Università degli Studi di Verona

DIPARTIMENTO DI INFORMATICA
Corso di Laurea Magistrale in Computer Science

Yearly report

blabla

Studente:
Manuel Tognon
Matricola VR456869

Relatore:
Prof. Rosalba Giugno

Correlatore:
Prof. Luca Pinello

Anno Accademico 2021–2022

Contents

List of Figures

List of Tables

Introduction

Transcription Factors

Transcription Factors (TFs) (**Fig.2.1**) are fundamental regulatory proteins playing a key role in regulating the transcriptional state, differentation and developmental state of cells (**???**). TFs present a modular structure, divide in three domains (**?**): (i) the DNA binding domain, (ii) the activation domain, and (iii) the signal sensing domain. The DNA binding domain guides the TF to the its target sites on the genome. The activation domain interacts with other transcriptional reagulators. The signal sensing domain captures external signals, transmitting them to the whole transcriptional complex. In human, about 1600 proteins are currently thought to function as TFs (**?**). Therefore, approximately the 8% of human genes encode for TFs. Transcription Factors often exploit their regulatory function on genes, by acting in a coordinate fashion. Moreover, TFs regulate multiple genes in different cell-types (**?**). TFs perform their regulatory function by employing different strategies: (i) helping or blocking RNA polymerase binding (**?**), (ii) weakining the DNA-histones associations opening the chromatin, (iii) catalyzing histones deacetylation (**?**), or (iv) strengthening the DNA-histone complexes closing the chromatin. By binding short DNA sequences (7-20 nucleotides (**?**)) called Transcription Factor binding sites (TFBS), TFs finely regulate gene expression in a cell-specific manner. TFBS are located within gene promoters (**?**) or in distal regulatory elements, such as enhancers, silencers or insulators (**???**). Although TFBS show recurring sequence patterns, which are often are referred to as *motifs*, TFs bind similar but not identical sequences, that can differ in a few nucleotides. The precise identity and configuration of TFBS, together with the conformation of flanking chromatin regions, critically regulate TFs regulatory function of cells (**??**). During DNA binding, Transcritpion Factors use a combination of electrostatic and Van der Waals forces. Although TFs bind their target sequences with high specificity, not all nucleotides in the binding sites directly interact with the TF. Since some interactions between the TF and the TFBS nucleotides are weaker than others, TFs do not bind a single signal sequence, but a subset of a closely related targets. However, the sequence composition defines the TF-DNA binding strength (binding affinity). Several studies linked different diseases and cancer types to gentic variants occurring in TFBS (**???**). Moreover, the misregulation of gene expression governed by TFs caused by variants occurring in TFBS could affect the entire cell environment and be propagated to neighboring cells. Therefore, identifying such regulatory motifs would provide fundamental insights on the complex mechanisms governing gene expression and the cell environment.

2.1 Discovering Transcription Factor Binding Site motifs

Several experimental assays have been developed to determine the binding site sequences of TFs in living cells or organisms (*in vivo*), or in test-tubes using synthetic or purified components (*in vitro*) (**?**). Early methods, like electrophoretic mobility shift assay (EMSA) (**?**) or footprinting (**?**), generally search the binding sites for the investigated TFs analyzing a relatively small number of target sequences, producing small datasets of bound sequences. The introduction of in vitro and in vivo high-throughput protocols, like PBM, SELEX or ChIP methods (**???**), enabled the analysis of most of all possible target sites for the investigated factors, returning large datasets of bound sequences, and providing an unprecedented opportunity to study and determine TFs binding landscapes. Experimental assays can recover the bound sequences along with their binding affinity values. However, such datasets can hide unbound sequences erroneously reported as binding sites (false positives). Moreover, most assays capture several additional nucleotides the target sites, limiting the data resolution, and making manual analyses difficult and often unfeasible. Motif discovery algorithms provide a computational framework to analyze the datasets produced by experimental assays, discovering the sequences potentially bound by TFs and reporting their predicted affinities. Given a sequence dataset, motif discovery algorithms recover sets of short and similar sequence elements. The prioritized sequence elements are later used to construct a motif model, summarizing the diverse binding site configurations observed among the prioritized sequences, and encoding

Figure 2.1. A human transcription factor (CTCF) binding its DNA target sequence.

their recurrent patterns and similarities. The development of motif discovery methods is one of the most investigated and active research fields in computational biology (**?????**). Several different algorithms and models have been proposed to discover and represent TFBS motifs. Position weight matrices (PWMs) (**?**) are the most popular models to represent TFBS motifs. PWMs are powerful and interpretable models, encoding the probability of observing a given nucleotide in each TFBS position. However, PWMs have some limitations, like the assumption of independence among the binding site positions. Therefore, several alternative models have been proposed to describe TFBS motifs (**???**). The derived motif models can be employed in many downstream analyses, like searching potential binding site occurrences in regulatory genomic sequences, predicting the sets of genes regulated by the investigated TFs, or assessing how genetic variants could affect the factors' binding landscape.

2.1.1 Experimental methods to discover Transcription Factor Binding Sites

During the last decades, several techniques have been introduced to experimentally identify and assess TF binding sites and binding preferences (**?**) (**Fig.2.2 (A)** and **Table 2.1**). Early studies on TF binding focused their analysis on gene promoters (**?**) and employed in vitro methods, such as Electro-Mobility Shift Assay (EMSA) (**?**) or DNase footprinting (**?**). EMSA exploits non-denatured polyacrylamide gel properties to separate bound and unbound DNA sequences, while DNase footprinting combines EMSA with DNase I cleavage. Generally, these assays produce datasets of a few hundreds of bound sequences, exploring a limited spectrum of TFs binding landscape. Moreover, EMSA and DNase footprinting often suffer from high false positive rates due to the identification of unwanted protein-DNA interactions, caused by nonspecific DNA binding factors potentially leading to wrongly measured binding preferences (**?**). The introduction of NGS technologies revolutionized the study of TFBS identification by encouraging researchers to develop methods that exploit the power of massively parallel sequencing, to identify TFBS. These methods have two major advantages: (i) they do not require any prior knowledge on the binding site sequence (**??**), and (ii) produce datasets of thousands of bound sequences allowing to better characterize TFs binding preferences (**?**). Protein binding microarrays (PBMs) (**??**) recover short TFBS sequences

