Department of Computer Science
Series of Publications A
Report A-2019-10

Modeling and learning monomeric and dimeric transcription factor binding motifs

Jarkko Toivonen

Doctoral dissertation, to be presented for public examination with the permission of the Faculty of Science of the University of Helsinki, in Room D122, Exactum building, on the 22nd of November, 2019 at 12 o'clock.

University of Helsinki
Finland

Supervisor
Esko Ukkonen, University of Helsinki, Finland

Pre-examiners
Harri Lähdesmäki, Aalto University, Finland
Matti Nykter, Tampere University, Finland

Opponent
Juho Rousu, Aalto University, Finland

Custos
Veli Mäkinen, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Pietari Kalmin katu 5)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://cs.helsinki.fi/
Telephone: +358 2941 911

ISSN 1238-8645
ISBN 978-951-51-5601-3 (paperback)
ISBN 978-951-51-5602-0 (PDF)
Computing Reviews (1998) Classification: G.1.6, G.2.1, I.2.6, I.5.1, J.3
Helsinki 2019
Unigrafia

Modeling and learning monomeric and dimeric transcription factor binding motifs

Jarkko Toivonen

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
jarkko.toivonen@cs.helsinki.fi

PhD Thesis, Series of Publications A, Report A-2019-10
Helsinki, November 2019, 61+109 pages
ISSN 1238-8645
ISBN 978-951-51-5601-3 (paperback)
ISBN 978-951-51-5602-0 (PDF)

Abstract

In this thesis we aim to learn models that can describe the sites in DNA that a transcription factor (TF) prefers to bind to. We concentrate on probabilistic models that give each DNA sequence, of fixed length, a probability of binding. The probability models used are inhomogeneous 0th and 1st order Markov chains, which are called in our terminology Position-specific Probability Matrix (PPM) and Adjacent Dinucleotide Model (ADM), respectively. We consider both the case where a single TF binds in isolation to DNA, and the case where two TFs bind to proximal locations in DNA, possibly having interactions between the two factors. We use two algorithmic approaches to this learning task.

Both approaches utilize data, which is assumed to have enriched number of binding sites of the TF(s) under investigation. Then the binding sites in the data need to be located and used to learn the parameters of the binding model. Both methods also assume that the length of the binding sites is known beforehand.

We first introduce a combinatorial approach where we count ℓ-mers that are either binding sites, background noise, or belong partly to both of these categories. The most common ℓ-mer in the data and its Hamming neighbours are declared as binding sites. Then an algorithm to align these binding sites in an unbiased manner is introduced. To avoid false binding sites, the fraction of signal in the data is estimated and used to subtract the counts that rise from the background.

The second approach has the following additional benefits. The division into signal and background is done in a rigorous manner using a maximum likelihood method, thus avoiding the problems due to the ad hoc nature of the first approach. Secondly, use of a mixture model allows learning multiple models simultaneously. Then, subsequently, this mixture model is extended to include dimeric models as combinations of two binding models. We call this reduction of dimers as monomers modularity. This allows investigating the preference of each distance, even the negative distance in the overlapping case, and relative orientation between these two models. The most likely mixture model that explains the data is optimized using an EM algorithm. Since all the submodels belong to the same mixture model, their relative popularity can be directly compared. The mixture model gives an intuitive and unified view of the different binding modes of a single TF or a pair of TFs.

Implementations of all introduced algorithms, SeedHam and MODER for learning PPM models and MODER2 for learning ADM models, are freely available from GitHub. In validation experiments ADM models were observed to be slightly but consistently better than PPM models in explaining binding-site data. In addition, learning modularic mixture models confirmed many previously detected dimeric structures and gave new biological insights about different binding modes and their compact representations.

