1000 in the window of [-150 bp, +150 bp] around each peak summit. The control set was obtained from 100 random genomic regions that match the GC content of the testing set. In the second filter, a testing set was constructed from peaks ranked 501 and beyond in the [-150 bp, +150 bp] window and the control set was obtained from the regions of 300bp long flanking the window. The motifs were retained if they were more enriched in the testing set than in the control set. After filtering, 1092 motifs were left for further analysis. They manually merged redundant motifs by taking account of the literature support and information on DNA-binding domains of TFs to derive 79 distinct motifs represented by PWMs which can be found in the supplementary Table S2 for their paper. The top ranked motif discovered from a dataset is called the canonical motif and others are called noncanonical motifs. They used FIMO [82] to scan the whole ChIP-seq dataset to look for peaks that contain the canonical motif. Similarly, the peaks containing the noncanonical motif were obtained. The percentage of peaks that contain both motifs, known as enrichment, was used as an indicator to determine whether or not to report such pair as a co-binding motif. They proposed that 68 of out 457 datasets contain some co-binding motifs. They showed the distance between the canonical and the noncanonical motif in many co-occurring pairs has a preference within 30 bp.

6.3 Evaluation Method

If possible, each motif discovery method is asked to return 30 top ranked patterns in the form of PWMs. For DFGP, a flexible gap pattern of r is associated with r PWMs for its boxes and $r - 1$ major mode PWMs for itself and its prefix patterns with at least 2 boxes. For Bioprospctor, a pattern contains two PWMs. For RISOTTO, a structured model of r boxes produces r PWMs for each box.

For GLAM2, MEME, MEME-Chip, MDmodule and Weeder, each reported pattern corresponds to one PWM.

The 79 distinct motifs represented in PWMs provided by Wang et al. serve as a reference motif database. Those discovered patterns can be searched against this reference database to find matches. To determine whether a discovered PWM matches a motif in the motif reference database, TOMTOM [83] is used. Given a PWM, TOMTOM attempts to search for PWMs in a database that match the query by aligning the query PWM and a database PWM and calculating the similarity score for their overlapped part. PWMs in the database are reported to match the query PWM if the E-value of their matching score is above a threshold. As suggested by the authors of TOMTOM in the DREME paper [49], this threshold is set to 0.05. The E-value indicates if the motif database of the same size is randomly constructed, the expected number of matches to retrieve from such a database of random motifs. A query PWM may get multiple matches from the database, but only the match with the lowest E-value is kept.

A discovered pattern matches a co-binding motif (in this case, a canonical and noncanonical motif pair) if any of its two PWMs match the canonical motif and noncanonical motif respectively or one of its PWM matches both the canonical and noncanonical motif where the matched regions need to be at least 5 sties apart. The latter case considers major mode PWMs from DFGP and single PWMs from GLAM2, MEME, MEME-Chip, MDmodule and Weeder that might capture co-binding motifs theoretically. A discovered pattern matches a canonical or a noncanonical motif if one of its PWM matches that motif.

For the 68 datasets in the experiments, each dataset has one or two co-binding motifs. A method is said to succeed in finding co-binding motifs in a dataset if one of its top ranked patterns matches a co-binding motif in that dataset. The performance of a method in identifying co-binding motifs, canonical motifs and noncanonical motifs is measured as the percentage of datasets in which it succeeds in finding those corresponding motifs.

6.4 Parameter Settings for Motif Discovery Methods

DFGP is run using its default settings. The minimum frequency th_s is set to $max(0.05N, 5)$. In most experiments, the number of sequences is $N = 500$, hence $th_s = 25$. The number of boxes r is set to 2 to look for a canonical and noncanonical motif pair. The maximum number of gaps d is set to 30 as suggested by Wang et al. since DFGP is insensitive to d and its runtime does not depend on d as well.

Hence, different values of d can be tried, but d in DFGP needs no fine tuning. In general, setting it to a large value such as 30 is a good guess since there are physical constraints in the interactions between biomolecules. The latter experiment shows that a larger value such as 50 further improves the performance of DFGP.

RISOTTO is run using the following parameters. The number of boxes for a structured model is set to 2. Each box length is set to 6, which is a good guess for the length of the core part of a TF motif. The number of mutations allowed for each box is set to 1, the same way as DFGP. The minimum and maximum distance between boxes is set to 1 and 10 respectively. The minimum required quorum is set to $0.1N = 50$. RISOTTO can discover models from repeat masked sequences. However, its pattern significance evaluation program cannot shuffle sequences masked with “N” characters. The program is modified so that it shuffle the unmasked sequences instead of masked ones for computing the model score. The runtime of this program depends on the number of structured patterns outputted by RISOTTO. Using the settings above, RISOTTO outputs 532120 models in 41 seconds for a moderate size dataset (150546 bp). However, it took 7.65 hours to compute scores for the models. Increasing the maximum distance allowed from 10 to 30 leads to an output of 1976770 models and the ranking of these models took 2.72 days. The very time consuming computing process for assessing model statistical significance makes it impossible to try a more relaxed motif template such as allowing box length to range from 4 to 12 and the minimum quorum to be 25. RISOTTO under this setting generated 59730851 models in 1.75 hours, not including the time for computing statistical significance. Other parameters not mentioned are in default values. Its program can be obtained from the web at http://www.lx.it.pt/~asmc/software/risotto.html.

Bioprospector is run with the following parameter settings. Bioprospector is a 2-box motif discovery method hence it is restricted to finding co-binding motifs of two components. The minimum and maximum distance between boxes is set to 1 and 30 respectively. Each box length can be 6, 8 and 10. So there are 9 combinations in total. For each combination of box lengths, Bioprospector is asked to return 3 top ranked patterns. Hence, the total number of patterns to be outputted is 27. The option to search for motifs in the reverse and complement sequences is turned on. Default values are used for other parameters not mentioned.

The minimum required quorum for GLAM2 is set to 25. In the default settings, the program tries different motif lengths for PWM from 2 to 50. It adopts a faster algorithm which deviates slightly from the strict definition of simulated annealing. This option is changed to a slow algorithm that implements strict simulated annealing, trying to find the best motif. The alignment runs is increased from 10 to 30 in order to obtain 30 top ranked PWMs. The option to search for motifs in the reverse and complement sequences is turned on.

MEME is run with most of its parameters set to default. In the default settings, it tries motif lengths for PWM ranging from 8 to 50. The option to search motif in reverse and complement sequences is enabled to increase its ability of finding motifs. MEME is asked to report only 5 PWMs per dataset because it takes increasingly longer time to discover more motifs. In these settings, it can take a couple hours to finish for a mid-sized dataset.

MEME-Chip is run using its default settings. It tries motif lengths for PWM ranging from 6 to 30 with the reverse and option to search off for its MEME part. Its DREME part tries sequence patterns of length from 4 to 8. MEME would report 3 PWMs and DREME would report about 6 sequence patterns, so there are approximately 9 patterns in total in the output.

The programs for GLAM2, MEME and MEME-Chip can be obtained at http://meme.nbcr.net/meme/.

MDmodule is run using default settings. PWMs with motif length ranging from 8 to 24 (its maximum allowed value) with step 4 was tried, and for each length, MDmodule is asked to report 5 PWMs. Hence, there are 25 patterns in the output. Its program can be obtained at http://www.math.umass.edu/~conlon/mr.html.

Weeder is run with default settings. It is fixed to find motif lengths 6, 8 and 10 in its program. The program can be obtained at http://159.149.160.51/modtools/.

6.5 Results

Five set of experiment results are presented. The first set focuses on the comparison between DFGP and other motif discovery methods in terms of the ability to find co-binding motifs, canonical motifs and noncanonical motifs, and the runtime. The second set compares the difference between co-binding motifs found by DFGP and those found by the CP method. The third set compares DFGP with DFGP-M and DFGP-CO, which aims at demonstrating the benefits of DFGP when using representative patterns and delegate occurrences. The fourth set investigates how the performance of DFGP varies as its input parameters changes across a certain range. The final set shows the scalability of DFGP by applying it to whole ChIP-seq datasets of all peaks instead of top ranked 500 peaks.

6.5.1 Comparison of DFGP with other Motif Discovery Methods

Figure 11 shows the performance of motif discovery methods in identifying co-binding motifs, canonical motifs and noncanonical motifs proposed by the CP method in 68 ChIP-seq datasets. The details of the matching between discovered patterns and motifs for each method are shown in Appendix B. The top ranked flexible gap patterns by DFGP identify such co-binding motifs in 38% of datasets. This further recapitulates some of these proposed motifs. RISOTTO and Bioprospector has succeeded in finding co-binding motifs in 10% and 5% of the datasets. The ineffectiveness of RISOTTO to identify co-binding motifs is likely due to its restricted motif template. However, whether a more relaxed template can improve its performance in co-binding motifs identification cannot be tested since its pattern statistical significance evaluation program is too time consuming. Bioprospctor does not perform well in finding co-binding motifs. This is probably due to the presence of the dominant canonical motif and the relative low enrichments of co-binding motifs which aggravate the tendency of Gibbs Sampling method towards a local optimal solution. It is shown that GLAM2 is unable to identify co-binding motifs even though its authors have speculated that it might be able to find complex gapped motifs. Its accuracy in finding canonical motifs is close to that of MDmodule and Weeder and is just slightly lower than that of DFGP, MEME and MEME-Chip. However, its accuracy in finding noncanonical motifs is the lowest among all methods tested. This demonstrates that GLAM2 is more suitable for finding the most dominating simple motifs or rigid gapped motifs. MEME, MEME-Chip, MDmodule and Weeder are not designed for co-binding motif discovery and hence their inability to find co-binding motifs is anticipated.

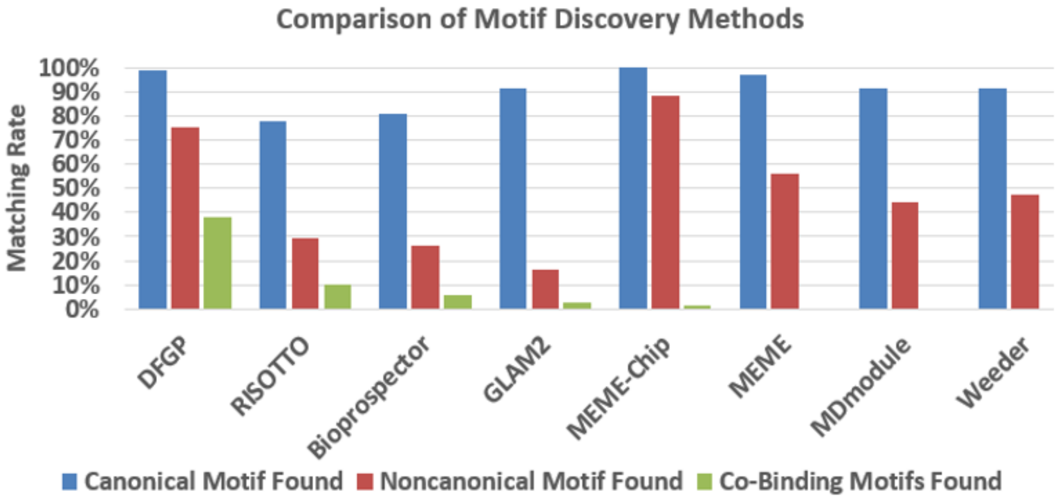


Figure 11. The performance of motif discovery methods over 68 ChIP-seq datasets. The performance is measured by the percentage of datasets in which the canonical motif, the noncanonical motifs and the co-binding motifs are identified respectively by a method.

As for identifying the canonical and noncanonical motifs, MEME-Chip achieves the best accuracy. The accuracy of DFGP is just a slightly lower than MEME-Chip, achieving the second best place. The performances of MEME, MDmoudle and Weeder are similar. RISOTTO, Bioprospector and GLAM2 perform poorly in finding noncanonical motifs. For RISOTTO, the possible reason is that the box length for a structured model is restricted to 6 and the top ranked models tend to identify dominating motif better than secondary motifs. For Bioprospector and GLAM2, the reason is that they always aim to find the best solution in the Gibbs sampling process.

Figure 12 shows the average runtime for motif discovery methods over 68 datasets. The average runtimes for DFGP, MEME-Chip and Weeder are within 7, 10 and 11minuntes respectively and hence are very close. The average runtimes for Bioprospector, RISOTTO, MEME and GLAM2 are in the range of 1 to 5 hours. Note that GLAM2 is running in a slow but more accurate mode. By switching it to the fast mode, the average runtime drops from 5 hours to 5 minutes. The performance of fast mode GLAM2 leads to a slightly worse performance than the slow mode in finding noncanonical and co-binding motifs. Comparing with the co-binding motif discovery methods Bioprospector and RISOTTO, DFGP is approximately 8 and 18 times faster respectively. MDmodule with average runtime 15 seconds is the fastest method for simple motif and rigid gapped motif discovery. However, its performance for noncanonical motif identification is substantially worse than

MEME-Chip and DFGP. In addition, MDmodule expects that the input sequences are ranked. The reason MEME-Chip is much faster than MEME is that MEME-Chip runs its subprogram MEME in a less exhaustive way as indicated in their parameter settings.

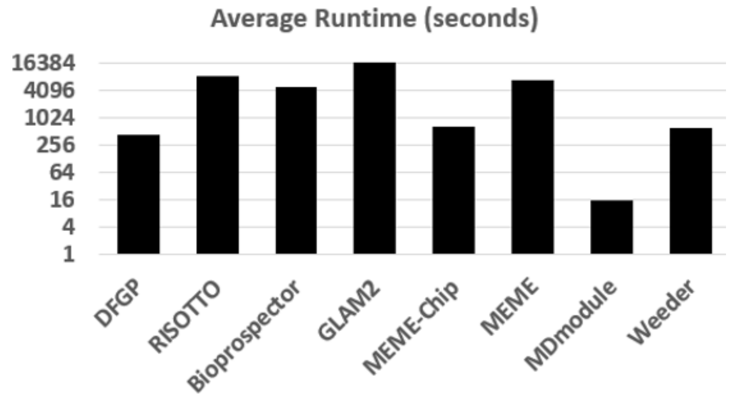


Figure 12. The average runtime of motif discovery methods over 68 ChIP-seq datasets. The y-axis is in log_2 scale.

In summary, DFGP outperforms RISOTTO, Bioprospector and GLAM2 in terms of the capability of co-binding motif identification and the runtime. DFGP achieves similar performance as MEME-Chip, one of the best simple and rigid gapped motif discovery methods, in terms of the capability of identifying canonical and noncanonical motifs as well as runtime.

6.5.2 Comparison of DFGP with the Computational Pipeline

Though 38% of the datasets in which co-binding motifs proposed by the CP method are found by DFGP, there are approximately 62% of the datasets for which DFGP fails to identify the proposed co-binding motifs. To investigate this discrepancy, the enrichment of co-binding motifs proposed by the CP method and those proposed by DFGP in a dataset is compared. The enrichment of a co-binding motif in a ChIP-seq dataset is the percentage of peaks that contain the co-binding motif used by the CP method. The patterns discovered by DFGP undergo the following procedures to obtain co-binding motifs whose component motifs are known in the reference motif database. A top ranked flexible gap pattern reports a co-binding motif for a dataset if this pattern matches the canonical motif in that dataset and another motif not similar to the canonical motif in the reference motif database. If a motif is similar to a canonical motif, then the co-binding motif formed by them can have high enrichment but probably due to high overlapping of the two TF motifs. To obtain similar motifs for a motif in the motif database, this motif is used as a query motif for TOMTOM to extract matching motifs in the database. The summary of motifs and their similar ones can be found in the Appendix C. Three co-binding motifs MAX-USF, NFE2-AP1 and UA5-GABP proposed by the CP method are considered as exceptional cases though MAX, NFE2 and UA5 are similar to USF, AP1 and GABP respectively.

After obtaining co-binding motifs found DFGP, their enrichment in the datasets is calculated as in the CP method. The details of the enrichment of each co-binding motif proposed by the CP method and DFGP are shown in Table S4 in Appendix D. In the table, co-binding motifs proposed by DFGP, which have higher enrichment than those by the CP method are all retained. For those datasets without higher enrichment co-binding motifs found by DFGP, the co-binding motif with the highest enrichment is retained.