Experimental assay	**Description**	**Output**	***De novo* motif discovery capability**	**Type**	**Identification of genomic binding locations**	**Throughput**
Competition EMSA	Bound DNA sequences are identified by observing changes in the electrophoretic migration of DNA sequences through non-denatured polyacrylamide gel.	Bound DNA sequences.	No. Used to validate known binding sites.	*in vitro*	No	Low
DNase footprinting	Pools of DNA sequences are incubated with the TF of interest; then, the DNA is degraded using DNase-I. The unbound fragments are cut in all positions, while the bound DNA is protected by the TF.	Bound DNA sequences.	No. Used to validate known binding sites.	*in vitro*	No	Low
Protein Binding Microarrays	Arrays of 40,000 spots with short immobilized DNA sequences are incubated with a GST-tagged TF, and then washed to remove weakly bound proteins. The bound sequences are identified through fluorescence-based detection.	Continuous values describing fluorescence intensity on each array spot.	Yes. Limited to short motifs (12bp).	*in vitro*	No	High
HT-SELEX	The TF is added to a pool of randomized DNA fragments. The bound sequences are selected and constitute the starting pool for the next experimental round. The procedure is repeated for several rounds. Sequencing is employed to recover the sequence of the bound DNA fragments.	DNA sequences.	Yes	*in vitro*	No	High
ChIP-based technologies	TF-DNA complexes are crosslinked with formaldehyde, and immunoprecipitated employing TF-specific antibodies. The bound sequences are then prioritized employing qPCR microarrays (ChIP on Chip) or through sequencing (ChIP-seq). ChIP-exo integrates exonuclease treatment to enhance sequence resolution.	Genomic binding location coordinates.	Yes. Limited by the inability to distinguish direct and indirect binding.	*in vivo*	Yes	Low

Table 2.1. In vivo and in vitro experimental assays to identify and validate Transcription Factor binding sites. Generally, EMSA and DNase footprinting are used to validate known TFBS, while currently PBMs, HT-SELEX, and ChIP-based methods are preferred to discover novel binding sites. Importanlty, ChIP-based assays are the only methods that recover the TF genomic binding locations. The throughput column refers to the number of samples that can be processed in parallel by each method (high: about hundreds of samples; low: few samples).

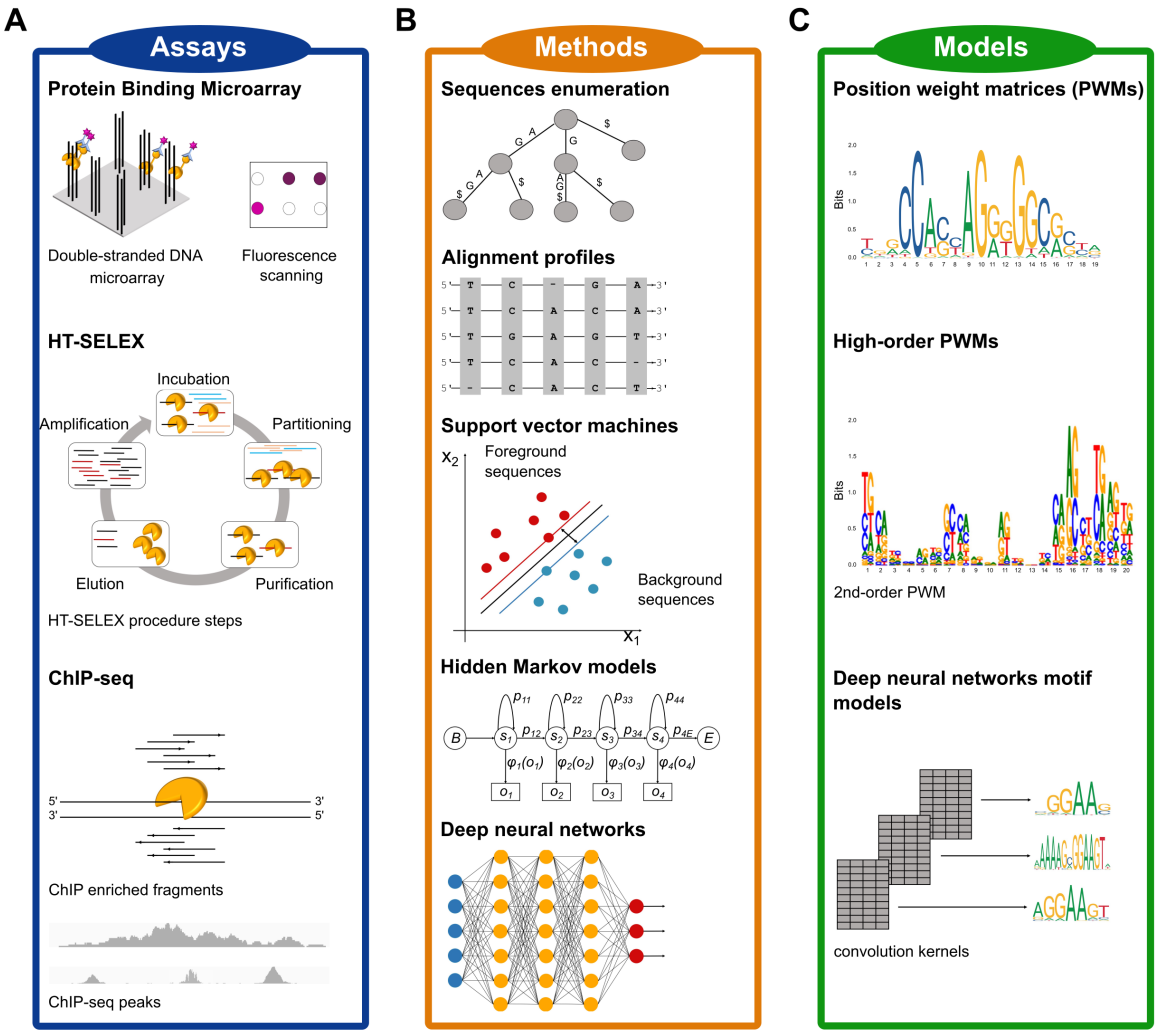

Figure 2.2. Experimental and computational methods to discover TFBS and popular models to represent binding site motifs. (A) Protein binding microarray (PBM), HT-SELEX and ChIP-seq during the last years have become the most popular assays to determine TF binding profiles and identify their target sites (TFBS). **(B)** Computational motif discovery methods can be grouped in five classes, based on the algorithms employed to discover TFBS: enumerative, alignment-based, probabilistic graphical models-based, support vector machines (SVMs), and deep neural network (DNN) based methods. **(C)** TFBS sequences prioritized by motif discovery algorithms are encoded in computational models representing the binding preferences of the investigated TFs.