Computing Reviews (1998) Categories and Subject Descriptors:

G.1.6	Optimization: Constrained optimization, Global optimization
G.2.1	Combinatorics: Combinatorial algorithms, Counting problems
I.2.6	Learning: Parameter learning
I.5.1	Models: Statistical
J.3	Life and medical sciences: Biology and genetics

General Terms:
algorithms, machine learning, bioinformatics

Additional Key Words and Phrases:
expectation maximization, Markov chain, motif discovery, gene expression, transcription regulation

Acknowledgements

I would like to thank my supervisor Esko Ukkonen for patiently guiding me through my doctoral studies, even while officially retired. I would also like to thank Veli Mäkinen for being the professor officially in charge after Esko retired.

Jussi Taipale has been invaluable source of biological problems to solve, and he has also great insight in possible ways of solving them. Teemu Kivioja has been my main source of information related to biology and bioinformatics. I am thankful to my pre-examiners for comments that greatly enhanced the thesis. I also appreciate the constructive comments about the thesis from Teemu Kivioja and Leena Salmela. Special thanks for funding my studies go to FDK, Algodan, SYSCOL, HIIT, and Digiloikka.

The people of the IT support of the department as well as the IT for science group deserve my thanks for getting me out of the abnormal situations I have often managed to get myself into. In addition, Pirjo Moen has been really helpful with all the practicalities related to the final stages of the graduation.

Thanks as well to my current and former colleagues Esa Junttila, Janne Korhonen, Jussi Kollin, Leena Salmela, Niina Haiminen, Pauli Miettinen, Pekka Parviainen, and Teppo Niinimäki for random discussions and lunch companion. Thanks also to all the people I met at the DoCS pizza evenings.

Lastly I would like to thank my parents and my sister.

Helsinki, November 2019
Jarkko Toivonen

Contents

Original papers

This thesis consists of an overview and the following three papers, referred to by Roman numerals I, II, and III in this overview. These papers are reproduced at the end of this thesis.

I Jarkko Toivonen, Jussi Taipale, and Esko Ukkonen. Seed-driven learning of position probability matrices from large sequence sets. In *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)*, Ed. by R. Schwartz and K. Reinert. Vol. 88. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pages 25:1–25:13, 2017.

II Jarkko Toivonen, Teemu Kivioja, Arttu Jolma, Yimeng Yin, Jussi Taipale, and Esko Ukkonen. Modular discovery of monomeric and dimeric transcription factor binding motifs for large data sets. *Nucleic Acids Research*, 46(8):e44, 2018.

III Jarkko Toivonen, Pratyush K. Das, Jussi Taipale, and Esko Ukkonen. MODER2: first-order Markov modeling and discovery of monomeric and dimeric binding motifs. Submitted.

Chapter 1

Introduction

Control of gene expression, the process of producing proteins from DNA sequences, is the mechanism that allows different cells to look and act differently, even though each cell contains the same DNA. Regulation of gene transcription is one major aspect of this control. Proteins called transcription factors (TFs) can affect gene transcription by binding to regulatory areas of corresponding genes. The transcription factor binding is specific to the nucleotides in the binding site. In order to understand regulation of gene transcription, it is necessary to be able to model these binding sites. From a set of DNA sequences that are known to be enriched in transcription factor binding sites, one can try to learn models for binding sites.

In this thesis we consider probabilistic binding site models and algorithms for learning the models. The probabilistic models we use are inhomogeneous Markov chains of order either zero or one. Our aim is to create algorithms that can handle the large data sets that modern high-throughput experimental methods produce. Also, when trying to learn multiple binding models from a data set, many previous algorithms sequentially learn a model from data, then remove its occurrences from the data, and then start over again to learn the next model. This process might not treat each model equally as the end result will depend on the order in which the models are learned. We aim to create a method that can learn each model simultaneously, and hence treat each model the same way. Another aim is to test whether the commonly assumed independence holds between the positions of the binding sites or not.

The regulatory areas of genes can contain multiple binding sites, possibly for different transcription factors. Therefore, it is interesting to analyze if two close-by transcription factors bind independently or in co-operation. This co-operativity can be measured as the popularity of each possible distance and relative orientation of two transcription factors. In addition, it is interesting how the binding models change as the distance between two binding sites gets smaller.