The summary result for enrichment of co-binding motifs proposed by both methods is shown in Figure 13. Co-binding motifs found by DFGP have the same highest enrichment as those proposed by the CP method in 25% of the datasets, which are labelled as TYPE1 datasets. Co-binding motifs found by DFGP capture those motifs reported by the CP method but are novel and have higher enrichment in 13.2% of the datasets, which are labelled as TYPE2. In 27.9% of the datasets labelled as TYPE3, DFGP does not obtain co-binding motifs proposed by the CP method but find novel co-binding motifs that have higher enrichment. DFGP fails to identify co-binding motifs that have higher enrichment than those reported by the CP method in the remaining 33.8% of the datasets, which are labelled as TYPE4. In summary, among 66% of 68 ChIP-seq datasets, DFGP identifies co-binding motifs that have higher or equal enrichment in the dataset. This indicates that, among 61.7% of the datasets for which DFGP fails to find the previous proposed co-binding motifs, around half of them contain stronger co-binding motifs. DFGP discovers higher enrichment co-binding motifs than the CP method in 41% of the datasets combining the TYPE2 and TYPE3 datasets. This suggests that the noncanonical motifs found in the motif discovery step by MEME of the CP method miss important TF motifs.

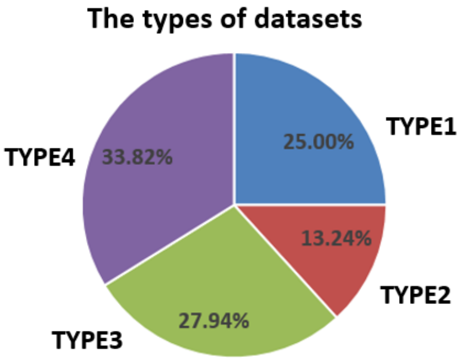


Figure 13. Four types of 68 ChIP-seq datasets. TYPE1: datasets where co-binding motifs agreed by Wang et al. and DFGP have the best enrichment. TYPE2: datasets where DFGP captures the co-binding motifs proposed by the CP method but has found other novel motifs that have better enrichment. TYPE3: datasets where DFGP does not capture co-binding motifs proposed by the CP method but identifies novel motifs that have better enrichment. TYPE4: datasets where DFGP fails to capture co-binding motifs proposed by the CP method and the motifs proposed by DFGP have lower enrichment.

Here several examples are shown to illustrate that the CP method might miss some higher enrichment co-binding motifs and DFGP identifies some novel co-binding motifs that reveal novel biological insights. Wang et al. reported that the canonical motif YY1 and a noncanonical motif GABP form a co-binding motif for the four ChIP-seq datasets whose target TF is YY1. However, the co-binding motifs found by DFGP suggest that it is more probable that motif YY1 and an unannotated motif UA5 (a motif that is identified by the CP method and is similar to motif GABP) form a co-binding motif. The enrichment of YY1-UA5 is consistently higher than YY1-GABP in all the four YY1 datasets. In the dataset whose cell line is K562b (wgEncodeSydhTfbsK562bYy1UcdAlnRep0), YY1-UA5 is enriched in 23.5% of the peaks whereas the enrichment of YY1-GABP is 20.1%. In addition, the flexible gap pattern M16 discovered in this dataset shows that UA5 is more compatible to form a co-binding motif with YY1 in Figure 14. Note that in the dashed line box the characters “CC” match better with UA5 than GABP. More interesting, the co-binding motif YY1-UA5 also matches perfectly with another unannotated motif UA4 that has 31.5% enrichment in the dataset. This indicates that the co-binding between YY1 and UA5 significantly explains UA4, which is unnoticed by Wang et al. The existence of UA4 provides further evidence that YY1-UA5 is a strong co-binding motif in YY1 ChIP-seq datasets. The intermediate region between UA5 and YY1 is more conserved than the tail of UA5 alone. This indicates that some

UA5 site does not associate with YY1. In fact, UA5 occupies 30% of the peaks while the enrichment of YY1-UA5 is 23.5%. Nonetheless, most of the UA5 sites co-occur with YY1. UA4 and UA5 were reported as the canonical motifs in the ChIP-seq dataset whose target TF is THAP1. DFGP also discovers the co-binding motif YY1-UA5 which is enriched in 8% of the peaks in the THAP1 dataset. The enrichment of UA4 and UA5 in the dataset is 14% and 35% respectively. Hence YY1-UA5 contributes significantly to the existence of UA4. Based on the evidences that (1) YY1-UA5 significantly explains the motif UA4, and (2) UA5 has higher enrichment than UA4 in the THAP1 dataset, the hypothesis is that UA5 other than UA4 is the canonical motif responsible for the binding of THAP1 in DNA sequences. The CP method reports the canonical motif UA5 and a noncanonical motif GABP (a motif similar to UA5) in the THAP1 dataset forms a co-binding motif, yet missing the co-binding motif UA5-YY1.

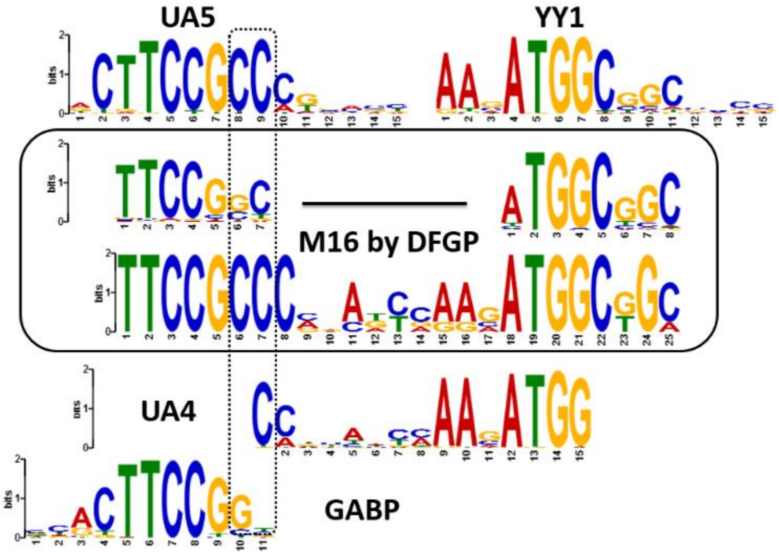


Figure 14. The co-binding motif YY1-UA5 in the ChIP-seq dataset wgEncodeSydhTfbsK562bYy1UcdAlnRep0. The motif within the black rectangle is the flexible gap pattern M16 discovered by DFGP in the dataset. Other motifs are from the motif reference database reported by Wang et al. The dashed line box shows that characters "CC" match better to UA5 than GABP.

However, a small discrepancy between the enrichment of UA4 and YY1-UA5 indicates that UA4 cannot be entirely the result of YY1-UA5. DFGP discovers another co-binding motif YY1-ESRRA in three of the four YY1 ChIP-seq datasets. The flexible gap pattern M2 discovered in the K562b cell line dataset as shown in Figure 15 demonstrates that YY1-ESRRA also contributes to the existence of

UA4. Interestingly, YY1-ESRRA prefers the reverse and complementary orientation from the perspective of YY1-UA5. The enrichment of YY1-ESRRA in the K562b cell line dataset is 8.3%, revealing that this co-binding motif is a relatively weaker motif. The 31.8% combined enrichment of co-binding motifs YY1-UA5 and YY1-ESRRA probably explains the 31.5% enrichment of UA4 in the dataset.

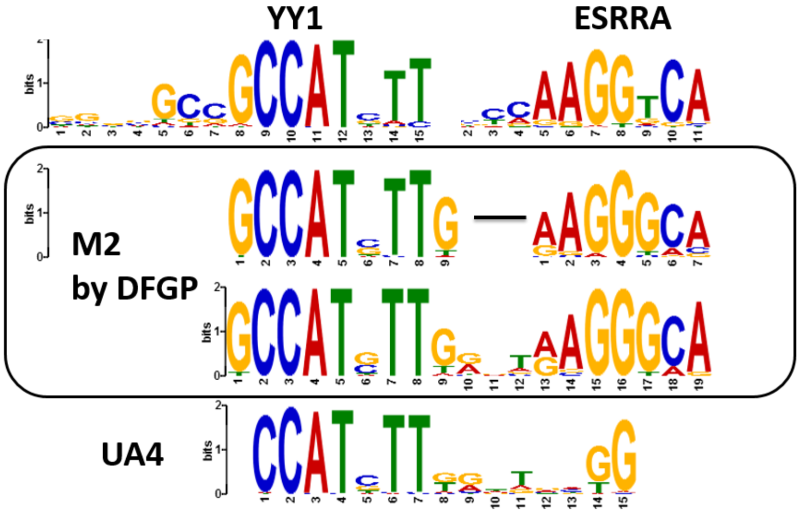


Figure 15. The co-binding motif YY1-ESRRA in the ChIP-seq dataset wgEncodeSydhTfbsK562bYy1UcdAlnRep0. The motif within the black rectangle is the flexible gap pattern M16 discovered by DFGP in the dataset. Other motifs are from the motif reference database reported by Wang et al.

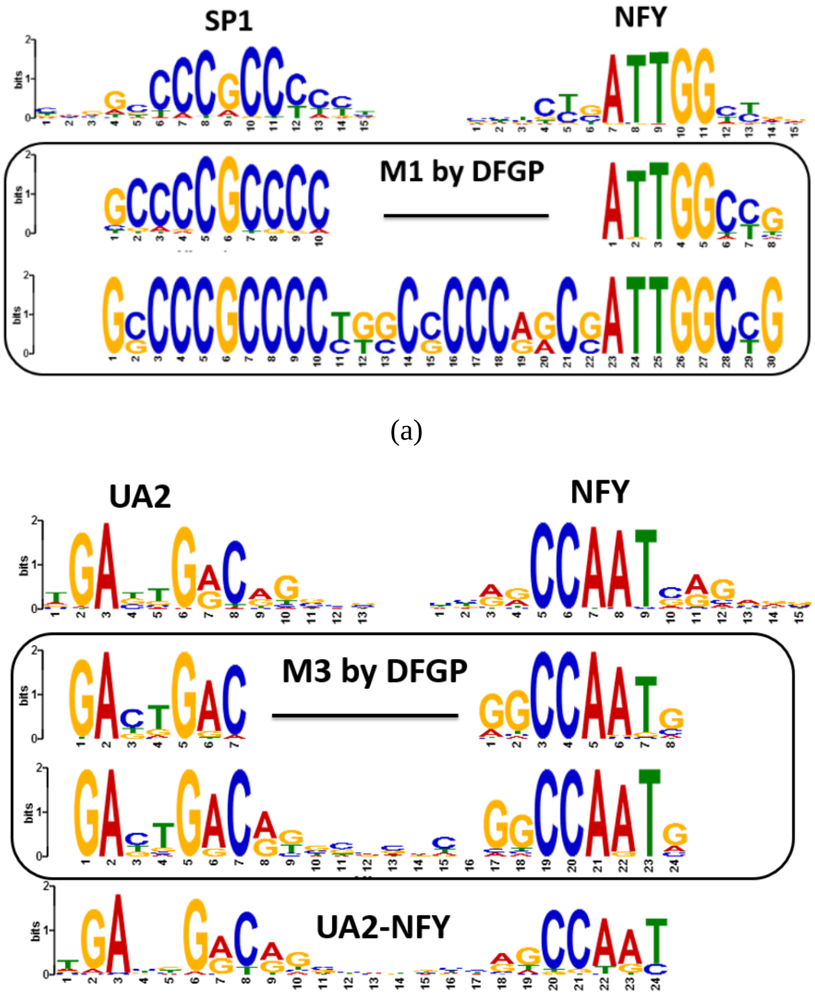
For those 30% of datasets that DFGP fails to find co-binding motifs with higher or equal enrichment than those reported by the CP method, there could be several reasons.

Firstly, the interaction distance between the canonical and noncanonical motifs is in a range longer than the maximum distance of 30 specified in the experiments. For example, in a later experiment, when the maximum allowed distance increases to 50, the matching rate increases from 38% to 50%.

Secondly, the enrichment of some co-binding motifs found by DFGP is actually very close to that of those reported by the CP method. The co-binding motif MAX-CTCF reported by the CP method enriches in 18.6% of peaks in the dataset wgEncodeSydhTfbsH1hescMaxUcdAlnRep0 while MAX-SP1 found by DFGP has 18.4% enrichment.

Thirdly, the canonical motif and the noncanonical motif of some co-binding motifs reported by the CP method may not form a co-binding motif directly but through an intermediate motif. For example, in the dataset wgEncodeHaibTfbsGm12878Pbx3Pcr1xAlnRep0 whose target TF is PBX3, UA2 is the canonical motif appearing in 42% of the dataset. SP1 has 36% enrichment, and UA2-SP1 was reported by the CP method as the co-binding motif. However, DFGP discovers co-binding motifs UA2-NFY and NFY-SP1, which have 12% and 13.6% of enrichment in the dataset respectively, but does not identify UA2-SP1. These two co-binding motifs are shown in Figure 16.

I therefore hypothesizes that UA2 and SP1 does not directly form a co-binding motif but through the bridge motif NFY. This is supported by the following evidences. In the three ChIP-seq datasets of TF SP1, the co-binding motif SP1-NFY is discovered by DFGP as the highest enrichment motif. Wang et al. reported this co-binding motif SP1-NFY in two of the three datasets and SP1-UA2 in one dataset. SP1-UA2 in the dataset wgEncodeHaibTfbsGm12878Sp1Pcr1xAlnRep0 occupies 14.4% of peaks while SP1-NFY is enriched in 17.5% of the peaks. In the ChIP-seq dataset of TF NFY wgEncodeSydhTfbsK562NfybStdAlnRep0, Wang et al. reported the co-binding motif NFY-USF which is enriched in 16% of peaks. In contrast, DFGP discovers both SP1-NFY and NFY-UA2 which have 49.3% and 20% enrichment in the dataset. Interestingly, the co-binding motif SP1-UA2 does not show up as a top ranked pattern for all these ChIP-seq datasets of TF SP1, NFY and PBX3. The dataset of TF PBX3 is the only one that SP1-UA2 is more enriched than SP1-NFY and NFY-UA2. However, the reason is that both SP1 and UA2 are highly concentrated in the dataset with enrichment equal to 42% and 36% respectively. Though they co-occur in a large fraction of peaks, they could be far apart or irrelevant to functional co-binding for some of the peaks. This indicates that some of the co-binding motifs reported by Wang et al. might not directly involve in co-binding. DFGP avoids this issue by having the maximum distance to constraint motifs involved in co-binding as well as using statistical significance to assess the pattern quality instead of using only the enrichment statistics. This paper estimates that the functional co-binding of SP1 and UA2 in the PBX3 dataset is enriched in less than 12% of the peaks, a bit lower than the 20% enrichment of NFY-UA2.



(a)

(b)

Figure 16. The co-binding motifs (a) SP1-NFY and (b) UA2-NFY in the ChIP-seq dataset wgEncodeHaibTfbsGm12878Pbx3Pcr1xAlnRep0. The motifs within the black rectangle are the flexible gap patterns M1 and M3 discovered by DFGP in the dataset. Other motifs are from the motif reference database reported by Wang et al.

Fourthly, the enrichment of some co-binding motifs reported by Wang et al. may be too low and therefore cannot be identified by DFGP.

In summary, though the CP method developed by Wang et al. could proposed some novel co-binding motifs, it has several limitations: (1) the co-binding motifs proposed might not be functionally relevant co-binding between its component motifs due to the lack of distance constraint between component motifs and the use of only enrichment statistics to evaluate; (2) the number of co-binding motifs proposed is limited since the number of motifs reported by MEME is up to 5; (3) a substantial number of important co-binding motifs with high enrichment could be missed since some of the components motifs are not strong alone and MEME searches motifs in a local optimal manner. In contrast, co-binding motifs discovered by DFGP have the distance constraint imposed between their component motifs and are statistically significant. Hence, these physically close component motifs are very likely to be functionally involved in the co-binding interaction. The discovered co-binding motifs are not limited to a few canonical and noncanonical pairs but are more comprehensive and consistent. They reveal interesting biological insights previously unknown to researchers.