(10 bp), and measure TF binding preferences in vitro. In PBMs a tagged TF is released on a glass slide containing thousands of spots filled with short immobilized DNA sequences. The tagged TFs are then incubated with fluorescent antibodies against the tag and subsequently washed to remove weakly bound factors. The fluorescence and DNA sequence enrichment are then used to quantify the TF-DNA binding strength and capture the bound sequences. Generally, the recovered sequences do not contain nucleotides flanking the investigated binding sites, producing high resolution datasets. However, since the number of possible sequences grows as a function of the target length, PBMs can assess only a limited number of target sequences (**??**). Therefore, PBMs analysis is usually constrained to binding sites 10-12 bp long. HT-SELEX (**??**) is a widely used in vitro method, coupling SELEX with high throughput sequencing. A TF is released on a pool of randomized DNA sequences, to allow the factor to select its target sites. The resulting TF-DNA complexes are separated from unbound sequences, and subsequently amplified through PCR and sequenced. The resulting DNA library is enriched in binding sites for the studied TF and is used as the starting pool for another SELEX run (**??**). SELEX captures the binding preferences of TFs without requiring any prior knowledge on their target sites (**?**). Since SELEX reaction is typically performed in liquid phase and consequently does not suffer from physical constraints, the sequence space covered by HT-SELEX is often larger than that of PBMs. Moreover, by coupling sequencing with DNA barcode indexing, HT-SELEX allows to analyze of hundreds of TFs in a single run. HT-SELEX produces datasets of thousands of high resolution bound sequences, which include

only a few nucleotides flanking the binding sites. However, since the starting DNA library is constituted by randomized sequences, HT-SELEX cannot recover the genomic binding locations for the investigated factor. The introduction of chromatin immunoprecipitation (ChIP) technologies (**?**) radically changed the study of TFBS binding, enabling the genome-wide identification of regions bound by TFs in vivo. In ChIP the TF-DNA complexes are cross-linked using formaldehyde. The DNA is then fragmented in 100-1000 bp long fragments, and subsequently immunoprecipitated with antibodies specific for the investigated TF. To recover the bound sequences, the cross-links are reverted. Then, the resulting fragments are amplified through microarray hybridization (ChIP on Chip (**??**)), or sequencing (ChIP-seq (**??**)). To locate the binding regions, the recovered DNA fragments are mapped onto the genome. After ChIP-seq reads mapping, peak calling algorithms (**???**) are employed to predict the genomic binding locations for the investigated factor. Peak calling algorithms identify the genomic regions showing greater enrichment in mapped DNA probes with respect to a control experiment (ChIP-seq peaks), and mark those regions as binding locations, or peaks (**?**). ChIP methods produce large datasets of thousands of genomic regions enriched in TFBS, whose length ranges from few hundreds to thousands of nucleotides. Although ChIP technologies, and particularly ChIP-seq, are currently considered the current 'golden standard', they have several limitations. (i) experimental results are influenced by chromatin states (most TFs bind only open chromatin [1]). (ii) ChIP can detect indirect binding, identifying other TFBS not belonging to the investigated factor (**?**). (iii) ChIP-seq peaks may be false positives, recovered because of poor antibody quality, for example (**?**). (iv) ChIP-seq returns low resolution datasets, whose sequences include several nucleotides flanking the target TFBS. ChIP-exo (**?**) addresses the latter issue, employing a lambda exonuclease to trim ChIP sequences, removing some of the nucleotides flanking the target sites. In summary, the current high-throughput assays produce large datasets consisting of thousands of sequences potentially containing all possible binding configurations of TFBS allowing a better characterization of TFs binding landscapes. However, the sequences recovered by most of the described methods contain nucleotides flanking the TFBS, and only PBMs and HT-SELEX data marginally suffer marginally from this issue. Therefore, computational methods and models to identify and characterize TFBS become fundamental to exploit the information contained in the datasets produced by experimental assays.

2.1.2 Computational methods to discover Transcription Factor Binding Sites

TFBS motif discovery is one of the most investigated problems in computational biology (**?**). During the last decades, several different algorithmic approaches have been proposed to discover potential binding sites in DNA sequences: *enumerative*, *alignment-based*, *probabilistic graphical models*, *support vector machines-based*, and *deep neural network-based* methods (**Fig.2.2 (B)**). Regardless of the algorithms employed and the computational model used to represent TFBS motifs, the TFBS motif discovery problem can be formalized as follows. Given a set of positive DNA sequences S, obtained from an experimental assay targeting a certain TF, and a set of negative sequences B, the goal is to find one or more recurrent, short and similar subsequences in S, that maximize the discriminatory power between S and B. Such subsequences are called patterns or motifs, and are likely bound by the investigated TF. The negative set B can contain randomly generated or selected genomic sequences, with similar nucleotide content and length of those in S. The retrieved patterns are used to construct and train a computational model M (motif model), representing the discovered motif. These models can then be used to identify new potential binding sites, given a new set of sequences, and to predict the strength of the TF-DNA binding. Motif discovery can be considered a classification or a regression problem, depending on the type of data used to train M. The datasets derived by experimental assays like ChIP-seq or HT-SELEX provide hundreds or thousands of sequences containing binding sites. In this setting, motif discovery becomes a classification problem. In fact, the goal is to discriminate between bound and unbound sites in the input sequences, and train the motif model with the identified binding sites. Datasets produced by other experimental technologies like PBMs provide the binding strength for large sets of sequences of equal length. Therefore, rather than discriminating between bound and unbound sequences, in this setting the motif model learns the binding affinity values associated to each target site in the input dataset, transforming motif discovery into a regression problem. In both settings the final goal is to derive a computational model M, describing the recovered TFBS and capable of predicting new binding events along with their affinity in sequences not used during model training. The most common models to represent TFBS are *consensus sequences* (**?**), *position weight matrices* (PWMs) (**??**), *high-order PWMs* (**??**), *k-mer-based* (**?**), and *deep neural network-based* (**?**) models. Consensus sequences summarize in a single sequence the TFBS structure, displaying at each position the most frequent nucleotide. However, consensus sequences do not provide the contribution of each nucleotide in each binding site position to the TF-DNA binding. PWMs address

this limitation providing the contribution of each nucleotide to TF binding. However, PWMs do not account for potential dependencies among TFBS nucleotides. High-order PWMs extend the PWM model to consider local dependencies between neighboring TFBS positions. K-mer-based models provide the contribution of each k-mer to the TFBS, implicitly integrating dependency among nucleotides and local sequence features in the model. Deep neural network-based motif models integrate in the model complex genomic features governing TF-DNA binding, significantly improving model's performance, but reducing its interpretability.