Once we have learned binding models, we can use them, for example, to predict putative binding sites in genomes. In addition to mutations in genes, a major cause of disease are mutations outside genes, for example in the regulatory areas. The binding models can be used to assess the effect of a mutation in a binding site on the binding strength and subsequently to the expression of a related gene. The transcription factors can be grouped into families based on the amino-acid similarity. As another application, the binding models of transcription factors provide another way of classifying TFs. These two classification methods are largely in accordance.

About ten years ago several new technologies appeared that are capable of producing large amounts of binding data from which to learn binding models. Previously, when example data was scarce, the models learned were not very accurate, and learning of more complicated models was not possible. After this revolution in experimental technology, the computational methods needed an update as well to handle the data from these high-throughput methods.

Over the years several different computational methods have been used for learning binding models. Word based methods try to find sequences that are over-represented in comparison to either a background model or to another, negative, data set. A binding model is then built from these sequences. Probability based methods first choose a model family and then try to find model parameters that maximize the likelihood of the model. An EM algorithm or Gibbs sampling can, for instance, be used to perform this optimization task. Regression methods and recently also deep learning have been used for learning of binding models as well. In this thesis we develop both word based methods and probabilistic methods that use an EM algorithm.

1.1 Original papers

This thesis consists of three papers. Paper I presents a combinatorial method for learning a single, monomeric, Position-specific Probability Matrix (PPM). Paper II considers both monomeric and dimeric PPMs in a single mixture model, learned by an EM algorithm. Paper III extends the method of Paper II to first order Markov chains called Adjacent Dinucleotide Models (ADMs).

Paper I. This paper gives a combinatoric method, called SeedHam, to learn monomeric PPM models using a frequent ℓ-mer and its Hamming neighbours, when given the length ℓ of the binding model and sequences known to contain binding sites as input. New contributions include the correct, unbiased, alignment of a Hamming sample of binding sites into a PPM model, and an equation relating the counts of three classes of ℓ-mers in the data. An implementation of SeedHam is freely available. Experimental testing showed that SeedHam performed favourably compared to two other common methods.

Paper II. The paper introduces a mixture model of multiple monomeric models and their dimeric combinations. An EM algorithm, called MODER, to find the maximum likelihood parameter estimates of this mixture model is given. New contributions include a representation for the dimeric cases of a TF pair (COB table), which allows investigation of the relative abundances of the different dimeric cases. The mixture model also allows a rigorous investigation of overlapping dimeric cases in a unified probabilistic model, which has previously been done in ad hoc manner. Freely available implementation of MODER was used in various qualitative and quantitative experimental testing presented in the paper.

Paper III. This paper extends the mixture model and the corresponding learning algorithm to the ADM models. It also extends the alignment algorithm of a Hamming sample of binding sites to the case of ADM models. An additional contribution is the analysis of the overlapping case of two ADM models. Extended software implementation of MODER2, which is freely available, can learn both PPM and ADM models. Large-scale experimental testing showed ADM models performing slightly better than PPM models.

The contribution of the thesis author to the original papers is substantial and can be specified as follows. All the algorithms and experimental testing were implemented by the author. Paper I was written by the thesis author and Esko Ukkonen. The algorithms were designed by the three authors. Experimental testing was co-designed with Esko Ukkonen. Paper II was mostly written by the thesis author and Esko Ukkonen, with all authors giving critical feedback. The algorithm was mostly designed by the thesis author. The experimental testing was mostly designed by the thesis author, but with Teemu Kivioja, Jussi Taipale, and Esko Ukkonen contributing as well. Paper III was mostly written by the thesis author and Esko Ukkonen, with all authors giving critical feedback. The algorithm was co-designed with Jussi Taipale and Esko Ukkonen. The experimental testing was co-designed with Esko Ukkonen, with other authors contributing as well.