6.5.3 Comparison of DFGP, DFGP-M and DFGP-CO

This thesis proposes two novel reductions to drastically reduce the enormous computational burden of the deterministic approach for co-binding motif discovery. The first one is to use representative patterns instead of maximal patterns as boxes for constructing flexible gap patterns. It thus eliminates a great amount of box combinations. The second one is to extract a set of delegate occurrences instead of the complete set of occurrences during the process of building a flexible gap pattern from its prefix pattern and its last box. It thus achieves a runtime that is not dependent on the maximum allowed distance as well as the number of boxes. It is natural to question whether these two reduction methods would risk losing too much information. The following experiments compare the performance of DFGP, DFGP-M and DFGP-CO in terms of their ability to find co-binding motifs and the runtime. DFGP-M is the same as DFGP except that a large set of maximal patterns is used as boxes for flexible gap pattern assembling. DFGP-CO is the same as DFGP except that the method COA is used instead of DOA to extract a complete set of occurrences for a flexible gap pattern during its construction.

As shown in Figure 17 (a), the use of representative patterns or counting only delegate occurrences does not impede the ability of DFGP in identifying co-binding motifs. There is only a very slight drop from 39.7% to 38.2% of datasets that contain top ranked patterns matching co-binding motifs proposed by Wang et al. The number of datasets where noncanonical motifs are found by DFGP-M decreases moderately, suggesting the top ranked patterns are stronger but many are redundant patterns. The Figure 17 (b) shows the runtime of these three methods. DFGP is 18.5 times and 3.5 times faster than DFGP-M and DFGP-CO for the average runtime over 68 datasets. Hence, DFGP achieves great reduction in runtime without sacrificing the ability to identify co-binding motifs. This demonstrates the effectiveness of reducing redundant consensus patterns and flexible gap pattern occurrences by the use of representative patterns and delegate occurrences. The ability of DFGP and DFGP-CO to identify co-binding also shows that the statistical significances defined for complete occurrences and delegate occurrences respectively are effective in ranking flexible gap patterns.

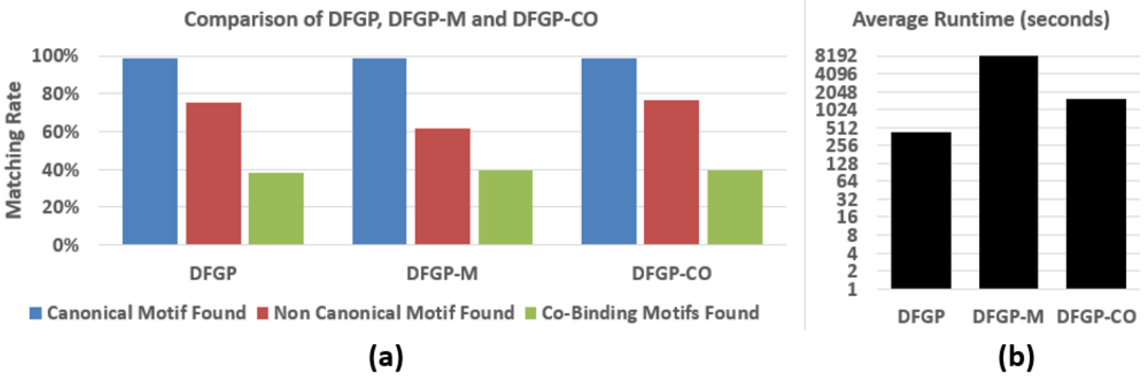


Figure 17. The performance of DFGP, DFGP-M and DFGP-CO over 68 ChIP-seq datasets. (a) The percentage of datasets in which the canonical motif, the noncanonical motifs and the co-binding motifs are identified respectively. (b) The average runtime. The y-axis is in log_2 scale.

6.5.4 Performance of DFGP as Parameter Values Vary

DFGP has three parameters: the frequency threshold th_f, the maximum distance allowed d and the number of maximum number of boxes r set in the flexible gap pattern discovery process. How these parameters affect the ability and runtime of DFGP in motif discovery are shown in this section. Though the similarity threshold th_s, which defines the pattern neighborhood for the extraction of representative patterns, is set to 0.8 according to the theoretical analysis in Chapter 3.3, its effect is also shown.

The setting of r depends upon the need of the user. Setting r to 2 is usually a good starting point especially for ChIP-seq datasets. Flexible gap patterns of 2 boxes could provide abundant information about how TF pairs interact and can have an application in uncovering combinatorial regulatory code. Nonetheless, increasing r from 2 to 3 slightly improves the accuracy of finding the co-binding motifs proposed by the CP method as shown in Figure 18 (a). The matching accuracy for co-binding motifs increases slightly from 38% to 44%, showing that the flexible gap patterns of 3 boxes capture some more information, an indication of the capability and proficiency of DFGP in more general co-binding motif discovery scenarios where r could be increased. Such capability is very important to unveil the regulatory mechanisms in noncoding DNA regions. However, as expected in the current research settings, there are not many patterns of 3 boxes discovered due to the fact that the frequency of a co-binding motif consisting of 3 simple motifs is often very low (i.e., combinations of three co-binding TFs provides very specific regulatory role and have much less enrichment in a dataset). Thus, it is missed by DFGP with $th_f = 0.05N$. The runtime of DFGP shown in Figure 18 (b) does not increase much for $r = 3$ due to the statistical pruning which excludes insignificant 2-boxes patterns when expanding to 3-boxes patterns.

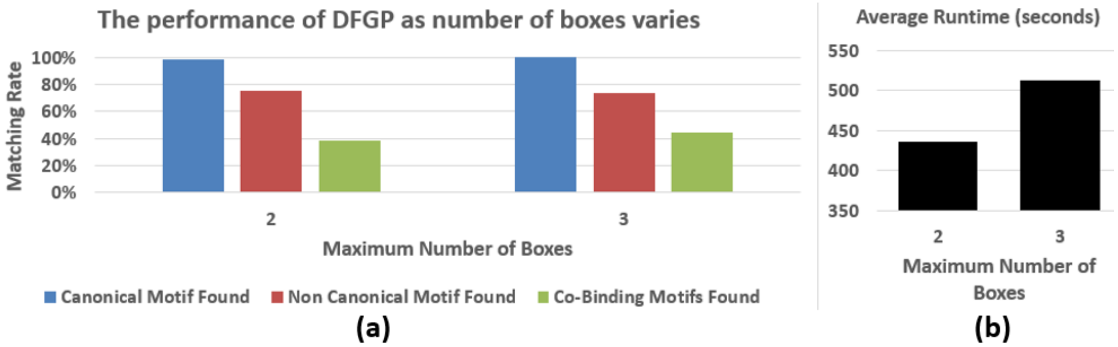


Figure 18. The performance of DFGP over 68 ChIP-seq datasets for $r = 2$ and $r = 3$. (a) The percentage of datasets in which the canonical motif, the noncanonical motifs and the co-binding motifs are identified respectively. (b) The average runtime.

To investigate the effect of the maximum distance d, DFGP was run with its value changing from 10 to 80 with a step size of 10. Figure 19(a) shows that the accuracy in matching co-binding motifs proposed by Wang et al. increases as d increases from 10 to 50 and that the best matching rate 50% occurs at $d = 50$. This indicates that some co-binding motifs have a long interaction range or their component motifs do not directly co-bind. However, accuracy starts to decrease as d continues to increase from 60 to 80. The reason is that the statistical significance for patterns decreases as the maximum distance becomes too large. As a result, some patterns become statistically insignificant and thence are excluded from the output. Nonetheless, setting d to 30 is good enough for DFGP to explore co-binding motifs in a dataset and a suitable range for it is from 30 to 60. As expected, the runtime of DFGP does not depend upon d as shown in Figure 19(b), which is an advantage in finding long range interacting biomolecules.

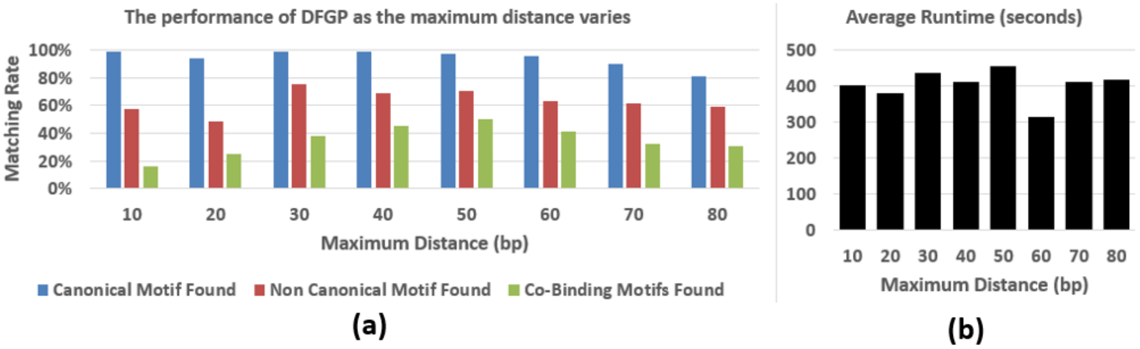


Figure 19. The performance of DFGP over 68 ChIP-seq datasets as d varies. (a) The percentage of datasets in which the canonical motif, the noncanonical motifs and the co-binding motifs are identified respectively. (b) The average runtime.

To investigate the effect of the frequency threshold, th_f was varied from $0.01N$ to $0.09N$ with a step size of $0.02N$. Since $N = 500$ in the experiments, DFNG was run with th_f ranging from 5 to 45 with a step size of 10. A significant performance drop can be seen in Figure 20(a) at $th_f = 5$. A low frequency threshold introduces many noisy consensus patterns of long length due to repetitive elements in one or two sequences, which are not masked by the repetitive element mask program RepeatMasker. In addition, statistical significance is undermined with small samples. Hence, th_f should not be too low to avoid this issue. This parameter value cannot be too high as well. Otherwise, weaker simple motifs would be excluded. As shown in Figure 20(a), the accuracy for the noncanonical motifs decreases as th_f increases. A suitable range for th_f is from $0.03N$ to $0.07N$. The runtime of DFGP shown in Figure 20(b) decreases as th_f increases as the result of the decreasing number of consensus patterns. Note that when N is small, i.e. 10, other methods might be better alternatives.

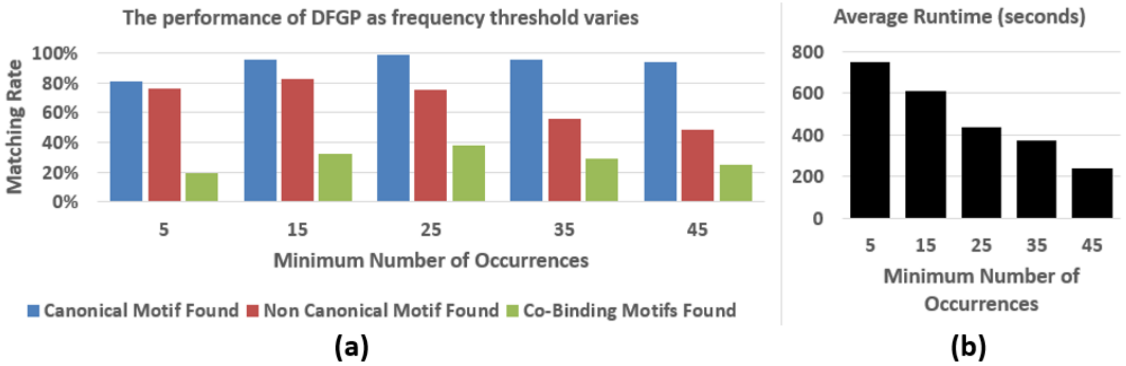


Figure 20. The performance of DFGP over 68 ChIP-seq datasets as th_f varies. (a) The percentage of the datasets in which the canonical motif, the noncanonical motifs and the co-binding motifs are identified respectively. (b) The average runtime.

To investigate the effect of the similarity threshold, th_s was varied from 0.7 to 0.9 with a step size of 0.05. Figure 21(a) shows that the performance of DFGP remains relatively the same as th_s varies from 0.9 to 0.75 but has a significant drop when th_s is set to 0.7. Hence, DFGP is not sensitive to the change of th_s when it is not set to a too small value, which would result in selecting too few representative patterns as the candidate boxes for assembling flexible gap patterns and thus causing great information loss. However, a large th_s would retain a large proportion of frequent solid patterns as representative patterns leading to the significant increase of the computational burden for DFGP. Hence, fixing th_s to 0.8 is justified by both the theoretical analysis and experimental results, and is a good tradeoff between the performance and the computational efficiency.

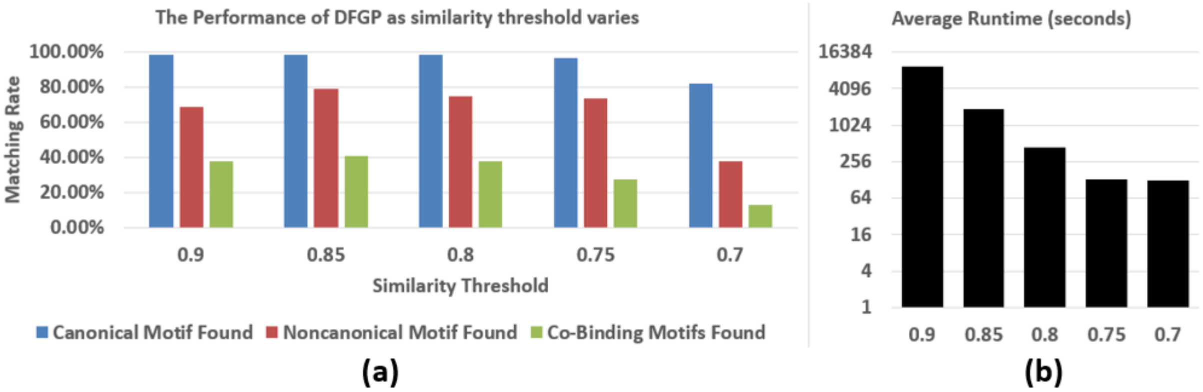


Figure 21.The performance of DFGP over 68 ChIP-seq datasets as th_s varies. (a) The percentage of the datasets in which the canonical motif, the noncanonical motifs and the co-binding motifs are identified respectively. (b) The average runtime. The y-axis is in log_2 scale.

6.5.5 Scalability of DFGP

In order to test the scalability of DFGP, it was run using the same parameter settings but was applied to 68 ChIP-seq datasets containing all peaks instead of the top ranked 500 peaks. The average dataset size is 2497165 bp and the standard deviation is 2055841 bp. Figure 22 shows the runtime of DFGP against the dataset size. It is approximately linear to the dataset size up to 6 million bp. The runtime of DFGP for those datasets of size beyond this number may be affected by the memory usage of DFGP. Though the space complexity of DFGP is linear in the input size, in handling such a great amount of data, the current implementation of the algorithm does not optimize memory usage and hence consumes all the 32GB memory available in the computer, causing unnecessarily frequent garbage collection. The average runtime is 10011 seconds (2.78 hours) with standard deviation 11438 seconds. In practice, however, 500 to 2000 top ranked peaks in a ChIP-seq dataset are good enough for motif discovery. Low quality peaks on the contrary might add more noise to the datasets and hinder the performance of motif discovery.

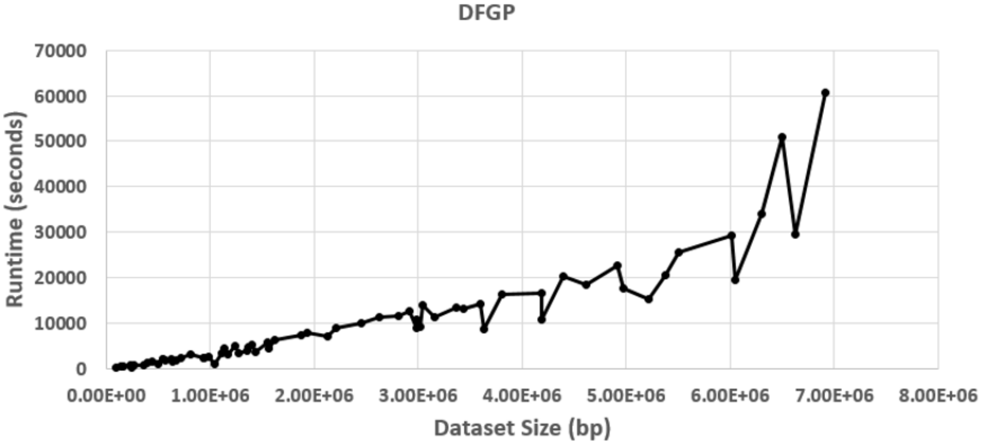


Figure 22. The runtime of DFGP for 68 ChIP-seq datasets with all peaks used.