Enumerative motif discovery algorithms (**Fig.2.2 (B)**) assume that motifs are overrepresented patterns in the input dataset S with respect to a set of background genomic sequences B. Enumerative algorithms may assume that the motif length $|M|$ is known a priori. Let us assume $|M| = k$, the general idea is to collect the approximate occurrences of all potential 4^k k-mers in the sequences of S and assess if the difference between the number of matches found in S and B, or the number of expected matches according to a background model, is statistically significant. Then, an ungapped alignment is built from the statistically significant k-mers to derive a PWM. However, searching the approximate occurrences of all 4^k k-mers quickly becomes impractical, even for small k. Early proposals introduced the usage of heuristics to reduce the search space, like searching exact occurrences in a subset of the input sequences (**?**), or restricting mismatching locations to fixed motif positions (**?**). Since mismatches can occur at any motif position, these solutions partially address the motif discovery problem. The application of indexing data structures like suffix trees (STs) (**?**) allowed to explore the entire motif search space, by providing an efficient data structure speeding-up the exhaustive search for motif candidates. Weeder (**??**) and SMILE (**?**) extended this idea to consider approximate pattern matches without restrictions on mismatches locations. To assess the statistical significance of motif candidates, SMILE compares the number of motif occurrences found in S with those retrieved in a background dataset. Similarly, Weeder compares the frequency of motif candidates with their expected frequency in promoters region of the same organism from which the input sequences were collected. However, these methods can take several hours to search for long motifs, or analyze large sequence datasets enriched in similar motif occurrences like those produced by PBMs, HT-SELEX, or ChIP methods (**?**). In such datasets is sufficient to allow one or two mismatching positions to recover the TF binding sequences, while Weeder performs an exhaustive search on all k-mers, allowing more than two substitutions, increasing the algorithm running time. Therefore, during the last decades the community proposed several enumerative algorithms specifically designed to analyze the large datasets produced with NGS-based assays. MDscan (**?**) uses word enumeration to find motif candidates in ChIP on Chip datasets. Instead of enumerating all the potential words from the sequences in S, MDscan extracts the non-redundant patterns from the most enriched sequences. To assess the statistical significance of motif candidates, MDscan employs a third-order Markov model as background. Amadeus (**?**) evaluates all k-mers in S, obtained from ChIP on Chip experiments, and groups the similar patterns in lists. The similar patterns lists are grouped in motifs, which are statistically evaluated using a hypergeometric test. DREME (**?**) searches motif candidate sequences using regular expressions. To statistically evaluate the discovered motifs, DREME employs the Fisher's exact test, comparing the number of sequences in S and B, in which the motifs occur. Trawler, HOMER and STREME (**???**) use STs similarly SMILE or Weeder. Trawler enumerates all sequence patterns using STs indexing S and B, and measures patterns frequencies to find overrepresented sequences in the input dataset. To match motif sequences, Trawler employs degenerate consensuses, allowing mismatching positions. To statistically evaluate the prioritized patterns, Trawlers uses the z-scores, derived from the normal-approximation to the binomial distribution. Since the function does not correct for the effect of overlapping motif instances, the computations are significantly faster. The similar and significant patterns are clustered to form motifs. HOMER algorithm was specifically designed to discover TFBS motifs in ChIP-seq data. HOMER indexes the foreground and background datasets using STs and searches for k-mers overrepresented in S, through approximate pattern matching on the ST. HOMER employs the hypergeometric or binomial test to statistically evaluate each motif candidate. Using the hypergeometric or the binomial tests reduces the running time, by avoiding the explicit count of k-mers frequencies. STREME builds a ST from S and B, and counts the number of exact matches of seed words of different lengths in both datasets, evaluating the statistical significance of their enrichment using Fisher's exact test or the binomial distribution. Then, STREME counts the number of the approximate matches of the most significant words on the ST. The prioritized words are then grouped to derive the corresponding motifs. Importantly, STREME requires one single tree visit to discover motifs of different lengths, significantly speeding-up the discovery of different TFBS motifs.

Alignment-based motif discovery algorithms compute alignment profiles to describe motif binding preferences (**Fig.2.2 (B)**), avoiding the exhaustive k-mers enumeration. The rationale is to build an