1.2 Outline

The rest of this overview consists of the following parts: Chapter 2 introduces the biological background necessary to understand this thesis. Basic concepts of molecular biology are introduced and more details are given about the regulation of transcription. Chapter 3 gives different formal representations of transcription factor binding sites and focuses on probabilistic models. In addition to the binding models, ways to characterize and compare models are also summarized. As a new contribution, the Co-Operative Binding table (COB) is introduced to describe the strength of different dimeric binding cases. In Chapter 4 three common application areas of binding models are described for completeness. Chapter 5 gives an introduction to the new methods of this thesis for learning binding models. Section 5.2 gives an overview of the SeedHam algorithm that is the topic of Paper I. Section 5.3 recapitulates the EM algorithm in the setting of learning binding models from sequence data, and in Section 5.4 the EM algorithm is extended to learning dimeric binding models and COBs, which is the main topic in Papers II and III. In Section 5.5 we show what needs to be changed in the EM algorithm in order to be able to learn ADM models as well, which is done in Paper III. At the end of the chapter we discuss the evaluation of the goodness of the models learned. Chapter 6 contains the results from experimental evaluations. Chapter 7 contains a concluding discussion of the thesis.

Chapter 2

Biological background

In this chapter we review some biological preliminaries needed in the rest of this thesis. For more details, see, for example, the book Molecular Biology of the Cell [1], especially its chapter seven on Control of Gene Expression.

The main components in cell biology are the macro molecules DNA, RNA, and protein. The DNA serves as storage of information, and proteins are used to build the main body of cells and function in different ways as cell machinery. *The central dogma of molecular biology* states the flow of information between these three molecules. Information can be copied from DNA to RNA (*transcription*), which is done by a molecular machine called *RNA polymerase*, and information can be copied from RNA to proteins (*translation*). However, there are no known cases where information flows from proteins to either RNA or DNA. Hence, proteins are end products of this flow. RNA, besides being the intermediate messenger between DNA and proteins, can also function as an end product. For example, *ribosomes*, that perform the translation, are machines built out of protein and RNA molecules. This central dogma of the information flow is visualized in Figure 2.1.

These three molecules can be considered as linear molecules built out of basic building blocks. In terms of computer science, these molecules can be expressed as strings over certain alphabets. For DNA this alphabet consists of four *nucleotides*. The distinguishing part that separates the four nucleotides is called a *base*, and the bases (and the related nucleotides) are marked with letters A (adenine), C (cytosine), G (guanine), and T (thymine). In RNA the base T is replaced by U (uracil). The protein alphabet consists of 20 *amino acids*. The DNA has a double-stranded structure, where two strands are joined together with base-to-base bindings, like a zipper. The base pairs in this double-stranded structure are complementary: A binds to T and C binds to G.

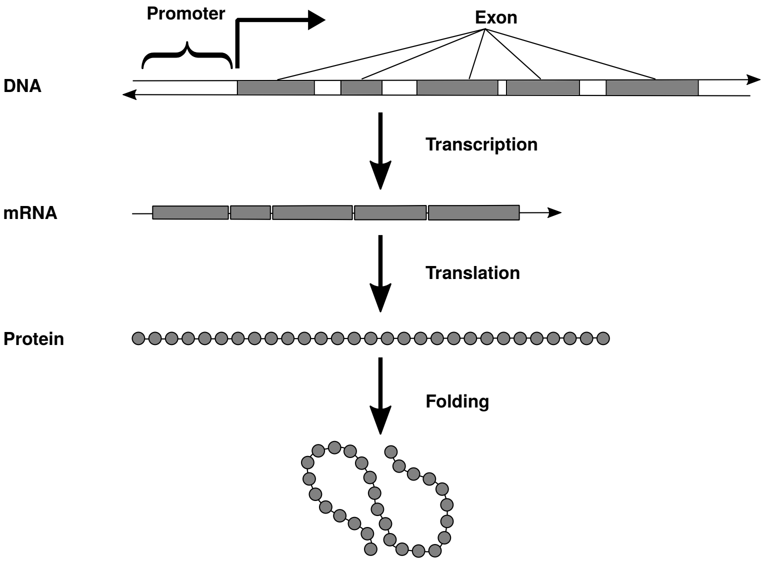


Figure 2.1: The information flow between DNA, RNA, and proteins. Binding of TFs to the promoter area of a gene can help recruit the RNA polymerase to the transcription start site (marked with the arrow with an angle), which starts transcribing the gene into a message RNA (mRNA). Afterwards, a ribosome can translate the mRNA into an amino acid sequence, which folds into a 3-dimensional structure of the final protein.