Chapter 7
Conclusion and Future Research

7.1 Summary of Contribution

Non-coding DNA regions are enriched with motifs bound by proteins and ncRNAs. These binding motifs are parts of the enormously complex regulatory network at work in eukaryotic organisms. Much of the complexity in the mechanism of gene expression regulation is attributed to the cooperating biomolecules exerting combinatorial control. In combinatorial regulation, regulatory biomolecules co-bind with others in near distance to provide specific regulatory behaviors. The observed intriguing gene expression patterns in the developmental processes and cell differentiation result from such combinatorial regulation. To decipher the combinatorial regulatory mechanism, identifying co-binding motifs for cooperating biomolecules in the non-coding DNA sequences is a crucial task. Discovering these co-binding motifs not only reveals the combinatorial regulatory codes but also achieves better specificity in binding sites prediction while discovering simple motifs corresponding to individual biomolecules alone would produce many false positives that are unlikely to function in vivo.

This thesis addresses the problem of discovering co-binding motifs in genomic sequences. The combinatorial nature of cooperating biomolecules and their flexible interaction distance result in a unique challenging task of finding their corresponding co-binding motifs with component motifs separated by flexible gaps. Existing methods are ineffective for dealing with the following the difficulties: (1) relatively low enrichment of co-binding motifs in sequence datasets, (2) unknown lengths of component motifs and (3) potentially large distance between component motifs. The probabilistic approach is incapable of identifying co-binding motifs in realistic datasets as it is easily trapped into local optimal solutions during the search process. For deterministic approach using a relaxed motif template for generating flexible gap patterns, it faces the challenges of searching a huge pattern space and handling an enormous output due to its exhaustive enumeration nature and its lack of effective statistical significant measures for ranking discovered patterns.

This thesis develops an effective and scalable method, known as DFGP, for identifying co-binding motifs in large datasets. This provides a useful tool to facilitate biologists to uncover combinatorial regulatory codes. DFGP does not require the users to specify component motif lengths. In addition it allows large maximum distance between component motifs. It resolves the computational burden and the need of effective statistical significance measures faced by existing deterministic methods as follows:

(1) The concept of representative patterns is proposed and the method DRP is developed to extract these patterns. The use of the compact and non-redundant set of representative patterns instead of all consensus patterns by DFGP to construct flexible gap patterns significantly reduces the computational burden by excluding a large number of consensus pattern combinations.

(2) The concept of delegate occurrences is proposed for flexible gap pattern to reduce redundancy among pattern occurrences and the method DOA is developed to efficiently construct a flexible gap pattern with these delegate occurrences from its one box less prefix pattern and its last box, resulting in the runtime of DOA independent of the maximum distance parameter.

(3) The statistical significance measures are developed for ranking flexible gap patterns with complete occurrences and delegate occurrences respectively. The measures fill the need of effective pattern significance assessment methods for existing deterministic methods that use the classical but extremely slow Monte Carlo method.

Extensive experimental results show that (1) existing methods are ineffective for identifying co-binding motifs in large ChIP-seq datasets, (2) DFGP outperforms the state-of-the-art methods in co-binding motif discovery in terms of the capability of finding co-binding motifs and the speed, (3) DFGP achieves similar performance as one of the best methods MEME-Chip in finding simple motifs and rigid gapped motifs, (4) the co-binding motifs found by DFGP reveal interesting biological insights previously unknown, (5) the two proposed redundancy reduction methods drastically reduce the computational burden without sacrificing quality, (6) the proposed statistical significance measures are effective for ranking flexible gap patterns, and (7) DFGP is scalable to massive datasets.

7.2 Suggested Future Research

The following is a list of future research directions for extending the concepts and methods proposed in this thesis.

1. The use of two frequency thresholds

In this thesis, only one frequency threshold is used for obtaining both representative patterns and flexible gap patterns. The use of a smaller frequency threshold for representative patterns and a relative larger threshold for flexible gap patterns might improve DFGP on finding co-binding motifs whose component motifs are more degenerated and tackling small size datasets where the solid pattern corresponding to the component motifs might appear less than 5 times.

2. Applications of the statistical measure for existing deterministic methods

The Monte Carlo method used by existing deterministic methods for co-binding motif discovery is very time consuming. The application of the statistical measure can help deal with ranking a huge output size.

3. Extension of DFGP for protein sequences

The problem of protein sequence divergence in a protein family makes unlikely the existence of a consensus pattern for a functional unit in the sequence. The use of ambiguous codes for amino acids might help to obtain a set of patterns of ambiguous amino acids as boxes for building flexible gap patterns for extracting structural motifs.

4. Extension of DFGP for ncRNA sequences

DFGP might be able to directly apply for related ncRNA sequences to extract conserved secondary structures that correspond to their functional roles in interacting with biomolecular complexes such as polycomb repressive complex 2.

5. Extension of DFGP to support searching the reverse and complementary strand

Currently DFGP only supports searching the forward strand. The easy attempt includes transforming the entire input sequences into reverse and complementary sequences, and using both forward and reverse strands as the new input. Certainly, there could be a better approach to handle this issue.

6. Utilizing the gap distribution between two consecutive boxes in a flexible gap pattern

The gap distribution for a flexible gap might serve as an indication of the pattern quality.

Appendix A

Table S1 The ENCODE ChIP-seq datasets used in experiments

The 1st column is the name of a ChIP-seq dataset from ENCODE. The 2nd column is the transcription factor targeted in a ChIP-seq experiment. The 3rd column is the canonical TF binding motif for the targeted TF. The 4th column is the noncanonical TF binding motifs in a ChIP-seq dataset. The 5th column is the cell line for which a ChIP-seq experiment was performed. The 6th column is the dataset size in bp of the top ranked 500 peaks in a ChIP-seq dataset, which are the input to motif discovery methods. The canonical and noncanonical motifs are shown in the supplementary Table S2 and Figure S2 in the paper by Wang et al. [2]. These datasets can be downloaded from http://factorbook.org where the datasets are labeled as SPP or http://encodeproject.org/ENCODE/downloads.html under TFBS SPP-based Peaks.

TF ChIP-Seq Dataset	ChIP'ed TF	Canonical Motif	Noncanonical Motif	Cell Line	Dataset Size (500 top ranked peaks)
wgEncodeSydhTfbsK562CjunIfna6hStdAlnRep0	JUN	AP-1;v-JUN	GATA1	K562	121012
wgEncodeSydhTfbsK562CjunIfng30StdAlnRep0	JUN	AP-1;v-JUN	GATA1	K562	92301
wgEncodeSydhTfbsK562CjunIfng6hStdAlnRep0	JUN	AP-1;v-JUN	GATA1	K562	123102
wgEncodeSydhTfbsK562CjunStdAlnRep0	JUN	AP-1;v-JUN	GATA1	K562	93730
wgEncodeUchicagoTfbsK562EjunbControlAlnRep0	JUNB	AP-1;v-JUN	GATA1	K562	117524
wgEncodeHaibTfbsHepg2JundPcr1xAlnRep0	JUND	AP-1;v-JUN	FOXA	HepG2	110612
wgEncodeSydhTfbsGm12878JundStdAlnRep0	JUND	AP-1;v-JUN	PU.1	GM12878	126233
wgEncodeUchicagoTfbsK562EjundControlAlnRep0	JUND	AP-1;v-JUN	GATA1	K562	152911
wgEncodeHaibTfbsHepg2Fosl2Pcr1xAlnRep0	FOSL2	AP-1	FOXA;HNF4	HepG2	131339
wgEncodeSydhTfbsHelas3Ap2alphaStdAlnRep0	TFAP2A	AP-2	AP-1	HeLa-S3	146888
wgEncodeSydhTfbsHelas3Ap2gammaStdAlnRep0	TFAP2C	AP-2	AP-1	HeLa-S3	174089
wgEncodeSydhTfbsHepg2CebpbForsklnStdAlnRep0	CEBPB	CEBPB	FOXA	HepG2	105321
wgEncodeSydhTfbsHepg2CebpbIggrabAlnRep0	CEBPB	CEBPB	FOXA	HepG2	120603
wgEncodeHaibTfbsGm12878Atf3Pcr1xAlnRep0	ATF3	CREB;CREB-ext	NRF1	GM12878	66184
wgEncodeSydhTfbsK562Atf3StdAlnRep0	ATF3	CREB;CREB-	GABP	K562	103114

		ext			
wgEncodeSydhTfbsHelas3E2f4StdAlnRep0	E2F4	E2F4	NRF1	HeLa-S3	129519
wgEncodeSydhTfbsK562bE2f4UcdAlnRep0	E2F4	E2F4	NF-Y;NRF1	K562b	119628
wgEncodeHaibTfbsK562E2f6h50V0416102AlnRep0	E2F6	E2F4	MAX	K562	199449
wgEncodeSydhTfbsHelas3E2f6StdAlnRep0	E2F6	E2F4	MYC	HeLa-S3	141727
wgEncodeSydhTfbsK562bE2f6UcdAlnRep0	E2F6	E2F4	MAX;NF-Y	K562b	210434
wgEncodeHaibTfbsH1hescEgr1V0416102AlnRep0	EGR1	EGR1	GABP	H1-hESC	75748
wgEncodeHaibTfbsK562Egr1V0416101AlnRep0	EGR1	EGR1	AP-1	K562	102728
wgEncodeHaibTfbsGm12878Elf1sc631V0416101AlnRep0	ELF1	ELF1	YY1	GM12878	191190
wgEncodeSydhTfbsHepg2ErraForsklnStdAlnRep0	ESRRA	ESRRA	HNF4	HepG2	124882
wgEncodeHaibTfbsHepg2Foxa1sc101058Pcr1xAlnRep0	FOXA1	FOXA	HNF4	HepG2	168220
wgEncodeHaibTfbsHepg2Foxa2sc6554V0416101AlnRep0	FOXA2	FOXA	HNF4	HepG2	171879
wgEncodeHaibTfbsH1hescGabpPcr1xAlnRep0	GABPA	GABP	YY1	H1-hESC	110949
wgEncodeHaibTfbsK562GabpV0416101AlnRep0	GABPA	GABP	YY1	K562	184272
wgEncodeHaibTfbsHepg2Hnf4ah171Pcr1xAlnRep0	HNF4A	HNF4	CEBPB;FOXA	HepG2	148184
wgEncodeSydhTfbsHepg2Hnf4aForsklnStdAlnRep0	HNF4A	HNF4	FOXA	HepG2	87728
wgEncodeHaibTfbsHepg2Hnf4gsc6558V0416101AlnRep0	HNF4G	HNF4	CEBPB;FOXA	HepG2	111995
wgEncodeSydhTfbsGm12878MaxStdAlnRep0	MAX	MAX	PU.1;USF	GM12878	129298
wgEncodeSydhTfbsH1hescMaxUcdAlnRep0	MAX	MAX	CTCF	H1-hESC	84065
wgEncodeSydhTfbsHelas3MaxStdAlnRep0	MAX	MAX	USF	HeLa-S3	91536
wgEncodeSydhTfbsK562MaxStdAlnRep0	MAX	MAX	NRF1;USF	K562	78423
wgEncodeHaibTfbsGm12878Mef2aPcr1xAlnRep0	MEF2A	MEF2	PU.1	GM12878	146230
wgEncodeHaibTfbsGm12878Mef2csc13268Pcr1xAlnRep0	MEF2C	MEF2	AP-1;PU.1	GM12878	100707
wgEncodeOpenChromChipHepg2CmycAlnRep0	MYC	MYC	CTCF	HepG2	87985
wgEncodeOpenChromChipK562CmycAlnRep0	MYC	MYC	YY1	K562	63015
wgEncodeSydhTfbsK562CmycIfna30StdAlnRep0	MYC	MYC	CTCF	K562	132335
wgEncodeSydhTfbsK562CmycIfna6hStdAlnRep0	MYC	MYC	GABP	K562	112919
wgEncodeSydhTfbsK562NfybStdAlnRep0	NFYB	NF-Y	USF	K562	188325
wgEncodeSydhTfbsK562Nfe2StdAlnRep0	NFE2	NFE2	AP-1;USF	K562	77353
wgEncodeSydhTfbsGm12878NfkbIggrabAlnRep0	NFKB1	NFKB1	PU.1	GM12878	95177
wgEncodeSydhTfbsGm12891NfkbIggrabAlnRep0	NFKB1	NFKB1	PU.1	GM12891	94398
wgEncodeSydhTfbsHepg2bTr4UcdAlnRep0	NR2C2	NR2C2	GABP	HepG2b	123595
wgEncodeSydhTfbsK562bTr4UcdAlnRep0	NR2C2	NR2C2	GABP	K562b	134503
wgEncodeSydhTfbsHelas3Prdm1vIggrabAlnRep0	PRDM1	PRDM1	AP-1	HeLa-S3	92020
wgEncodeHaibTfbsHepg2RxraPcr1xAlnRep0	RXRA	RXRA	FOXA	HepG2	128422

wgEncodeHaibTfbsGm12878Sp1Pcr1xAlnRep0	SP1	SP1	UA2	GM12878	158381
wgEncodeHaibTfbsK562Sp1Pcr1xAlnRep0	SP1	SP1	NF-Y	K562	113929
wgEncodeHaibTfbsK562Sp2sc643V0416102AlnRep0	SP2	SP1	CTCF;NF-Y	K562	148009
wgEncodeSydhTfbsK562Stat1Ifna30StdAlnRep0	STAT1	STAT1	STAT2	K562	129336
wgEncodeSydhTfbsK562Stat2Ifna30StdAlnRep0	STAT2	STAT2	STAT1	K562	132253
wgEncodeHaibTfbsGm12878Tcf12Pcr1xAlnRep0	TCF12	TCF12	PU.1	GM12878	138708
wgEncodeHaibTfbsH1hescTcf12Pcr1xAlnRep0	TCF12	TCF12	SOX2-OCT4	H1-hESC	75950
wgEncodeSydhTfbsHepg2Tcf4UcdAlnRep0	TCF7L2	TCF7L2	FOXA;RXRA	HepG2b	286925
wgEncodeHaibTfbsGm12878Pbx3Pcr1xAlnRep0	PBX3	UA2	SP1	GM12878	133306
wgEncodeHaibTfbsK562Zbtb7asc34508V0416101AlnRep0	ZBTB7A	UA3	CTCF	K562	84944
wgEncodeHaibTfbsK562Thap1sc98174V0416101AlnRep0	THAP1	UA4;UA5	GABP;NRF1	K562	110833
wgEncodeHaibTfbsGm12878Usf1Pcr2xAlnRep0	USF1	USF	YY1	GM12878	112571
wgEncodeHaibTfbsK562Usf1V0416101AlnRep0	USF1	USF	NF-Y	K562	131121
wgEncodeSydhTfbsH1hescUsf2IggrabAlnRep0	USF2	USF	YY1	H1-hESC	110058
wgEncodeSydhTfbsK562Usf2StdAlnRep0	USF2	USF	NF-Y	K562	98022
wgEncodeHaibTfbsGm12878Yy1V0416101AlnRep0	YY1	YY1	GABP	GM12878	179740
wgEncodeHaibTfbsK562Yy1V0416101AlnRep0	YY1	YY1	GABP	K562	150546
wgEncodeHaibTfbsK562Yy1V0416102AlnRep0	YY1	YY1	GABP	K562	181777
wgEncodeSydhTfbsK562bYy1UcdAlnRep0	YY1	YY1	GABP;USF	K562b	146164

Appendix B

Table S2 The matching of discovered patterns by motif discovery methods to motifs in ChIP-seq datasets

This table shows the details of which top ranked pattern of a motif discovery method matches a motif in the 68 ChIP-seq datasets. The top ranked patterns are numbered from M1 to M30. The 1st column is the motif discovery method. The 2nd column is the TF ChIP-seq dataset. The 3rd, 4th and 5th columns show a top ranked pattern obtained by a method that matches the canonical motif, the noncanonical motifs and the co-binding motifs respectively in a ChIP-seq dataset if applicable.