alignment from k-mers selected from the input dataset S, and score the resulting profile through appropriate measures, like nucleotide conservation, information content, or profile statistical significance. Motif statistical significance is based on the probability of obtaining the same alignment from random sequences or from a background dataset B. Alignment-based motif discovery algorithms usually assume that the motif length $|M|$ is known *a priori*. For alignment-based algorithms motif discovery can be formalized as a combinatorial problem. Let us assume $|M| = k$, the goal is to find the best alignment profile built combining k-mers from S, according to a scoring criterion. The best alignments are then used to derive the corresponding PWMs. Most alignment-based algorithms assume that each sequence in S contains one or zero binding sites. Therefore, there exist $(\sum_{s \in S} |s| - |M| + 1)^{|S|}$ possible profiles, built by combining k-mers in all possible ways. Since enumerating all possible solutions is computationally impractical even for small dataset, alignment-based algorithms explore the solution space using heuristics, such as greedy (**?**), expectation-maximization (EM) (**?**), stochastic (e.g. Gibbs sampling) (**?**), or genetic algorithms (**?**). CONSENSUS (**?**) proposed a greedy approach to incrementally build the motif alignment profiles. The problem is initially solved on two sequences, then is progressively solved by adding the remaining sequences $s \in S$ one by one. The MEME algorithm (**???**) proposed a different strategy to explore the solution space by iteratively refining an initial profile, substituting some k-mers in the profile with others more likely to produce better solutions. By employing an EM strategy, MEME evaluates how well each k-mer in $s \in S$ fits the current alignment profile, rather than a background model. The algorithm has two main steps: the E-step and the M-step. During the E-step, MEME computes a likelihood score for each k-mer in S, using the current alignment profile. In the M-step, MEME assigns a weight to each k-mers in the current profile proportional to the scores computed during the E-step and updates the alignment by substituting low scoring k-mers with others better fitting the current profile. However, the EM algorithm can prematurely converge to local maxima and convergence depends on the algorithm starting conditions. Stochastic optimization strategies like Gibbs sampling (**?**) address these limitations. Given an initial profile built with k-mers randomly selected from each $s \in S$, at each iteration the algorithm removes from the profile the k-mer coming from a certain s. Then, the algorithm assigns a likelihood score to each k-mer in s using the modified profile. Similarly to MEME, the score describes how well each k-mer fits the profile rather than a background model. Then, a new k-mer from s is chosen to replace the removed subsequence in the profile, with probability proportional to its likelihood score. The procedure is repeated until a fixed number of iterations or no further updates applied to the profile. Therefore, while EM optimization strategies like MEME update the profile selecting the k-mers deterministically according to how well they fit the alignment, Gibbs sampling algorithms optimize the profile selecting the subsequences to remove and insert using a stochastic approach. However, the basic Gibbs sampling algorithm assumes that each sequence contains exactly one binding site. In (**?**) the authors extended the algorithm to consider sequences containing one, multiple, or no binding sites. AlignACE (**?**) and ANN-spec (**?**) furtherly modified the Gibbs sampling algorithm, enabling the simultaneous search for TFBS on both strands. Moreover, ANN-spec coupled the stochastic motif search with a perceptron artificial neural network learning the best alignment from the binding specificities observed in S. BioProspector (**?**) and MotifSampler (**?**) used third order Markov models as background, improving the predictive performance of Gibbs sampling. GLAM (**?**) modified the Gibbs sampling strategy to estimate the optimal alignment length, and employed simulated annealing for profile optimization. GLAM algorithm was then furtherly extended to also consider gapped motifs (**?**). Although stochastic and EM are the most popular optimization strategies for alignment-based algorithms, researchers also focused on adopting other optimization methods. GADEM (**?**) paired EM local search with genetic algorithms for profile refinement. However, using alignment profiles the solution search space grows exponentially with the size of S. Even employing heuristics, the analysis of the datasets produced by NGS assays rapidly becomes impractical and requires large amounts of time and computational resources (**?**). Therefore, researchers focused on developing new alignment-based motif discovery algorithms tailored to analyze the datasets produced by high-throughput protocols. MEME-ChIP (**?**) and STEME (**?**) improved the original MEME algorithm for the analysis of ChIP datasets. MEME-ChIP focuses the analysis on a random subset of sequences in S, reducing the theoretical size of the solution space. STEME indexes the sequences in S using suffix trees and avoids to evaluate the likelihood of all possible k-mers during MEME's E-step, by employing a branch and bound strategy on the ST. ChIPMunk (**?**) proposed a greedy profile optimization similar to EM, to discover motifs in large ChIP datasets, while accounting for ChIP peaks shape. ChIPMunk explores the solution space to find the alignment maximizing the profile discrete information content (**?**), and uses peaks shape to weight the contribution of each sequence $s \in S$ to the motif. XXmotif (**?**) combined enumerative motif discovery with profile refinement, by iteratively selecting sets of k-mers from S maximizing their fitness to the profile, and increasing the motif fitness to S. Similarly, ProSampler

(**?**) proposed a highly optimized hybrid method to discover motifs in large ChIP-seq datasets, combining motif enumeration with Gibbs sampling to refine preliminary motif profiles.

The importance of dependencies between TFBS nucleotides and how including them in motif models would improve the models' performance has been controversially debated (**???**). However, some studies demonstrated the existence of such dependencies in TFBS, between neighboring and non-neighboring nucleotides (**??**). Enumerative and alignment-based algorithms represent motifs without accounting for potential dependencies between the binding site positions, modeling TFBS as PWMs. PWMs can be extended to count the frequency di- or tri-nucleotides, producing high-order PWMs, like dinucleotide weight matrices (DWMs) (**?**). Dimont (**?**) and diChIPMunk (**?**) employed DWMs to discover and represent motifs. However, dependencies may exist also between non-neighboring positions and not between neighboring nucleotides. Therefore, the models could learn dependencies not in the data, often overfitting the training set S. Probabilistic graphical models (**Fig.2.2 (B)**) like Bayesian networks (BNs) or Markov models (MMs) provide flexible and efficient frameworks to capture and encode dependencies between TFBS nucleotides. Barash and coworkers (**?**) proposed to model TFBS motifs as BNs trained through an EM strategy. Importantly, BNs can capture dependencies between neighboring and non-neighboring motif positions. However, the model assumes the same order of dependence throughout the entire binding site. Similarly, in (**?**) the authors introduced VOBN models representing TFBS as BNs, but accounting for variable orders of dependencies between neighboring and non-neighboring TFBS positions. However, BNs training is often computationally demanding and requires large amounts of data to avoid the model to overfit S. MMs and Hidden Markov models (HMMs) provide efficient frameworks to include dependencies between motif positions, and more scalable and efficient training procedures compared to BNs. TFFMs (**?**) proposed a HMM-based model capturing dinucleotide dependencies between neighboring motif positions. TFFM models also capture the properties of the sequences flanking the binding site and accommodate changes in the motif length, by employing the background and insertion/deletion states, respectively. TFFMs are trained using a set of previously prioritized motif candidate sequences and employing the Baum-Welch algorithm. TFFM models assume the same order of dependency between neighboring positions across the whole TFBS. Slim models (**?**) introduced a framework to learn neighboring and non-neighboring dependencies between TFBS nucleotides, using different orders of dependency. Dependency orders are pruned in a data-driven fashion, establishing the TFBS positions on which the motif nucleotides depend on. The models are trained using the motif candidates prioritized using Dimont. Discrover (**?**) proposed a discriminative motif discovery method designed to analyze ChIP-seq datasets and modeling TFBS as HMMs. Given S and B, Discrover searches for overrepresented motifs in S using regular expressions. Then, an initial HMM is trained using the prioritized motif candidates as seeds. To identify good seeds, Discrover uses different objective functions such as likelihood, or mutual information. The initial HMM is then iteratively refined through a gradient optimization of likelihood. In (**?**) the authors proposed a method to discover CTCF (**?**) motif using variable-order MMs to capture different orders of dependency between neighboring nucleotides. The model is trained using an EM strategy iteratively refining the MM parameters, to optimize its fitness to the input dataset S. BaMMotif (**?**) introduced an efficient method to discover TFBS motifs in large datasets, employing a Bayesian approach to train Markov models. To avoid model overfitting, the motif model is trained using the conditional probabilities of $(q-1)$th order as priors for the qth order conditional probabilities. The algorithm iteratively optimizes the model parameters until convergence through an EM strategy. In the E-step, the algorithm estimates the probability that each position of each $s \in S$ is a motif starting position, using the current model. During the M-step, BaMMotif uses the motif candidates identified in the previous step to refine the model. Moreover, the algorithms dynamically adapt the model parameters during training to capture variable orders of dependencies between motif positions. Recently, the authors developed a faster and improved of the algorithm (**?**).