The human *genome* consists of DNA of total length approximately 3 billion base pairs (bp), which is organized into 23 DNA molecules manifesting as *chromosomes.* Each cell in the human body, with the exception of sperm and egg cells, contains two copies of this genome, one from the paternal side and the other from the maternal side.

Each protein is coded by a piece of DNA called *gene.* A current estimate for the number of genes in humans is as low as 19 000 [25]. These genes cover only about 1–2% of the human genome. The fact that the number of different proteins in humans is much higher than 19 000, is explained by a process called *alternative splicing*, which can assemble the DNA of a gene in multiple ways to produce different messenger RNAs, and hence different proteins. The pieces of DNA of a gene, which are available to use in this assembly are called *exons*, whereas the non-coding parts between exons are called *introns.*

Even though each cell in the human body contains the same genome, different cells can look and function very differently from each other. For instance, a liver cell has completely different shape and size than a neuron. This diversity is due to different genes being *expressed* in different cell types, that is, different sets of genes are used for producing proteins. A typical cell expresses 30–60% of the total number of genes. In addition to the cell type, the stage of development of an individual (e.g., embryo, fetus, child, adult) affects the gene expression, as do various signals between cells and changes in the environment. For example, expression of heat shock proteins is increased during stressful conditions to the cell, such as increase in temperature.

There are several mechanisms that a cell can use to control the expression of genes: control of transcription, epigenetic mechanisms, RNA processing control, transportation and placement of RNA within the cell, degradation of RNA, control of translation, and post-translational control. These happen at different phases of the information flow from DNA through RNA to protein. Of these mechanisms, the control of transcription of DNA to messenger RNA is the most important, and it is also the one that is considered in this thesis. This control can both up-regulate the expression of a gene, so that more proteins from that gene are produced, or in the case of down-regulation fewer or zero proteins are produced.

The transcription is mainly regulated by proteins called *transcription factors* (TFs) that can affect the transcription of its target gene by chemically binding to areas of DNA related to the target gene. Typically these *regulatory areas* are outside the coding region of the gene, but can still be quite close. For example, a *promoter* is a stretch of DNA just before the transcription

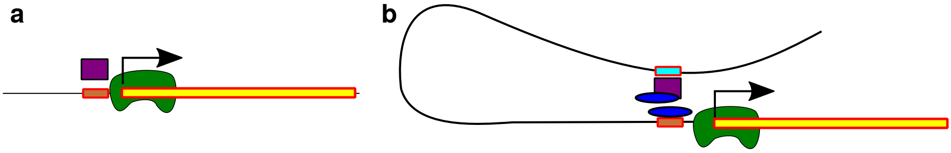


Figure 2.2: Schematic of regulatory areas. The following colour coding is used: yellow for gene, brown for promoter, cyan for enhancer, green for RNA polymerase, purple for the TF under consideration, blue for other proteins. The direction and start site of transcription is indicated by the black arrow. Binding of the transcription factor to the regulatory area recruits the RNA polymerase, which does the transcription of the gene. (a) The TF binds to the promoter area. (b) The TF binds to the enhancer area. Note that DNA bending is required for the enhancer to reach proximity with the transcription start site.

start site of the target gene. Typically the promoter is contained in a region of 1000 bp before the transcription start site. Several TFs can be bound to this area. The short stretch of DNA a single TF is bound to is called a *binding site*. These typically have a length of about 6–12 bp [41]. For example, the transcription factor GATA1 can bind to sequence TGATAG [1]. The nucleotide content of the binding site affects the chemical binding energy between the DNA and the transcription factor. In other words, the binding sites of TFs are sequence-specific.

A classic example of regulation of transcription is the production of trypthophan in bacterium *E. coli* [66]. When the transcription factor called tryptophan repressor binds to the promoter of a group of genes called tryptophan operon, it prevents the transcription of tryptophan operon, and subsequently stops the production of tryptophan.