Method	TF ChIP-seq Dataset	Canonical Motif	Noncanonical Motif	Co-Binding Motifs
	wgEncodeSydhTfbsK562CjunIfna6hStdAlnRep0	AP1;vJUN	GATA1	AP1-GATA1;vJUN-GATA1
DFGP		M2;NA	M3	M18;NA
RISOTTO		M3;NA		
Bioprospector		M1;NA		
GLAM2		M2;M1		
MEME-Chip		M1;M4	M7	
MEME		M1;M3		
MDmodule		M1;M2		
Weeder		M1;M8	M9	
	wgEncodeSydhTfbsK562CjunIfng30StdAlnRep0	AP1;vJUN	GATA1	AP1-GATA1;vJUN-GATA1
DFGP		M1;NA	M14	
RISOTTO		M1;NA		
Bioprospector		M1;NA		
GLAM2		M1;M2		
MEME-Chip		M1;M3	M5	
MEME		M1;M2		
MDmodule		M1;M2	M3	
Weeder		M1;NA	M8	
	wgEncodeSydhTfbsK562CjunIfng6hStdAlnRep0	AP1;vJUN	GATA1	AP1-GATA1;vJUN-GATA1
DFGP		M1;M14		
RISOTTO		M1;NA		
Bioprospector		M1;NA		
GLAM2		M1;M2		
MEME-Chip		M1;M4	M7	
MEME		M1;M3		
MDmodule		M1;M2		

Weeder		M1;NA	M8	
	wgEncodeSydhTfbsK562CjunStdAlnRep0	AP1;vJUN	GATA1	AP1-GATA1;vJUN-GATA1
DFGP		M2;M5	M15	
RISOTTO		M1;NA		
Bioprospector		M1;NA		
GLAM2		M2;M6		
MEME-Chip		M1;M3	M5	
MEME		M1;M2		
MDmodule		M1;NA		
Weeder		M1;M6		
	wgEncodeUchicagoTfbsK562EjunbControlAlnRep0	AP1;vJUN	GATA1	AP1-GATA1;vJUN-GATA1
DFGP		M1;NA	M11	M24;NA
RISOTTO		M1;NA		
Bioprospector		M1;NA		
GLAM2		M6;M2		
MEME-Chip		M1;NA	M4	
MEME		M1;NA		
MDmodule		M1;NA		
Weeder		M1;M6	M8	
	wgEncodeHaibTfbsHepg2JundPcr1xAlnRep0	AP1;vJUN	FOXA	AP1-FOXA;vJUN-FOXA
DFGP		M2;M11	M4	M7;M11
RISOTTO		M1;NA		
Bioprospector		M1;NA		
GLAM2		M2;M1		
MEME-Chip		M1;NA	M3	
MEME		M1;NA	M3	
MDmodule		M1;NA	M7	
Weeder		M1;NA		
	wgEncodeSydhTfbsGm12878JundStdAlnRep0	AP1;vJUN	PU1	AP1-PU1;vJUN-PU1
DFGP		M2;NA	M20	M20;NA
RISOTTO		M1;NA		
Bioprospector		M1;NA		
GLAM2		M4;M1	M27	
MEME-Chip		M1;NA	M11	
MEME		M1;NA		
MDmodule		M1;NA		
Weeder		M1;NA		
	wgEncodeUchicagoTfbsK562EjundControlAlnRep0	AP1;vJUN	GATA1	AP1-GATA1;vJUN-GATA1
DFGP		M1;M26	M19	

RISOTTO		M1;NA		
Bioprospector		M1;NA		
GLAM2		M1;M2		
MEME-Chip		M1;M2	M6	
MEME		M1;M3		
MDmodule		M1;M2		
Weeder		M1;M6	M9	
	wgEncodeHaibTfbsHepg2Fosl2Pcr1xAlnRep0	AP1	FOXA;HNF4	AP1-FOXA;AP1-HNF4
DFGP		M2	M18;NA	
RISOTTO		M1	NA;M20	NA;M20
Bioprospector		M1	M22;NA	M22;NA
GLAM2		M3		
MEME-Chip		M1	M4;NA	
MEME		M1	NA;M5	
MDmodule		M1		
Weeder		M1		
	wgEncodeSydhTfbsHelas3Ap2alphaStdAlnRep0	AP2	AP1	AP2-AP1
DFGP		M1	M10	
RISOTTO				
Bioprospector		M5	M3	
GLAM2		M1		
MEME-Chip		M1	M3	
MEME		M1	M2	
MDmodule		M2	M3	
Weeder		M1	M2	
	wgEncodeSydhTfbsHelas3Ap2gammaStdAlnRep0	AP2	AP1	AP2-AP1
DFGP		M1	M22	
RISOTTO				
Bioprospector		M8	M12	
GLAM2		M1		
MEME-Chip		M1	M2	
MEME		M1	M2	
MDmodule		M2	M1	
Weeder		M1	M2	
	wgEncodeSydhTfbsHepg2CebpbForsklnStdAlnRep0	CEBPB	FOXA	CEBPB-FOXA
DFGP		M1		
RISOTTO				
Bioprospector		M1	M16	M16
GLAM2		M1		

MEME-Chip		M1	M8	
MEME		M1	M3	
MDmodule		M1		
Weeder		M1		
	wgEncodeSydhTfbsHepg2CebpbIggrabAlnRep0	CEBPB	FOXA	CEBPB-FOXA
DFGP		M2		
RISOTTO				
Bioprospector		M1	M23	M23
GLAM2		M1		
MEME-Chip		M1		
MEME		M1		
MDmodule		M1		
Weeder		M1		
	wgEncodeHaibTfbsGm12878Atf3Pcr1xAlnRep0	CREB;CREBext	NRF1	CREB-NRF1;CREBext-NRF1
DFGP		NA;M4		
RISOTTO		NA;M2		
Bioprospector				
GLAM2		NA;M3		
MEME-Chip		M11;M2		
MEME		NA;M3		
MDmodule				
Weeder		NA;M11		
	wgEncodeSydhTfbsK562Atf3StdAlnRep0	CREB;CREBext	GABP	CREB-GABP;CREBext-GABP
DFGP		NA;M9		
RISOTTO		NA;M5		
Bioprospector				
GLAM2		NA;M1		
MEME-Chip		M3;M3	M11	
MEME		M3;M3		
MDmodule		M13;M5		
Weeder		NA;M15		
	wgEncodeSydhTfbsHelas3E2f4StdAlnRep0	E2F4	NRF1	E2F4-NRF1
DFGP		M4	M11	
RISOTTO				
Bioprospector		M13		
GLAM2		M3		
MEME-Chip		M1	M12	

MEME		M1	M4	
MDmodule		M9		
Weeder		M1		
	wgEncodeSydhTfbsK562bE2f4UcdAlnRep0	E2F4	NFY;NRF1	E2F4-NFY;E2F4-NRF1
DFGP		M5	M7;M6	
RISOTTO		M8	M27;NA	
Bioprospector		M12		
GLAM2		M1		
MEME-Chip		M1	M2;NA	
MEME		M1	M2;NA	
MDmodule		M4	M2;M5	
Weeder		M1		
	wgEncodeHaibTfbsK562E2f6h50V0416102AlnRep0	E2F4	MAX	E2F4-MAX
DFGP		M8		
RISOTTO		M7		
Bioprospector			M2	
GLAM2				
MEME-Chip		M1	M3	
MEME		M2		
MDmodule		M13	M2	
Weeder		M1		
	wgEncodeSydhTfbsHelas3E2f6StdAlnRep0	E2F4	MYC	E2F4-MYC
DFGP		M2		
RISOTTO		M24		
Bioprospector		M11		
GLAM2		M1		
MEME-Chip		M1	M7	
MEME		M1		
MDmodule		M2	M5	
Weeder		M1	M3	
	wgEncodeSydhTfbsK562bE2f6UcdAlnRep0	E2F4	MAX;NFY	E2F4-MAX;E2F4-NFY
DFGP		M5		
RISOTTO				
Bioprospector				
GLAM2				
MEME-Chip		M1	M2;M5	
MEME		M1		
MDmodule				
Weeder		M3		

	wgEncodeHaibTfbsH1hescEgr1V0416102AlnRep0	EGR1	GABP	EGR1-GABP
DFGP		M1	M3	
RISOTTO		M13	M28	
Bioprospector		M2		
GLAM2		M1		
MEME-Chip		M1		
MEME		M1		
MDmodule		M1		
Weeder		M1	M20	
	wgEncodeHaibTfbsK562Egr1V0416101AlnRep0	EGR1	AP1	EGR1-AP1
DFGP		M1		
RISOTTO		M2		
Bioprospector		M1		
GLAM2		M1		
MEME-Chip		M1		
MEME		M1	M5	
MDmodule		M1		
Weeder		M1		
	wgEncodeHaibTfbsGm12878Elf1sc631V0416101AlnRep0	ELF1	YY1	ELF1-YY1
DFGP		M4	M13	M13
RISOTTO				
Bioprospector		M4		
GLAM2		M1		
MEME-Chip		M1	M10	
MEME		M1		
MDmodule		M1	M4	
Weeder		M1		
	wgEncodeSydhTfbsHepg2ErraForsklnStdAlnRep0	ESRRA	HNF4	ESRRA-HNF4
DFGP		M1	M5	M30
RISOTTO		M1	M6	
Bioprospector		M1	M8	
GLAM2		M2	M4	
MEME-Chip		M1	M3	
MEME		M1	M2	
MDmodule		M1	M4	
Weeder		M1	M14	
	wgEncodeHaibTfbsHepg2Foxa1sc101058Pcr1xAlnRep0	FOXA	HNF4	FOXA-HNF4
DFGP		M5	M9	

RISOTTO		M12		
Bioprospector		M1		
GLAM2		M1		
MEME-Chip		M1		
MEME		M1		
MDmodule		M1	M3	
Weeder		M1		
	wgEncodeHaibTfbsHepg2Foxa2sc6554V0416101AlnRep0	FOXA	HNF4	FOXA-HNF4
DFGP		M4	M1	
RISOTTO				
Bioprospector		M4		
GLAM2		M1		
MEME-Chip		M1		
MEME		M1		
MDmodule		M1		
Weeder		M1	M21	
	wgEncodeHaibTfbsH1hescGabpPcr1xAlnRep0	GABP	YY1	GABP-YY1
DFGP		M2	M28	
RISOTTO		M1		
Bioprospector		M1	M7	
GLAM2		M2		
MEME-Chip		M1	M7	
MEME		M1	M4	
MDmodule		M1		
Weeder		M1		
	wgEncodeHaibTfbsK562GabpV0416101AlnRep0	GABP	YY1	GABP-YY1
DFGP		M1	M8	
RISOTTO		M1		
Bioprospector		M4		
GLAM2		M1		
MEME-Chip		M1		
MEME		M1	M5	
MDmodule		M1		
Weeder		M1		
	wgEncodeHaibTfbsHepg2Hnf4ah171Pcr1xAlnRep0	HNF4	CEBPB;FOXA	HNF4-CEBPB;HNF4-FOXA
DFGP		M5	NA;M15	NA;M26
RISOTTO		M9		

Bioprospector		M4		
GLAM2		M1		
MEME-Chip		M1	NA;M3	
MEME		M1		
MDmodule		M1		
Weeder		M1		
	wgEncodeSydhTfbsHepg2Hnf4aForsklnStdAlnRep0	HNF4	FOXA	HNF4-FOXA
DFGP		M1	M1	M1
RISOTTO		M1		
Bioprospector		M1		
GLAM2		M1		
MEME-Chip		M1	M8	
MEME		M1	M3	
MDmodule		M1		
Weeder		M1		
	wgEncodeHaibTfbsHepg2Hnf4gsc6558V0416101AlnRep0	HNF4	CEBPB;FOXA	HNF4-CEBPB;HNF4-FOXA
DFGP		M8	NA;M8	NA;M8
RISOTTO		M1		
Bioprospector		M1	NA;M21	NA;M26
GLAM2		M1		
MEME-Chip		M1	NA;M3	
MEME		M1	NA;M3	
MDmodule		M1		
Weeder		M1		
	wgEncodeSydhTfbsGm12878MaxStdAlnRep0	MAX	PU1;USF	MAX-PU1;MAX-USF
DFGP		M4	M21;M4	NA;M4
RISOTTO		M2	NA;M19	
Bioprospector				
GLAM2		M1		
MEME-Chip		M1	NA;M3	
MEME		M1	M3;NA	
MDmodule		M11	NA;M1	
Weeder		M1	NA;M4	
	wgEncodeSydhTfbsH1hescMaxUcdAlnRep0	MAX	CTCF	MAX-CTCF
DFGP		M1		
RISOTTO				
Bioprospector		M5		
GLAM2		M1		

MEME-Chip		M1	M4	
MEME		M1	M3	
MDmodule		M1		
Weeder		M1		
	wgEncodeSydhTfbsHelas3MaxStdAlnRep0	MAX	USF	MAX-USF
DFGP		M1	M8	M8
RISOTTO		M8	M4	
Bioprospector				
GLAM2		M2	M16	
MEME-Chip		M2		
MEME				
MDmodule		M2		
Weeder				
	wgEncodeSydhTfbsK562MaxStdAlnRep0	MAX	NRF1;USF	MAX-NRF1;MAX-USF
DFGP		M1	M16;M5	
RISOTTO		M2	NA;M1	
Bioprospector		M16		
GLAM2		M1	NA;M3	
MEME-Chip		M1	M11;M3	
MEME		M1	M5;M3	
MDmodule		M14	M4;M1	
Weeder		M1	NA;M4	
	wgEncodeHaibTfbsGm12878Mef2aPcr1xAlnRep0	MEF2	PU1	MEF2-PU1
DFGP		M7	M4	M9
RISOTTO		M11	M23	M23
Bioprospector		M10		
GLAM2		M1		
MEME-Chip		M1	M3	
MEME		M1	M2	
MDmodule		M1		
Weeder		M1		
	wgEncodeHaibTfbsGm12878Mef2csc13268Pcr1xAlnRep0	MEF2	AP1;PU1	MEF2-AP1;MEF2-PU1
DFGP		M3	M12;M2	M25;M29
RISOTTO		M2	NA;M4	
Bioprospector		M1		
GLAM2		M1		
MEME-Chip		M1	M4;M3	
MEME		M1	M4;NA	

MDmodule		M1		
Weeder		M1		
	wgEncodeOpenChromChipHepg2CmycAlnRep0	MYC	CTCF	MYC-CTCF
DFGP		M1	M5	
RISOTTO		M30		
Bioprospector		M14	M21	
GLAM2		M1		
MEME-Chip		M1	M13	
MEME		M1		
MDmodule		M11		
Weeder		M2		
	wgEncodeOpenChromChipK562CmycAlnRep0	MYC	YY1	MYC-YY1
DFGP		M1		
RISOTTO		M20	M17	
Bioprospector		M1		
GLAM2		M1		
MEME-Chip		M1	M5	
MEME		M1	M3	
MDmodule		M1		
Weeder		M1		
	wgEncodeSydhTfbsK562CmycIfna30StdAlnRep0	MYC	CTCF	MYC-CTCF
DFGP		M4		
RISOTTO		M25		
Bioprospector			M16	
GLAM2		M1		
MEME-Chip		M1	M7	
MEME		M1	M5	
MDmodule		M6		
Weeder		M2		
	wgEncodeSydhTfbsK562CmycIfna6hStdAlnRep0	MYC	GABP	MYC-GABP
DFGP		M4	M5	
RISOTTO		M8	M7	
Bioprospector		M10		
GLAM2		M1		
MEME-Chip		M1	M10	
MEME		M1	M4	
MDmodule		M6	M2	
Weeder		M2	M9	
	wgEncodeSydhTfbsK562Nfe2StdAlnRep0	NFE2	AP1;USF	NFE2-AP1;NFE2-USF