Support Vector Machines (SVMs) (**?**) have been successfully applied to different problems in computational biology (**?**), including TFBS motif discovery (**Fig.2.2 (B)**). The general idea is to decompose the bound sequences (foreground dataset S) and the unbound sequences (background dataset B) in sequence elements of length k (k-mers). The k-mers frequencies are then used as features to train a sequence similarity kernel (**?**), to discriminate between bound and unbound sequences. Generally, to each k-mer is assigned a weight proportional to its contribution to the definition of the positive or negative training sets, or to its likelihood of being a motif candidate. Although early k-mer based kernels (**???**) were originally developed to tackle protein sequence homology problems, recently several SVM-based algorithms extending the original kernels were proposed to discover TFBS motifs. Kmer-SVM (**??**) proposed a SVM-based method to discover TFBS motifs in ChIP-seq datasets. Kmer-SVM employs the spectrum kernel (**?**), which counts the exact matches for all contiguous k-mers in S and B, building the k-mers

feature space. By using a trie (**?**) indexing S and B, the algorithm runs in linear time. Although TFBS motifs contain highly degenerate non-informative positions, kmer-SVM does not explicitly accommodate such sequence variability by counting the frequency of exact k-mers matches.

Motif Graph

Several studies showed that TFs present population-specific (**?**), cell-type-specific (**??**), and even individual-specific (**?**) binding sites. Algorithms analyzing only the reference genome would provide general models, which could return wrong TFBS predictions when analyzing personal genomic sequences. Motif models integrating data from different populations, cell-types, or individuals would recover more reliable TFBS occurrence predictions (**Fig.3.3 (A)**). Moreover, such models would better predict the consequences of noncoding variants on TF-DNA binding events, and they could encode variable orders of nucleotide dependencies. Importantly, graph data structures are often interpretable and intuitive. With this aims in mind we developed the Motif Graph, a framework integrating the advantages of *probabilistic graphical models-based* and *SVM-based* motif discovery algorithms, without sacrificing model interpretability. A Motif Graph model G is defined by a set of vertices V, a set of edges E, and a set of paths P. Each vertex $v \in V$ on the graph is labeled with a nucleotide, or mathematically $label(v) \in \{A, C, G, T\}$. Each edge $e \in E$ represents the allowed links between consecutive nucleotides in the TFBS motif. Each path $p \in P$ represents a *haplotype* embedded in G, which correspond to one sequence used to train the Motif Graph model. Currently, the model is limited to encode 1st order dependencies between consecutive

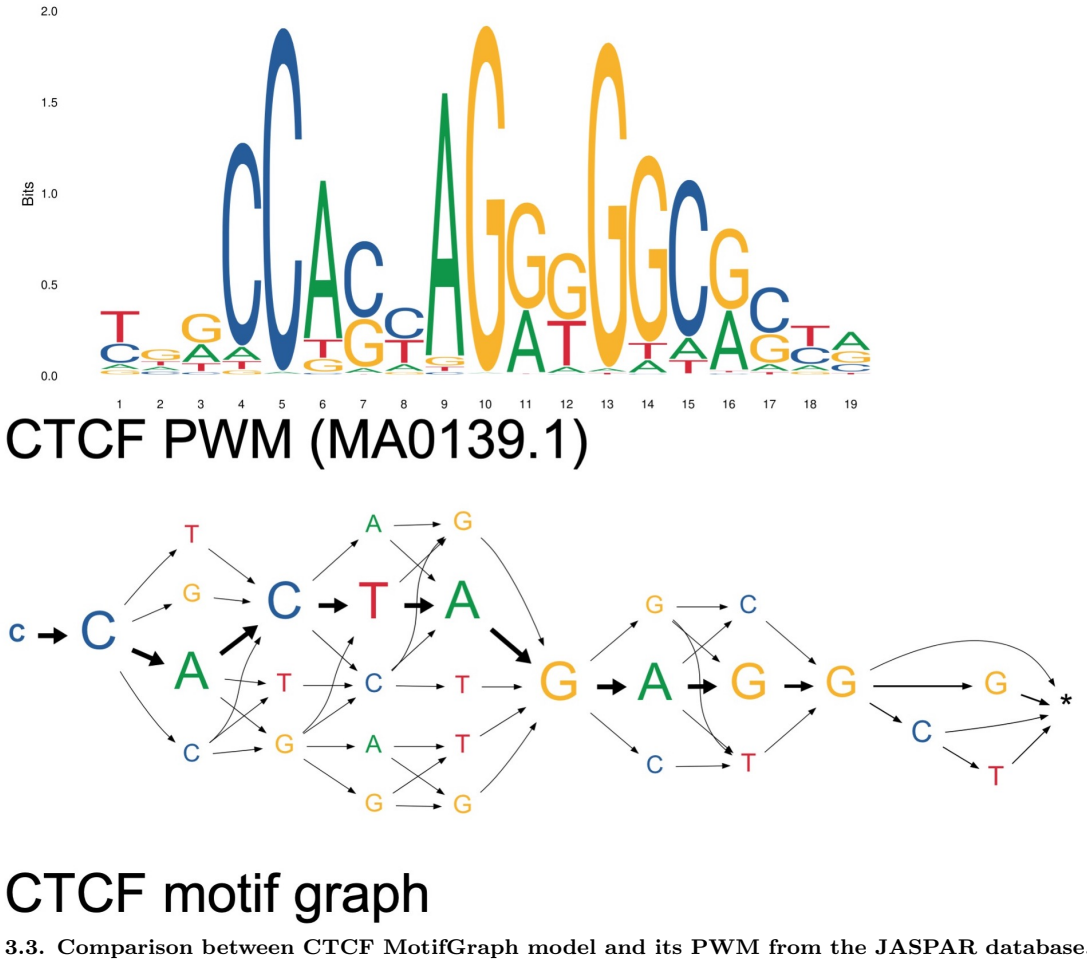

Figure 3.3. Comparison between CTCF MotifGraph model and its PWM from the JASPAR database. On top the CTCF motif available on JASPAR database (MA0139.1). On the bottom the Motif Graph model trained using 100 k-mers obtained from a ChIP-seq experiment trageting CTCF binding site on HepG2 cell line.

nucleotides. However, the model is flexible and accomodate variable length motifs, beside recording the training sequences.