Another example of a regulatory area is called *enhancer*. Unlike promoters, an enhancer can be even a million bps away from the gene it controls [3]. However, as the DNA molecules are very densely packed, like coils of thread, the enhancer can be close to the gene in the 3D distance. Figure 2.2 illustrates the relation of regulatory areas with respect to the target gene. Enhancers are involved in up-regulation of its target gene; the down-regulating counterparts of enhancers are called *silencers*. It has been estimated that the human genome contains hundreds of thousands of regulatory regions [15]. Therefore, these can explain a fraction of the genome outside the coding regions, whose function has so far been largely unknown.

As the transcription factor binding sites are relatively short, it would seem that they should appear in the genome very frequently, purely by chance. If we assume that each nucleotide position in the genome is uniformly and independently distributed, then we expect, for example, the site TGATAG to appear in the genome approximately $2 \cdot 3 \cdot 10^9/4^6 \approx 1\ 500\ 000$ times. However, in practice not all of these are possible binding sites. There are other mechanisms that affect the binding of a TF.

The dense packing of the genome allows the DNA molecules that otherwise would have a total length of about two meters, to fit in the nucleus of a cell, which has an average diameter of 6 micrometers. This condensing is enabled by "packing proteins" that bind to DNA and coil it into a small space. The complex formed by DNA and the packing proteins bound to it is called *chromatin*. This condensing of DNA is not uniform along the genome, and it is not static either: it can depend on the type and cell-division phase of the cell. The loosely condensed regions of the chromatin are called *euchromatin* and the more densely packed regions are called *heterochromatin*, although in practice there can be several classes of densities instead of just two clearly separate classes. The euchromatin allows transcription factors to easily access the regulatory areas, which is why it is also called open chromatin. Heterochromatin, on the other hand, makes it hard for a transcription factor to access the DNA, because of the dense packing. So, the openness of different areas of the genome further restricts the putative binding sites. Since the openness of DNA can be of a dynamic nature, this provides another mechanism for regulating the transcription of genes.

Another mechanism that modifies the specificity of TF binding is co-operation of two or more TFs. The binding energy when two factors bind together to DNA can be higher than the sum of binding energies when the two TFs bind to DNA independently. When two TFs bind DNA co-operatively it is called *dimeric binding*. There are basically two main ways for transcription factors to co-operate in binding: the TFs can first bind to each other to form a dimeric complex, which subsequently binds DNA; or, initially one TF can bind to DNA, which subsequently makes it possible for the other TF to bind to DNA as well. As the dimeric binding sites are longer, they also become more restrictive. Hence, one expects them to appear less frequently in the genome. We will study the discovery of dimeric binding sites later in Chapters 3 and 5.

TFs, like other proteins, consist of subunits called *domains* that can fold into 3-dimensional structures independently of each other [1]. A *DNA-binding domain* (DBD) is a domain responsible for binding to DNA. The DBDs can be divided into structural families based on the amino acid sequence similarity [82]. The binding sites of two TFs from the same family are often very similar, but the differences can manifest themselves, for example, as different dimeric binding sites these TFs make [34].

As humans have approximately only 1600 transcription factors [41], the co-operation between TFs allows more refined control of gene expression. In addition, as transcription factors are proteins and hence have genes that encode them, the production of TFs can be regulated as well, possibly by a different TF. So, transcription factors form complex regulatory networks with interdependencies and feedbacks.

Chapter 3

Representation of motifs

A *motif* is a reoccurring pattern in biological sequential data such as nucleotide or amino acid sequences. A motif can have different *instances*, that are subsequences occurring in the data. In this thesis we consider only nucleotide motifs of the binding sites of transcription factors. In addition, we assume that each motif has a fixed length, that is, all the motif instances have the same length, say ℓ. In this chapter, we describe two ways of representing these sequence motifs: as a set of sequences and as a probabilistic model. At the end of this chapter we introduce a way of representing the relative binding preferences among the dimeric binding modes of transcription factors.