DFGP		M4	M11;M1	M11;M20
RISOTTO		M18	NA;M1	
Bioprospector		M4	M1;NA	
GLAM2		M3	M1;M23	M12;NA
MEME-Chip		M1	M6;M2	
MEME		M1	NA;M2	
MDmodule		M1	NA;M2	
Weeder		M1	NA;M6	
	wgEncodeSydhTfbsGm12878NfkbIggrabAlnRep0	NFKB1	PU1	NFKB1-PU1
DFGP		M3	M8	
RISOTTO		M1		
Bioprospector		M1		
GLAM2		M1	M24	
MEME-Chip		M1	M3	
MEME		M1		
MDmodule		M1		
Weeder		M1		
	wgEncodeSydhTfbsGm12891NfkbIggrabAlnRep0	NFKB1	PU1	NFKB1-PU1
DFGP		M1	M22	M22
RISOTTO		M2		
Bioprospector		M2		
GLAM2		M1		
MEME-Chip		M1	M8	
MEME		M1		
MDmodule		M1		
Weeder		M1	M11	
	wgEncodeSydhTfbsK562NfybStdAlnRep0	NFY	USF	NFY-USF
DFGP		M1		
RISOTTO		M1		
Bioprospector		M2		
GLAM2		M1		
MEME-Chip		M1	M7	
MEME		M1		
MDmodule		M1		
Weeder		M1		
	wgEncodeSydhTfbsHepg2bTr4UcdAlnRep0	NR2C2	GABP	NR2C2-GABP
DFGP		M8	M1	M8
RISOTTO				
Bioprospector		M4	M7	

GLAM2			M1	
MEME-Chip		M2	M1	
MEME		M3	M1	
MDmodule			M1	
Weeder		M14	M1	
	wgEncodeSydhTfbsK562bTr4UcdAlnRep0	NR2C2	GABP	NR2C2-GABP
DFGP		M13	M1	
RISOTTO			M1	
Bioprospector		M5		
GLAM2				
MEME-Chip		M2	M1	
MEME		M1	M2	
MDmodule		M7		
Weeder		M9	M1	
	wgEncodeSydhTfbsHelas3Prdm1vIggrabAlnRep0	PRDM1	AP1	PRDM1-AP1
DFGP		M1	M21	M21
RISOTTO		M1		
Bioprospector		M5	M3	
GLAM2		M1		
MEME-Chip		M1	M4	
MEME		M1	M2	
MDmodule		M1	M2	
Weeder		M1	M10	
	wgEncodeHaibTfbsHepg2RxraPcr1xAlnRep0	RXRA	FOXA	RXRA-FOXA
DFGP		M14	M10	
RISOTTO		M1		
Bioprospector		M4		
GLAM2		M1		
MEME-Chip		M1	M2	
MEME		M1	M2	
MDmodule			M4	
Weeder		M1	M9	
	wgEncodeHaibTfbsGm12878Sp1Pcr1xAlnRep0	SP1	UA2	SP1-UA2
DFGP		M2	M1	
RISOTTO				
Bioprospector		M4		
GLAM2		M1		
MEME-Chip		M4	M6	
MEME		M3		

MDmodule			M9	
Weeder				
	wgEncodeHaibTfbsK562Sp1Pcr1xAlnRep0	SP1	NFY	SP1-NFY
DFGP		M10	M1	M15
RISOTTO		M9	M1	M9
Bioprospector		M4	M1	
GLAM2		M1	M1	M1
MEME-Chip		M3	M1	
MEME		M2	M1	
MDmodule		M7	M1	
Weeder			M1	
	wgEncodeHaibTfbsK562Sp2sc643V0416102AlnRep0	SP1	CTCF;NFY	SP1-CTCF;SP1-NFY
DFGP		M3	M14;M3	M14;M3
RISOTTO			NA;M1	
Bioprospector		M3	NA;M16	
GLAM2		M2		
MEME-Chip		M8	NA;M1	
MEME		M4	NA;M1	
MDmodule		M8		
Weeder			NA;M3	
	wgEncodeSydhTfbsK562Stat1Ifna30StdAlnRep0	STAT1	STAT2	STAT1-STAT2
DFGP		M4	M2	M5
RISOTTO		M9	M1	M9
Bioprospector			M1	
GLAM2			M1	
MEME-Chip		M2	M1	
MEME		M2	M1	
MDmodule		M3	M1	
Weeder		M1	M2	
	wgEncodeSydhTfbsK562Stat2Ifna30StdAlnRep0	STAT2	STAT1	STAT2-STAT1
DFGP		M1	M3	M3
RISOTTO		M1	M7	M19
Bioprospector		M5		
GLAM2		M3		
MEME-Chip		M1	M2	
MEME		M1	M3	
MDmodule		M1		
Weeder		M1	M12	
	wgEncodeHaibTfbsGm12878Tcf12Pcr1xAlnRep0	TCF12	PU1	TCF12-PU1

DFGP		M4	M9	
RISOTTO		M1		
Bioprospector				
GLAM2		M1		
MEME-Chip		M1	M3	
MEME		M1	M2	
MDmodule		M1	M2	
Weeder		M1		
	wgEncodeHaibTfbsH1hescTcf12Pcr1xAlnRep0	TCF12	SOX2OCT4	TCF12-SOX2OCT4
DFGP		M6		
RISOTTO		M2		
Bioprospector		M4		
GLAM2		M8		
MEME-Chip		M7	M3	
MEME			M2	
MDmodule			M6	
Weeder			M13	
	wgEncodeSydhTfbsHepg2Tcf4UcdAlnRep0	TCF7L2	FOXA;RXRA	TCF7L2-FOXA;TCF7L2-RXRA
DFGP			M6;NA	
RISOTTO				
Bioprospector				
GLAM2		M1	NA;M10	
MEME-Chip		M1	M3;NA	
MEME		M1		
MDmodule		M1	NA;M9	
Weeder		M1	M9;NA	
	wgEncodeHaibTfbsGm12878Pbx3Pcr1xAlnRep0	UA2	SP1	UA2-SP1
DFGP		M2	M1	
RISOTTO				
Bioprospector		M11	M1	
GLAM2			M1	
MEME-Chip		M2	M4	
MEME		M4	M3	
MDmodule		M2	M9	
Weeder				
	wgEncodeHaibTfbsK562Zbtb7asc34508V0416101AlnRep0	UA3	CTCF	UA3-CTCF
DFGP		M2	M4	

RISOTTO		M5		
Bioprospector		M5		
GLAM2		M1		
MEME-Chip		M1	M2	
MEME		M5	M1	
MDmodule		M7	M11	
Weeder		M2	M1	
	wgEncodeHaibTfbsK562Thap1sc98174V0416101AlnRep0	UA4;UA5	GABP;NRF1	UA4-GABP;UA4-NRF1;UA5-GABP;UA5-NRF1
DFGP		M1;M4	M7;M14	M13;NA;M28;M18
RISOTTO		M16;M21	M21;NA	NA;NA;M21;NA
Bioprospector				
GLAM2		NA;M4		
MEME-Chip		M1;M2	M5;M3	NA;NA;M5;NA
MEME		M1;M2	NA;M4	
MDmodule		M1;NA	NA;M13	
Weeder		M4;M3	M13;NA	
	wgEncodeHaibTfbsGm12878Usf1Pcr2xAlnRep0	USF	YY1	USF-YY1
DFGP		M1		
RISOTTO		M2	M6	
Bioprospector		M3		
GLAM2		M1		
MEME-Chip		M1	M12	
MEME		M1	M3	
MDmodule		M1	M13	
Weeder		M1	M11	
	wgEncodeHaibTfbsK562Usf1V0416101AlnRep0	USF	NFY	USF-NFY
DFGP		M1	M18	M18
RISOTTO		M1		
Bioprospector		M1		
GLAM2		M1		
MEME-Chip		M1	M9	
MEME		M1		
MDmodule		M1		
Weeder		M1		
	wgEncodeSydhTfbsH1hescUsf2IggrabAlnRep0	USF	YY1	USF-YY1
DFGP		M1	M17	M17
RISOTTO		M1	M6	M6

Bioprospector				
GLAM2		M1		
MEME-Chip		M1	M7	
MEME		M1	M2	
MDmodule		M1	M11	
Weeder		M1		
	wgEncodeSydhTfbsK562Usf2StdAlnRep0	USF	NFY	USF-NFY
DFGP		M1		
RISOTTO		M2		
Bioprospector				
GLAM2		M1		
MEME-Chip		M1	M9	
MEME		M1		
MDmodule		M1		
Weeder		M1		
	wgEncodeHaibTfbsGm12878Yy1V0416101AlnRep0	YY1	GABP	YY1-GABP
DFGP		M1	M3	M3
RISOTTO		M1		
Bioprospector		M4		
GLAM2		M1		
MEME-Chip		M1	M6	
MEME		M1	M5	
MDmodule		M1		
Weeder		M1	M23	
	wgEncodeHaibTfbsK562Yy1V0416101AlnRep0	YY1	GABP	YY1-GABP
DFGP		M1	M21	
RISOTTO		M4		
Bioprospector		M1		
GLAM2		M1		
MEME-Chip		M1	M8	
MEME		M1		
MDmodule		M1		
Weeder		M1	M14	
	wgEncodeHaibTfbsK562Yy1V0416102AlnRep0	YY1	GABP	YY1-GABP
DFGP		M1		
RISOTTO		M1		
Bioprospector		M7		
GLAM2		M1		
MEME-Chip		M1	M6	

MEME		M1		
MDmodule		M1	M17	
Weeder		M1		
	wgEncodeSydhTfbsK562bYy1UcdAlnRep0	YY1	GABP;USF	YY1-GABP;YY1-USF
DFGP		M1	M7;M18	M16;M18
RISOTTO		M1		
Bioprospector		M7		
GLAM2		M1		
MEME-Chip		M1	M6;M5	
MEME		M1		
MDmodule		M1	M17;M4	
Weeder		M1	M14;NA	

Here is the summary for the above 68 datasets. The table below shows the percentage of the datasets where the canonical motif, the noncanonical motifs and the co-binding motifs are found by a method.

Method	Canonical Motif Found	Noncanonical Motif Found	Co-Binding Motifs Found
DFGP	98.52941	75	38.23529
RISOTTO	77.94118	29.41176	10.29412
Bioprospector	80.88235	26.47059	5.882353
GLAM2	91.17647	16.17647	2.941176
MEME-Chip	100	88.23529	1.470588
MEME	97.05882	55.88235	0
MDmodule	91.17647	44.11765	0
Weeder	91.17647	47.05882	0

Appendix C

Table S3 Similar Motifs among 79 motifs obtained by Wang et al. [2]

In the table, the 2nd column indicates motifs that are similar to the one in the 1st column.

Motif	Similar Motifs
A-Box	
AP1	NFE2;v-Maf
AP2	
B-Box	
BARHL2	
BHLHE40	MAX;MYC;USF
CEBPB	CREB
CREB	v-JUN;CEBPB
CREB-ext	
CTCF	
CTCF-ext	
E2F1	E2F4;UA3
E2F4	E2F1;EGR1
EBF1	
EGR1	SP1;E2F4
ELF1	ELK4;GABP;ETS1;UA5
ELK4	ELF1;GABP;ETS1;UA5
ESR1	ESRRA;RXRA;NR3C1
ESRRA	RXRA;ESR1;NR2C2
ETS1	GABP;ELK4;ELF1
FOXA	
GABP	ELF1;ELK4;ETS1;UA5
GATA1	GATA1-ext;TAL1;GATA3
GATA1-ext	GATA1;TAL1
GATA3	
GFI1	
HNF4	RXRA;NR2C2;ESRRA
HSF1	
MAX	MYC;USF;BHLHE40;ZEB1
MEF2	
MYC	MAX;BHLHE40;USF
NFE2	v-Maf;AP1
NFKB1	
NFY	NFY-UA2
NFY-UA2	UA2;NFY
NR2C2	RXRA;HNF4;ESRRA
NR3C1	ESR1
NRF1	

PAX5	
POU2F2	SOX2-OCT4
PRDM1	STAT2;PU1
PU1	PRDM1;ETS1
REST	
RFX5	
RUNX1	
RXRA	ESRRA;ESR1;HNF4;NR2C2;UA3
SOX2	SOX2-OCT4
SOX2-OCT4	POU2F2
SP1	EGR1;ZNF281
SREBF1	
SRF	
STAT1	ETS1
STAT2	PRDM1
TAL1	GATA1;GATA1-ext
TBP	
TCF12	ZEB1
TCF7L2	
TEAD1	
UA1	
UA10	
UA11	
UA12	
UA2	NFY-UA2
UA3	RXRA
UA4	YY1
UA5	GABP;ELF1;ELK4
UA6	
UA7	
UA8	
UA9	
USF	MAX;BHLHE40;MYC
YY1	UA4
ZEB1	TCF12
ZNF143	ZNF143-ext;UA9
ZNF143-ext	ZNF143
ZNF263	EGR1
ZNF281	SP1;EGR1
v-JUN	CREB;CEBPB
v-Maf	NFE2;AP1;FOXA

Appendix D

Table S4 Comparison of co-binding motifs proposed by DFGP and Wang et al. [2]

This table compares the enrichment of the co-binding motifs proposed by Wang et al. and those proposed by DFGP in their corresponding ChIP-seq datasets. The enrichment of a co-binding motif is the fraction of peaks in a ChIP-seq dataset where a TF motif pair (a co-binding motif) occurs. The number in a parenthesis after a motif pair indicates the fraction of peaks containing this pair. The 1st column is the ChIP-seq dataset. The 2nd column is the co-binding motifs proposed by Wang et al. The 3rd column is the co-binding motifs proposed by DFGP. The datasets can be grouped into 4 types based upon co-binding motifs proposed by Wang et al. and DFGP. The 4th column shows the type for a dataset.

TYPE 1 datasets are those where the co-binding motifs proposed by DFGP and Wang et al. have the best enrichment. TYPE 2 datasets are those where DFGP captures the co-binding motifs proposed by Wang et al. but has found other novel motifs that have better enrichment. TYPE 3 datasets are those where DFGP does not capture co-binding motifs proposed by Wang et al. but has found novel motifs that have better enrichment. TYPE 4 datasets are those where DFGP fails to capture co-binding motifs proposed by Wang et al. and the motifs proposed by DFGP have lower enrichment. Those motifs proposed by DFGP, bolded in the column 3, have better or equal enrichments compared with those by Wang et al. in 66% of 68 ChIP-seq datasets.