3.1 Motif Graph model construction

Motif Graph motif discovery procedure (**Algorithm 1**) is constitued of two main steps: k-mers prioritization, and graph model training and construction. To prioritize the k-mers Motif Graph currently employs the k-mer based motif discovery procedure implemented in Gkm-SVM (**?**). Therefore, given a positive sequence dataset S and a background dataset B, for each k-mer of length k in S, the algorithm counts the number of matches in S and in B, allowing mismatching positions (see **Section 2.1.2** for further details). Then, a similarity kernel is trained using the recovered k-mers frequencies. The trained kernel assigns to each k-mer in S and in B a weight proportional to its contribution in defining the foreground or the background dataset. The algorithm then ranks the k-mers according to their weight scores. The Motif Graph model is iteratively trained employing a greedy approach, which adds the top ranked k-mers to G incrementally, one by one (see **Algorithm 2** for details). Each k-mer is aligned to the current G to maximize the number of nucleotides matching the current model. While aligning the k-mers to the current Motif Graph model, the algorithm shifts the input k-mer on the right and on the left up to a defined offset number of nucleotides. In our experiments with set the offset to 3. Once built the model, we construct a scoring matrix similar to the widely used $PSSM$. However, our scoring matrix account for 1st order dependencies between nucleotides, recalling the well-known DWMs (see **Section 2.1.3** for details). The scoring matrix is used to assign a likelihood score and classify new sequences as potentially bound or not bound by the investigated factor. In other words, the score describes how likely is the scanned sequence to contain a potential binding site. To score a sequence we slide the scoring matrix along the string employing a procedure similar to classical PWM scanning tools like FIMO (**?**).

Algorithm 1: Motif Graph motif discovery.

Input: S, B, k
Output: G
frequencies ← countFrequencies(S, B, k)
kernel ← trainKernel(frequencies)
kmers, weights ← extractWeights(kernel)
rankedKmer ← sort(kmers, weights)
$G \leftarrow \emptyset$
for *kmer in rankedKmers* **do**
    $G \leftarrow$ addKmers(G, kmer)
return G

Algorithm 2: Motif Graph model training.

Input: G, kmer, i
Output: G
if $i = 1$ **then**
    return G
for j *in* 1 *to* 3 **do**
    matchesLeftOffset, alignmnetLeft ← countMatchesLeftOffset(G, kmer, j)
for j *in* 1 *to* 3 **do**
    matchesRightOffset, alignmnetRight ← countMatchesRightOffset(G, kmer, j)
alignment ← getBestAlignment(matchesLeftOffset, alignmentLeft, matchesRightOffset, alignmentRight)
$G \leftarrow$ insertKmer(G, alignment)
return G

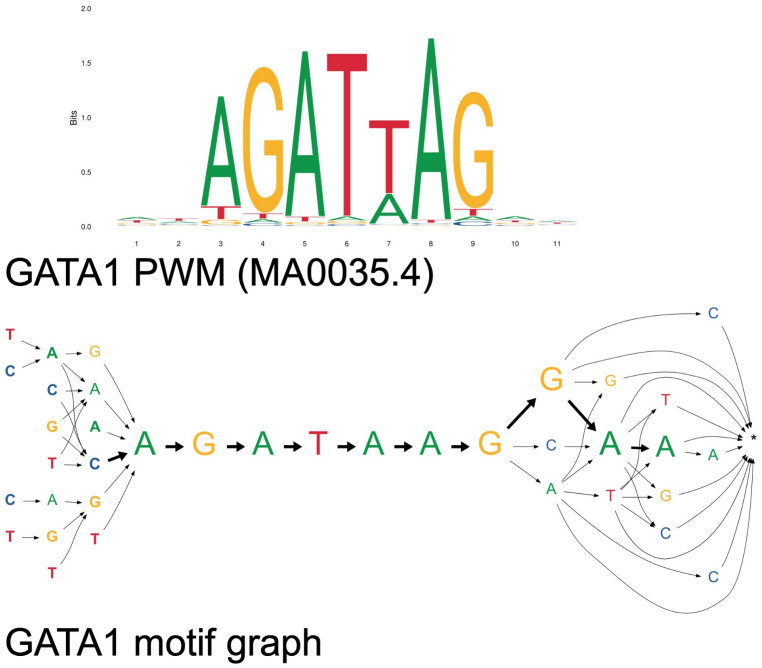

Figure 3.4. Comparison between GATA1 MotifGraph model and its PWM from the JASPAR database. On top the GATA1 motif available on JASPAR database (MA0035.4). On the bottom the Motif Graph model trained using 100 k-mers obtained from a ChIP-seq experiment trageting GATA1 binding site on K562 cell line.

Training k-mers	10	50	100	200	350	500	750
AUC	0.69	0.69	0.72	0.63	0.60	0.55	0.55
F1-score	0.62	0.69	0.71	0.65	0.62	0.57	0.56

Table 3.2. CTCF Motif Graph model AUC and F1-scores values with different number of training k-mers.