3.1 Sets of sequences

Some DNA-binding proteins are very strict about which DNA sites they choose to bind. For example, the restriction endonuclease EcoRI binds to site GAATTC only [50]. If there are more than one preferred sites, then one might list them all. However, as the diversity usually occurs in certain positions in the binding sites, a more compact representation can be used. A *consensus sequence* specifies one or more preferred sequences as (simplified) regular expressions or as IUPAC sequences [56]. For example, the set of sequences GTCACA, GTCGCA, GTTACA, GTTGCA can be represented as a regular expression GT[CT][AG]CA or an IUPAC sequence GTYRCA, where the IUPAC symbol Y corresponds to C or T, and the symbol R corresponds to A or G. For example, the PRODORIC2 database of prokaryotic TFs [19] includes a consensus sequence for each factor.

	1	2	3	4	5	6	7	8	9	10
A	0.59	0.07	0.06	0.00	0.00	0.99	0.79	0.43	0.03	0.24
C	0.05	0.80	0.93	0.00	0.00	0.01	0.00	0.01	0.24	0.18
G	0.27	0.11	0.00	0.99	1.00	0.00	0.00	0.56	0.04	0.44
T	0.09	0.02	0.00	0.00	0.00	0.00	0.20	0.00	0.69	0.13

Table 3.1: PPM model of monomeric FLI1 binding site. Each column in this matrix defines a probability distribution over nucleotides for the corresponding position in the motif.

3.2 Probabilistic models

Previously consensus sequences were used to represent binding sites also because the experimental methods produced only very few example sequences of motif instances, and therefore the need for more complicated representations was not realized. Now, however, due to the advent of high-throughput methods, the data is plentiful and a more continuous variation in the binding sites is detected. Next we review a few motif representations based on the probability of a sequence being a binding site.

3.2.1 Position-specific Probability Matrix

Position-specific Probability Matrix (PPM) [67] defines a probability for each nucleotide sequence of fixed length, say ℓ. The probability of a nucleotide in one position is independent of the nucleotides in other positions in this model. Hence, the model can be compactly defined as a product of ℓ categorical distributions. The parameters of this model can be represented as a matrix θ as follows. PPM θ is a $4 \times \ell$ matrix

$$\theta = \begin{bmatrix} \theta^{\mathrm{A},1} & \theta^{\mathrm{A},2} & \dots & \theta^{\mathrm{A},\ell} \\ \theta^{\mathrm{C},1} & \theta^{\mathrm{C},2} & \dots & \theta^{\mathrm{C},\ell} \\ \theta^{\mathrm{G},1} & \theta^{\mathrm{G},2} & \dots & \theta^{\mathrm{G},\ell} \\ \theta^{\mathrm{T},1} & \theta^{\mathrm{T},2} & \dots & \theta^{\mathrm{T},\ell} \end{bmatrix},$$

where $\theta^{a,h} := \theta[a,h]$ gives the probability for a symbol (nucleotide) a from alphabet $\Sigma = \{\mathrm{A},\mathrm{C},\mathrm{G},\mathrm{T}\}$ to occur in position h of θ, and ℓ denotes the length of θ. A real example of a PPM model of the TF FLI1 is shown in Table 3.1. For every sequence $X = X_1X_2\cdots X_\ell$ of length ℓ, the PPM model θ gives the probability $P(X) = \prod_{1\le h\le \ell} P(X_h) = \prod_{1\le h\le \ell} \theta^{X_h,h}$.

If we sample N sequences from the product distribution θ, then the number of nucleotides in the position h is multinomially distributed with parameters N and $\theta^{\cdot,h}$. If the sample size N is large enough, then the *maximum likelihood estimates*

$$\theta^{a,h}_{\mathrm{MLE}} = n_{a,h}/N,$$

for each $a \in \Sigma$, where $n_{a,h}$ gives the number of nucleotide a in position h of the sequences, give the most likely explanation of the nucleotide in the position. Hence, with a large enough unbiased sample of binding sites of a transcription factor, we can use maximum likelihood estimates for each position h to get an accurate PPM model for the data, assuming the positions in the binding sites really are independent. We call this PPM model learning algorithm the standard alignment method. The matrix $n_{a,h}$ of counts mentioned above is sometimes called the *Position-specific Frequency Matrix* (PFM).