Dataset	Co-Binding Motifs by Wang et al.	Co-Binding Motifs by DFGP	TYPE
wgEncodeSydhTfbsK562CjunIfna6hStdAlnRep0	AP1-GATA1(0.12604)	**AP1-GATA1(0.12604)**	1
wgEncodeSydhTfbsK562CjunIfng30StdAlnRep0	AP1-GATA1(0.11274)		4
wgEncodeSydhTfbsK562CjunIfng6hStdAlnRep0	AP1-GATA1(0.13455)		4
wgEncodeSydhTfbsK562CjunStdAlnRep0	AP1-GATA1(0.075197)	**AP1-UA12(0.14375)**	3
wgEncodeUchicagoTfbsK562EjunbControlAlnRep0	AP1-GATA1(0.095682)	**AP1-GATA1(0.095682)**	1
wgEncodeHaibTfbsHepg2JundPcr1xAlnRep0	AP1-FOXA(0.065756)	**AP1-FOXA(0.065756)**	1
wgEncodeSydhTfbsGm12878JundStdAlnRep0	AP1-PU1(0.12012)	**AP1-STAT2(0.14344);AP1-PU1(0.12012)**	2
wgEncodeUchicagoTfbsK562EjundControlAlnRep0	AP1-GATA1(0.064083)	**AP1-TCF12(0.068922)**	3

wgEncodeHaibTfbsHepg2Fosl2Pcr1xAlnRep0	AP1-HNF4(0.099777);
	AP1-FOXA(0.078929)
			4

wgEncodeSydhTfbsHelas3Ap2alphaStdAlnRep0	AP2-AP1(0.18148)	**AP2-ZNF263(0.32565)**	3
wgEncodeSydhTfbsHelas3Ap2gammaStdAlnRep0	AP2-AP1(0.15088)	**AP2-SP1(0.38072)**	3
wgEncodeSydhTfbsHepg2CebpbForsklnStdAlnRep0	CEBPB-FOXA(0.055536)		4
wgEncodeSydhTfbsHepg2CebpbIggrabAlnRep0	CEBPB-FOXA(0.05543)		4

wgEncodeHaibTfbsGm12878Atf3Pcr1xAlnRep0	CREBext-NRF1(0.098726)		4
wgEncodeSydhTfbsK562Atf3StdAlnRep0	CREBext-GABP(0.12768)		4
wgEncodeSydhTfbsHelas3E2f4StdAlnRep0	E2F4-NRF1(0.41629)	**E2F4-SP1(0.64056)**	3

wgEncodeSydhTfbsK562bE2f4UcdAlnRep0	E2F4-NRF1(0.41116);
	E2F4-NFY(0.16844)
		E2F4-SP1(0.58268)	3

wgEncodeHaibTfbsK562E2f6h50V0416102AlnRep0	E2F4-MAX(0.073022)	**E2F4-UA5(0.13395)**	3
wgEncodeSydhTfbsHelas3E2f6StdAlnRep0	E2F4-MYC(0.25443)		4

wgEncodeSydhTfbsK562bE2f6UcdAlnRep0	E2F4-MAX(0.11922);
	E2F4-NFY(0.079512)
			4

wgEncodeHaibTfbsH1hescEgr1V0416102AlnRep0	EGR1-GABP(0.20465)		4

wgEncodeHaibTfbsK562Egr1V0416101AlnRep0	EGR1-AP1(0.068074)	**EGR1-ZNF281(0.59685);**
		EGR1-UA12(0.18081)
			3

wgEncodeHaibTfbsGm12878Elf1sc631V0416101AlnRep0	ELF1-YY1(0.069905)	**ELF1-ZNF263(0.17873);**
		ELF1-UA6(0.09621);ELF1-YY1(0.069905)
			2

wgEncodeSydhTfbsHepg2ErraForsklnStdAlnRep0	ESRRA-HNF4(0.37209)	**ESRRA-HNF4(0.37209)**	1
wgEncodeHaibTfbsHepg2Foxa1sc101058Pcr1xAlnRep0	FOXA-HNF4(0.10673)		4
wgEncodeHaibTfbsHepg2Foxa2sc6554V0416101AlnRep0	FOXA-HNF4(0.095183)		4
wgEncodeHaibTfbsH1hescGabpPcr1xAlnRep0	GABP-YY1(0.096735)	**GABP-SP1(0.32456);GABP-NRF1(0.16935)**	3
wgEncodeHaibTfbsK562GabpV0416101AlnRep0	GABP-YY1(0.088093)	**GABP-PU1(0.13195)**	3

wgEncodeHaibTfbsHepg2Hnf4ah171Pcr1xAlnRep0	HNF4-FOXA(0.097266);
	HNF4-CEBPB(0.043767)
		HNF4-FOXA(0.097266)	1

wgEncodeSydhTfbsHepg2Hnf4aForsklnStdAlnRep0	HNF4-FOXA(0.12373)	**HNF4-UA12(0.24385);HNF4-FOXA(0.12373)**	2

wgEncodeHaibTfbsHepg2Hnf4gsc6558V0416101AlnRep0	HNF4-FOXA(0.10344);
	HNF4-CEBPB(0.052682)
		HNF4-FOXA(0.10344)	1

wgEncodeSydhTfbsGm12878MaxStdAlnRep0	MAX-USF(0.46239);
	MAX-PU1(0.16148)
		MAX-USF(0.46239)	1

wgEncodeSydhTfbsH1hescMaxUcdAlnRep0	MAX-CTCF(0.18607)		4
wgEncodeSydhTfbsHelas3MaxStdAlnRep0	MAX-USF(0.18114)	**MAX-USF(0.18114)**	1

wgEncodeSydhTfbsK562MaxStdAlnRep0	MAX-USF(0.52444);
	MAX-NRF1(0.23282)
			4

wgEncodeHaibTfbsGm12878Mef2aPcr1xAlnRep0	MEF2-PU1(0.089433)	**MEF2-PU1(0.089433)**	1

wgEncodeHaibTfbsGm12878Mef2csc13268Pcr1xAlnRep0	MEF2-PU1(0.11819);
	MEF2-AP1(0.066254)
		MEF2-STAT2(0.13723);
		MEF2-PU1(0.11819);
		MEF2-AP1(0.066254)
			2

wgEncodeOpenChromChipHepg2CmycAlnRep0	MYC-CTCF(0.35156)	**MYC-SP1(0.36133)**	3
wgEncodeOpenChromChipK562CmycAlnRep0	MYC-YY1(0.026341)	**MYC-SP1(0.11919)**	3
wgEncodeSydhTfbsK562CmycIfna30StdAlnRep0	MYC-CTCF(0.21945)		4
wgEncodeSydhTfbsK562CmycIfna6hStdAlnRep0	MYC-GABP(0.090244)		4

wgEncodeSydhTfbsK562Nfe2StdAlnRep0	NFE2-AP1(0.59695);
	NFE2-USF(0.057579)
		NFE2-AP1(0.59695);NFE2-USF(0.057579)	1

wgEncodeSydhTfbsGm12878NfkbIggrabAlnRep0	NFKB1-PU1(0.15437)		4

wgEncodeSydhTfbsGm12891NfkbIggrabAlnRep0	NFKB1-PU1(0.13005)	**NFKB1-ETS1(0.15014);**
		NFKB1-EBF1(0.13524);
		NFKB1-PU1(0.13005)
			2

wgEncodeSydhTfbsK562NfybStdAlnRep0	NFY-USF(0.16206)	**NFY-SP1(0.49356);NFY-ZNF281(0.46438);**
		NFY-EGR1(0.4035);NFY-UA2(0.20269)
			3

wgEncodeSydhTfbsHepg2bTr4UcdAlnRep0	NR2C2-GABP(0.17384)	**NR2C2-GABP(0.17384)**	1
wgEncodeSydhTfbsK562bTr4UcdAlnRep0	NR2C2-GABP(0.20522)		4
wgEncodeSydhTfbsHelas3Prdm1vIggrabAlnRep0	PRDM1-AP1(0.20711)	**PRDM1-AP1(0.20711)**	1
wgEncodeHaibTfbsHepg2RxraPcr1xAlnRep0	RXRA-FOXA(0.091234)		4
wgEncodeHaibTfbsGm12878Sp1Pcr1xAlnRep0	SP1-UA2(0.14446)	**SP1-NFY(0.17505)**	3

wgEncodeHaibTfbsK562Sp1Pcr1xAlnRep0	SP1-NFY(0.30515)	**SP1-NFY(0.30515)**	1
wgEncodeHaibTfbsK562Sp2sc643V0416102AlnRep0	SP1-NFY(0.45124); SP1-CTCF(0.32298)	**SP1-NFY(0.45124);SP1-CTCF(0.32298)**	1
wgEncodeSydhTfbsK562Stat1Ifna30StdAlnRep0	STAT1-STAT2(0.16609)	**STAT1-STAT2(0.16609)**	1
wgEncodeSydhTfbsK562Stat2Ifna30StdAlnRep0	STAT2-STAT1(0.087684)	**STAT2-STAT1(0.087684)**	1
wgEncodeHaibTfbsGm12878Tcf12Pcr1xAlnRep0	TCF12-PU1(0.098749)		4
wgEncodeHaibTfbsH1hescTcf12Pcr1xAlnRep0	TCF12-SOX2OCT4(0.067414)		4
wgEncodeSydhTfbsHepg2Tcf4UcdAlnRep0	TCF7L2-RXRA(0.14831); TCF7L2-FOXA(0.12074)		4
wgEncodeHaibTfbsGm12878Pbx3Pcr1xAlnRep0	UA2-SP1(0.19661)		4
wgEncodeHaibTfbsK562Zbtb7asc34508V0416101AlnRep0	UA3-CTCF(0.43336)	**UA3-EGR1(0.5674);UA3-ZNF281(0.54297)**	3
wgEncodeHaibTfbsK562Thap1sc98174V0416101AlnRep0	UA5-GABP(0.20193); UA5-NRF1(0.13218)	**UA5-GABP(0.20193);UA5-NRF1(0.13218)**	1
wgEncodeHaibTfbsGm12878Usf1Pcr2xAlnRep0	USF-YY1(0.079729)	**USF-SP1(0.32358);USF-NRF1(0.14249); USF-ELF1(0.11366);USF-ELK4(0.11366)**	3
wgEncodeHaibTfbsK562Usf1V0416101AlnRep0	USF-NFY(0.13712)	**USF-ZNF281(0.26947);USF-SP1(0.20093); USF-EGR1(0.17719);USF-NFY(0.13712)**	2
wgEncodeSydhTfbsH1hescUsf2IggrabAlnRep0	USF-YY1(0.091205)	**USF-ZNF281(0.41745);USF-SP1(0.35299); USF-EGR1(0.32573);USF-ELK4(0.10252); USF-YY1(0.091205)**	2
wgEncodeSydhTfbsK562Usf2StdAlnRep0	USF-NFY(0.32493)	**USF-SP1(0.3857)**	3
wgEncodeHaibTfbsGm12878Yy1V0416101AlnRep0	YY1-GABP(0.060914)	**YY1-EGR1(0.14927);YY1-SP1(0.14402); YY1-UA5(0.076771);YY1-GABP(0.060914)**	2
wgEncodeHaibTfbsK562Yy1V0416101AlnRep0	YY1-GABP(0.091199)	**YY1-UA5(0.10861)**	3
wgEncodeHaibTfbsK562Yy1V0416102AlnRep0	YY1-GABP(0.075493)	**YY1-SP1(0.18887);YY1-UA5(0.092772)**	3
wgEncodeSydhTfbsK562bYy1UcdAlnRep0	YY1-GABP(0.20148)	**YY1-UA5(0.23472);YY1-GABP(0.20148)**	2

Bibliography

[1] Khurana, Ekta, Yao Fu, Vincenza Colonna, Xinmeng Jasmine Mu, Hyun Min Kang, Tuuli Lappalainen, Andrea Sboner et al. "Integrative annotation of variants from 1092 humans: Application to cancer genomics." Science 342, no. 6154 (2013): 1235587.

[2] Wang, Jie, Jiali Zhuang, Sowmya Iyer, XinYing Lin, Troy W. Whitfield, Melissa C. Greven, Brian G. Pierce et al. "Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors." Genome research 22, no. 9 (2012): 1798-1812.

[3] Khalil, Ahmad M., Mitchell Guttman, Maite Huarte, Manuel Garber, Arjun Raj, Dianali Rivea Morales, Kelly Thomas et al. "Many human large intergenic noncoding RNAs associate with chromatin-modifying complexes and affect gene expression." Proceedings of the National Academy of Sciences 106, no. 28 (2009): 11667-11672.

[4] Kunarso, Galih, Na-Yu Chia, Justin Jeyakani, Catalina Hwang, Xinyi Lu, Yun-Shen Chan, Huck-Hui Ng, and Guillaume Bourque. "Transposable elements have rewired the core regulatory network of human embryonic stem cells." Nature genetics 42, no. 7 (2010): 631-634.

[5] Chu, Ci, Kun Qu, Franklin L. Zhong, Steven E. Artandi, and Howard Y. Chang. "Genomic maps of long noncoding RNA occupancy reveal principles of RNA-chromatin interactions." Molecular cell 44, no. 4 (2011): 667-678.

[6] Elnitski, Laura, Victor X. Jin, Peggy J. Farnham, and Steven JM Jones. "Locating mammalian transcription factor binding sites: a survey of computational and experimental techniques." Genome research 16, no. 12 (2006): 1455-1464.

[7] Stewart, Alexander J., Sridhar Hannenhalli, and Joshua B. Plotkin. "Why transcription factor binding sites are ten nucleotides long." Genetics 192, no. 3 (2012): 973-985.

[8] Das, Modan K., and Ho-Kwok Dai. "A survey of DNA motif finding algorithms." BMC bioinformatics 8, no. Suppl 7 (2007): S21.

[9] Tompa, Martin, Nan Li, Timothy L. Bailey, George M. Church, Bart De Moor, Eleazar Eskin, Alexander V. Favorov et al. "Assessing computational tools for the discovery of transcription factor binding sites." Nature biotechnology 23, no. 1 (2005): 137-144.

[10] Bailey, Timothy L., and Charles Elkan. "Unsupervised learning of multiple motifs in biopolymers using expectation maximization." Machine learning 21, no. 1-2 (1995): 51-80.

[11] Wei, Chia-Lin, Qiang Wu, Vinsensius B. Vega, Kuo Ping Chiu, Patrick Ng, Tao Zhang, Atif Shahab et al. "A global map of p53 transcription-factor binding sites in the human genome." Cell 124, no. 1 (2006): 207-219.

[12] Wasserman, Wyeth W., and Albin Sandelin. "Applied bioinformatics for the identification of regulatory elements." Nature Reviews Genetics 5, no. 4 (2004): 276-287.

[13] Whitfield, Troy W., Jie Wang, Patrick J. Collins, E. Christopher Partridge, Shelley Force Aldred, Nathan D. Trinklein, Richard M. Myers, and Zhiping Weng. "Functional analysis of transcription factor binding sites in human promoters." Genome Biol 13, no. 9 (2012): R50.

[14] Kamvysselis, Manolis Kellis. "Computational comparative genomics: genes, regulation, evolution." PhD diss., Massachusetts Institute of Technology, 2003.

[15] Balaji, S., M. Madan Babu, Lakshminarayan M. Iyer, Nicholas M. Luscombe, and L. Aravind. "Comprehensive analysis of combinatorial regulation using the transcriptional regulatory network of yeast." Journal of molecular biology 360, no. 1 (2006): 213-227.

[16] Ravasi, Timothy, Harukazu Suzuki, Carlo Vittorio Cannistraci, Shintaro Katayama, Vladimir B. Bajic, Kai Tan, Altuna Akalin et al. "An atlas of combinatorial transcriptional regulation in mouse and man." Cell 140, no. 5 (2010): 744-752.

[17] Wang, Kevin C., and Howard Y. Chang. "Molecular mechanisms of long noncoding RNAs." Molecular cell 43, no. 6 (2011): 904-914.

[18] Klepper, Kjetil, Geir K. Sandve, Osman Abul, Jostein Johansen, and Finn Drablos. "Assessment of composite motif discovery methods." BMC bioinformatics 9, no. 1 (2008): 123.

[19] Sandve, Geir Kjetil, and Finn Drablos. "A survey of motif discovery methods in an integrated framework." Biol Direct 1, no. 11 (2006).

[20] Nguyen, Tung T., and Ioannis P. Androulakis. "Recent advances in the computational discovery of transcription factor binding sites." Algorithms 2, no. 1 (2009): 582-605.

[21] Kato, Mamoru, Naoya Hata, Nilanjana Banerjee, Bruce Futcher, and Michael Q. Zhang. "Identifying combinatorial regulation of transcription factors and binding motifs." Genome biology 5, no. 8 (2004): R56.

[22] MacIsaac, Kenzie D., and Ernest Fraenkel. "Practical strategies for discovering regulatory DNA sequence motifs." PLoS computational biology 2, no. 4 (2006): e36.

[23] Hannenhalli, Sridhar. "Eukaryotic transcription factor binding sites—modeling and integrative search methods." Bioinformatics 24, no. 11 (2008): 1325-1331.

[24] Pavesi, Giulio, Giancarlo Mauri, and Graziano Pesole. "Methods for pattern discovery in unaligned biological sequences." Briefings in Bioinformatics 2, no. 4 (2001): 417.

[25] Stormo, Gary D. "DNA binding sites: representation and discovery." Bioinformatics 16, no. 1 (2000): 16-23.

[26] Pavesi, Giulio, Giancarlo Mauri, and Graziano Pesole. "In silico representation and discovery of transcription factor binding sites." Briefings in Bioinformatics 5, no. 3 (2004): 217-236.

[27] GuhaThakurta, Debraj. "Computational identification of transcriptional regulatory elements in DNA sequence." Nucleic acids research 34, no. 12 (2006): 3585-3598.