3.2 Results

To test our motif discovery algorithm we obtained 10,000 ChIP-seq peak sequences from the ENCODE Project database (**?**) for CTCF and GATA1 Transcription Factors, obtained on the HepG2 and K562 cell lines, respectively. The original ChIP-seq datasets were sorted according to the peaks enrichment score in decreasing order, in order to test our algorithm on reliable peaks. Interestingly, the trained Motif Graph models were closed to the motifs PWM available on the JASPAR database (**?**) for both CTCF and GATA1 (**Fig.3.3** and **Fig.3.4**). For both TFs, the Motif Graph models captured the main motif sequence. The main motifs are also enforced by the edge thickness which is proportional to the number of training k-mers supporting each $p \in P$. Then, we tested the discriminative performance of both Motif Graph model, with different number of training k-mers (**Fig.3.5**), to establish the optimal number of training sequences for the CTCF and GATA1 Motif Graph models. To compare the models discriminative performance, we performed a cross-validation experiment using splitting the S and B dataset in training and testing set (75% and 25%, respectively). We trained the models with 10, 50, 100, 200, 350, 500, and 750 k-mers. For CTCF the best prformance in terms of both AUC (0.72) and F1-score (0.71) were obtained training the Motif Graph on 100 k-mers (**Table 3.2**). For GATA1 the model returned the best AUC using 200 k-mers (0.76), while the best F1-score (0.70) was obtained training the model with 100 sequences (**Table 3.3**). Then, we compared the Motif Graph models discriminative performance against the corresponding PWM and DWM models, recovered from JASPAR (**?**) and HOCOMOCO (**?**) databases, respectively. For CTCF, both the PWM and the DWM models returned better predictive performance than our model (**Fig.3.6 (A)**). On the other hand, on GATA1 data our model showed better performance than PWMs, but still performed worse than DWMs (**Fig.3.6 (B)**).

Training k-mers	10	50	100	200	350	500	750
AUC	0.73	0.75	0.75	0.76	0.74	0.74	0.71
F1-score	0.68	0.68	0.70	0.69	0.70	0.69	0.67

Table 3.3. GATA1 Motif Graph model AUC and F1-scores values with different number of training k-mers.

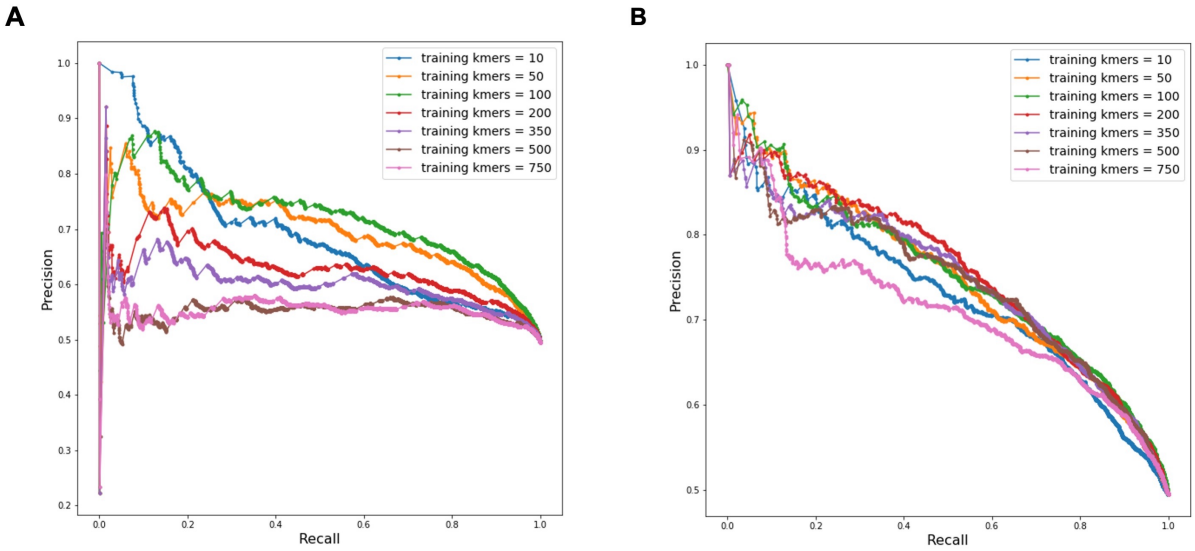

Figure 3.5. Precision-Recall curves obtained varying the number of k-mers used to train the Motif Graph models. To establish the number of training k-mers returning the best discriminative performance we computed the Precision-Recall curves of different Motif Graph models trained using 10, 50, 100, 200, 350, 500, and 750 k-mers on **(A)** CTCF and **(B)** GATA1

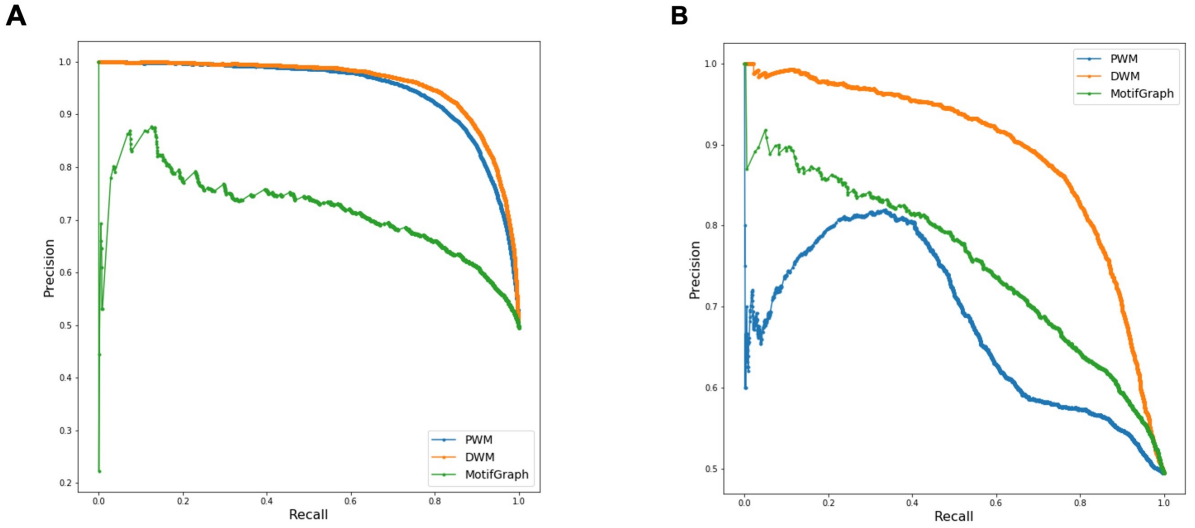

Figure 3.6. Comparing Motif Graph, PWM, and DWM Precision-Recall curves. We compared the discriminative power of the Motif Graph models against that of PWMs and DWMs for both **(A)** CTCF and **(B)** GATA1.