[28] Jaspar Database at http://jaspar.genereg.net/

[29] Bailey, T. L. "Likelihood vs. information in aligning biopolymer sequences." USCD Technique Report CS93-318, (1993).

[30] D'haeseleer, Patrik. "How does DNA sequence motif discovery work?." Nature biotechnology 24, no. 8 (2006): 959-961.

[31] Lawrence, Charles E., and Andrew A. Reilly. "An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences." Proteins: Structure, Function, and Bioinformatics 7, no. 1 (1990): 41-51.

[32] Lawrence, Charles E., Stephen F. Altschul, Mark S. Boguski, Jun S. Liu, Andrew F. Neuwald, and John C. Wootton. "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment." science 262, no. 5131 (1993): 208-214.

[33] Ao, Wanyuan, Jeb Gaudet, W. James Kent, Srikanth Muttumu, and Susan E. Mango. "Environmentally induced foregut remodeling by PHA-4/FoxA and DAF-12/NHR." Science 305, no. 5691 (2004): 1743-1746.

[34] Liu, Jun S., Andrew F. Neuwald, and Charles E. Lawrence. "Bayesian models for multiple local sequence alignment and Gibbs sampling strategies." Journal of the American Statistical Association 90, no. 432 (1995): 1156-1170.

[35] Thompson, William, Eric C. Rouchka, and Charles E. Lawrence. "Gibbs Recursive Sampler: finding transcription factor binding sites." Nucleic acids research 31, no. 13 (2003): 3580-3585.

[36] Thompson, William A., Lee A. Newberg, Sean Conlan, Lee Ann McCue, and Charles E. Lawrence. "The Gibbs centroid sampler." Nucleic acids research 35, no. suppl 2 (2007): W232-W237.

[37] Workman, C. T., and G. D. Stormo. "ANN-Spec: a method for discovering transcription factor binding sites with improved specificity." In Pac Symp Biocomput, vol. 5, pp. 464-475. 2000.

[38] Frith, Martin C., Ulla Hansen, John L. Spouge, and Zhiping Weng. "Finding functional sequence elements by multiple local alignment." Nucleic acids research 32, no. 1 (2004): 189-200.

[39] Kim, Nak-Kyeong, Kannan Tharakaraman, Leonardo Mariño-Ramírez, and John L. Spouge. "Finding sequence motifs with Bayesian models incorporating positional information: an application to transcription factor binding sites." BMC bioinformatics 9, no. 1 (2008): 262.

[40] Favorov, Alexander V., Mikhail S. Gelfand, Anna V. Gerasimova, Dmitry A. Ravcheev, Andrey A. Mironov, and V. J. Makeev. "A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length." Bioinformatics 21, no. 10 (2005): 2240-2245.

[41] Apostolico, Alberto, Mary Ellen Bock, Stefano Lonardi, and Xuyan Xu. "Efficient detection of unusual words." Journal of Computational Biology 7, no. 1-2 (2000): 71-94.

[42] Gusfield, Dan. Algorithms on strings, trees and sequences: computer science and computational biology. Cambridge University Press, 1997.

[43] Parida, Laxmi. Pattern discovery in bioinformatics: theory & algorithms. CRC Press, 2007.

[44] Pevzner, Pavel A., and Sing-Hoi Sze. "Combinatorial approaches to finding subtle signals in DNA sequences." In ISMB, vol. 8, pp. 269-278. 2000.

[45] Sagot, Marie-France. "Spelling approximate repeated or common motifs using a suffix tree." In LATIN'98: Theoretical Informatics, pp. 374-390. Springer Berlin Heidelberg, 1998.

[46] Aluru, Srinivas, ed. "Lookup Tables, Suffix Trees and Suffix Arrays." Handbook of computational molecular biology. CRC Press, 2005.

[47] Pavesi, Giulio, Giancarlo Mauri, and Graziano Pesole. "An algorithm for finding signals of unknown length in DNA sequences." Bioinformatics 17, no. suppl 1 (2001): S207-S214.

[48] Rigoutsos, Isidore, and Aris Floratos. "Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm." Bioinformatics 14, no. 1 (1998): 55-67.

[49] Grossi, Roberto, Andrea Pietracaprina, Nadia Pisanti, Geppino Pucci, Eli Upfal, and Fabio Vandin. "MADMX: A strategy for maximal dense motif extraction." Journal of Computational Biology 18, no. 4 (2011): 535-545.

[50] Califano, Andrea. "SPLASH: structural pattern localization analysis by sequential histograms." Bioinformatics 16, no. 4 (2000): 341-357.

[51] Parida, Laxmi, Isidore Rigoutsos, Aris Floratos, Dan Platt, and Yuan Gao. "Pattern discovery on character sets and real-valued data: linear bound on irredundant motifs and an efficient polynomial time algorithm." In Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms, pp. 297-308. Society for Industrial and Applied Mathematics, 2000.

[52] Pisanti, Nadia, Maxime Crochemore, Roberto Grossi, and Marie-France Sagot. "Bases of motifs for generating repeated patterns with wild cards." Computational Biology and Bioinformatics, IEEE/ACM Transactions on 2, no. 1 (2005): 40-50.

[53] Pisanti, Nadia, Maxime Crochemore, Roberto Grossi, and Marie-France Sagot. "A comparative study of bases for motif inference." String Algorithmics (2005): 195-225.

[54] Apostolico, Alberto, and Claudia Tagliacollo. "Optimal extraction of irredundant motif bases." International Journal of Foundations of Computer Science 21, no. 06 (2010): 1035-1047.

[55] Apostolico, Alberto, Matteo Comin, and Laxmi Parida. "VARUN: discovering extensible motifs under saturation constraints." IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 7, no. 4 (2010): 752-762.

[56] Comin, Matteo, and Laxmi Parida. "Subtle Motif Discovery for Detection of DNA Regulatory Sites." In APBC, pp. 27-36. 2007.

[57] Van Helden, Jacques, Alma F. Rios, and Julio Collado-Vides. "Discovering regulatory elements in non-coding sequences by analysis of spaced dyads." Nucleic acids research 28, no. 8 (2000): 1808-1818.

[58] Sinha, Saurabh, and Martin Tompa. "Discovery of novel transcription factor binding sites by statistical overrepresentation." Nucleic acids research 30, no. 24 (2002): 5549-5560.

[59] Bailey, Timothy L. "DREME: motif discovery in transcription factor ChIP-seq data." Bioinformatics 27, no. 12 (2011): 1653-1659.

[60] Sharov, Alexei A., and Minoru SH Ko. "Exhaustive search for over-represented DNA sequence motifs with CisFinder." DNA research 16, no. 5 (2009): 261-273.

[61] Machanick, Philip, and Timothy L. Bailey. "MEME-ChIP: motif analysis of large DNA datasets." Bioinformatics 27, no. 12 (2011): 1696-1697.

[62] Liu, X. Shirley, Douglas L. Brutlag, and Jun S. Liu. "An algorithm for finding protein–DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments." Nature biotechnology 20, no. 8 (2002): 835-839.

[63] Conlon, Erin M., X. Shirley Liu, Jason D. Lieb, and Jun S. Liu. "Integrating regulatory motif discovery and genome-wide expression analysis." Proceedings of the National Academy of Sciences 100, no. 6 (2003): 3339-3344.

[64] Liu, Xiaole, Douglas L. Brutlag, and Jun S. Liu. "BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes." In Pacific symposium on biocomputing, vol. 6, no. 2001, pp. 127-138. 2001.

[65] GuhaThakurta, Debraj, and Gary D. Stormo. "Identifying target sites for cooperatively binding factors." Bioinformatics 17, no. 7 (2001): 608-621.

[66] Bi, Chengpeng, and Peter K. Rogan. "Bipartite pattern discovery by entropy minimization-based multiple local alignment." Nucleic acids research 32, no. 17 (2004): 4979-4991.

[67] Frith, Martin C., Neil FW Saunders, Bostjan Kobe, and Timothy L. Bailey. "Discovering sequence motifs with arbitrary insertions and deletions." PLoS computational biology 4, no. 5 (2008): e1000071.

[68] Marsan, Laurent, and Marie-France Sagot. "Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification." Journal of Computational Biology 7, no. 3-4 (2000): 345-362.

[69] Carvalho, Alexandra M., Ana T. Freitas, Arlindo L. Oliveira, and Marie-France Sagot. "An efficient algorithm for the identification of structured motifs in DNA promoter sequences." Computational Biology and Bioinformatics, IEEE/ACM Transactions on 3, no. 2 (2006): 126-140.

[70] Pisanti, Nadia, Alexandra M. Carvalho, Laurent Marsan, and Marie-France Sagot. "RISOTTO: Fast extraction of motifs with mismatches." In LATIN 2006: Theoretical Informatics, pp. 757-768. Springer Berlin Heidelberg, 2006.

[71] Zhang, Yongqiang, and Mohammed J. Zaki. "EXMOTIF: efficient structured motif extraction." Algorithms for Molecular Biology 1, no. 1 (2006): 21.

[72] Eskin, Eleazar, and Pavel A. Pevzner. "Finding composite regulatory patterns in DNA sequences." Bioinformatics 18, no. suppl 1 (2002): S354-S363.

[73] Hu, Yuh-Jyh. "Finding subtle motifs with variable gaps in unaligned DNA sequences." Computer methods and programs in biomedicine 70, no. 1 (2003): 11-20.

[74] Fassetti, Fabio, Gianluigi Greco, and Giorgio Terracina. "Mining loosely structured motifs from biological data." Knowledge and Data Engineering, IEEE Transactions on 20, no. 11 (2008): 1472-1489.

[75] Apostolico, Alberto, Cinzia Pizzi, and Esko Ukkonen. "Efficient algorithms for the discovery of gapped factors." Algorithms for Molecular Biology 6, no. 1 (2011).

[76] Wong, Andrew KC, Dennis Zhuang, Gary CL Li, and E-SA Lee. "Discovery of delta closed patterns and noninduced patterns from sequences." Knowledge and Data Engineering, IEEE Transactions on 24, no. 8 (2012): 1408-1421.

[77] Smit, A.F.A., R. Hubley, and P. Green. RepeatMasker at http://repeatmasker.org

[78] Bergroth, Lasse, Harri Hakonen, and Timo Raita. "A survey of longest common subsequence algorithms." In String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on, pp. 39-48. IEEE, 2000.

[79] ENCODE Project Consortium. "An integrated encyclopedia of DNA elements in the human genome." Nature 489, no. 7414 (2012): 57-74.

[80] Gerstein, Mark B., Anshul Kundaje, Manoj Hariharan, Stephen G. Landt, Koon-Kiu Yan, Chao Cheng, Xinmeng Jasmine Mu et al. "Architecture of the human regulatory network derived from ENCODE data." Nature 489, no. 7414 (2012): 91-100.

[81] Kharchenko, Peter V., Michael Y. Tolstorukov, and Peter J. Park. "Design and analysis of ChIP-seq experiments for DNA-binding proteins." Nature biotechnology 26, no. 12 (2008): 1351-1359.

[82] Grant, Charles E., Timothy L. Bailey, and William Stafford Noble. "FIMO: scanning for occurrences of a given motif." Bioinformatics 27, no. 7 (2011): 1017-1018.

[83] Gupta, Shobhit, John A. Stamatoyannopoulos, Timothy L. Bailey, and William Stafford Noble. "Quantifying similarity between motifs." Genome Biol 8, no. 2 (2007): R24.

[84] Das, Debopriya, Nilanjana Banerjee, and Michael Q. Zhang. "Interacting models of cooperative gene regulation." Proceedings of the National Academy of Sciences of the United States of America 101, no. 46 (2004): 16234-16239.

[85] Hannenhalli, Sridhar, and Samuel Levy. "Predicting transcription factor synergism." Nucleic acids research 30, no. 19 (2002): 4278-4284.

[86] Ha, Nati, Maria Polychronidou, and Ingrid Lohmann. "COPS: detecting co-occurrence and spatial arrangement of transcription factor binding motifs in genome-wide datasets." PloS one 7, no. 12 (2012): e52055.

[87] Segal, Eran, R. Yelensky, and Daphne Koller. "Genome-wide discovery of transcriptional modules from DNA sequence and gene expression." Bioinformatics 19, no. suppl 1 (2003): i273-i282.

[88] Morgante, Michele, Alberto Policriti, Nicola Vitacolonna, and Andrea Zuccolo. "Structured motifs search." Journal of Computational Biology 12, no. 8 (2005): 1065-1082.

[89] Zhou, Qing, and Wing H. Wong. "CisModule: de novo discovery of cis-regulatory modules by hierarchical mixture modeling." Proceedings of the national academy of sciences of the United States of America 101, no. 33 (2004): 12114-12119.

[90] Hertz, Gerald Z., and Gary D. Stormo. "Identifying DNA and protein patterns with statistically significant alignments of multiple sequences." Bioinformatics 15, no. 7 (1999): 563-577.

[91] Zhuang, Dennis, Gary Li, and Andrew Wong. "Discovery of Temporal Associations in Multivariate Time Series," IEEE Transactions on Knowledge and Data Engineering, (2014): http://doi.ieeecomputersociety.org/10.1109/TKDE.2014.2310219

[92] Levine, Michael, and Robert Tjian. "Transcription regulation and animal diversity." Nature 424, no. 6945 (2003): 147-151.

[93] Lemon, Bryan, and Robert Tjian. "Orchestrated response: a symphony of transcription factors for gene control." Genes & development 14, no. 20 (2000): 2551-2569.

[94] Reményi, Attila, H. R. Scholer, and Matthias Wilmanns. "Combinatorial control of gene expression." Nature structural & molecular biology 11, no. 9 (2004): 812-815.

[95] Maston, Glenn A., Sara K. Evans, and Michael R. Green. "Transcriptional regulatory elements in the human genome." Annu. Rev. Genomics Hum. Genet. 7 (2006): 29-59.

[96] Pauli, Andrea, John L. Rinn, and Alexander F. Schier. "Non-coding RNAs as regulators of embryogenesis." Nature Reviews Genetics 12, no. 2 (2011): 136-149.

[97] Amaral, Paulo P., and John S. Mattick. "Noncoding RNA in development." Mammalian genome 19, no. 7-8 (2008): 454-492.

[98] Guttman, Mitchell, and John L. Rinn. "Modular regulatory principles of large non-coding RNAs." Nature 482, no. 7385 (2012): 339-346.

[99] Smallwood, Sébastien A., and Gavin Kelsey. De novo DNA methylation: a germ cell perspective." Trends in Genetics 28, no. 1 (2012): 33-42.

[100] Deaton, Aimée M., and Adrian Bird. "CpG islands and the regulation of transcription." Genes & development 25, no. 10 (2011): 1010-1022.

[101] Shen, Lanlan, Yutaka Kondo, Yi Guo, Jiexin Zhang, Li Zhang, Saira Ahmed, Jingmin Shu, Xinli Chen, Robert A. Waterland, and Jean-Pierre J. Issa. "Genome-wide profiling of DNA methylation reveals a class of normally methylated CpG island promoters." PLoS genetics 3, no. 10 (2007): e181

[102] Sparmann, Anke, and Maarten van Lohuizen. "Polycomb silencers control cell fate, development and cancer." Nature Reviews Cancer 6, no. 11 (2006): 846-856.

[103] Feltus, F. Alex, Eva K. Lee, Joseph F. Costello, Christoph Plass, and Paula M. Vertino. "DNA motifs associated with aberrant CpG island methylation."Genomics 87, no. 5 (2006): 572-579.

[104] Bock, Christoph, Martina Paulsen, Sascha Tierling, Thomas Mikeska, Thomas Lengauer, and Jörn Walter. "CpG island methylation in human lymphocytes is highly correlated with DNA sequence, repeats, and predicted DNA structure."PLoS genetics 2, no. 3 (2006): e26.

[105] Song, Lingyun, Zhancheng Zhang, Linda L. Grasfeder, Alan P. Boyle, Paul G. Giresi, Bum-Kyu Lee, Nathan C. Sheffield et al. "Open chromatin defined by DNaseI and FAIRE identifies regulatory elements that shape cell-type identity."Genome research 21, no. 10 (2011): 1757-1767