If, however, the data is scarce, then the maximum likelihood estimates may give non-optimal models. For instance, if the sample contains only ten sequences, and in the first position we have nucleotide counts $n_{\mathrm{A},1} = n_{\mathrm{C},1} = 5$ and $n_{\mathrm{G},1} = n_{\mathrm{T},1} = 0$, then the estimated model would give probability zero for all sequences starting with either G or T. The sample size ten is too small to infer that the probabilities are zero for these two nucleotides. If we have no *a priori* information that a nucleotide is impossible in some position, then we can try to correct this small sample error by adding a pseudo-count for each nucleotide. A common pseudo-count method, called *Laplace's rule of succession*, is to add one to each of the nucleotide counts.

Although using pseudo-counts may seem like an ad hoc solution, it can be theoretically supported using the Bayesian interpretation of probability. To this end, let us fix a position $1 \leq h \leq \ell$, and assume that the parameters of the multinomial distribution in position h are *a priori* distributed according to a Dirichlet distribution with parameters $\alpha = (\alpha_a)_{a\in\Sigma}$, that is, $\theta^{\cdot,h} \sim \mathrm{Dir}(\alpha)$. Then it can be shown (see, for example, [18]) that the posterior distribution of the parameters is also Dirichlet, but with different parameters: $\theta^{\cdot,h}|(n_{\mathrm{A}}, n_{\mathrm{C}}, n_{\mathrm{G}}, n_{\mathrm{T}}) \sim \mathrm{Dir}((n_{\mathrm{A}}, n_{\mathrm{C}}, n_{\mathrm{G}}, n_{\mathrm{T}}) + \alpha)$. In addition the *posterior mean estimator*

$$\theta^{\cdot,h}_{\mathrm{PME}} := \int \theta^{\cdot,h} \frac{P(n_{\mathrm{A}}, n_{\mathrm{C}}, n_{\mathrm{G}}, n_{\mathrm{T}}|\theta)P(\theta^{\cdot,h}|\alpha)}{P(n_{\mathrm{A}}, n_{\mathrm{C}}, n_{\mathrm{G}}, n_{\mathrm{T}})} d\theta^{\cdot,h}$$

is in fact

$$\frac{n_a + \alpha_a}{N + A},$$

where $A = \sum_{b\in\Sigma} \alpha_b$ and $a \in \Sigma$. So, the pseudo-counts can be thought of as parameters α_a of the prior Dirichlet distribution.

3.2.2 Position-specific Weight Matrix

Another way to represent binding motifs is the *Position-specific Weight Matrix* (PWM) [77]. The PWM M is an array with shape $4 \times \ell$ like the PPM, but, instead of probabilities, the values $M^{a,h}$ in the array are arbitrary real numbers called *weights*. All the positions are again assumed independent and the total weight of a sequence X is $\sum_{1\leq h\leq \ell} M^{X_h,h}$. Note that the contributions of each position are now added together instead of multiplying.

One common way of defining a PWM is through a PPM θ and a background model $\theta_0 = (\theta_0^{\mathrm{A}}, \theta_0^{\mathrm{C}}, \theta_0^{\mathrm{G}}, \theta_0^{\mathrm{T}})$, a categorical distribution. The background model can, for example, give the nucleotide distribution of the genome of an organism under consideration. The weights are now defined by $M^{a,h} := \log_2 \frac{\theta^{a,h}}{\theta_0^a}$ for each $a \in \Sigma$ and $1 \leq h \leq \ell$. Sometimes these weights are also called (log-ratio) *scores*.

Another example is the energy PWM H, whose elements give the contribution of each nucleotide in each position of the site to the total binding energy between the DNA and the TF. The contribution of each DNA position is again assumed independent of the other positions. In this thesis we will not give methods for learning the energy PWMs, but we note that according to Heumann et al [30], the PWM M defined above approximates the energy PWM H. This connection establishes that our probability models learned from count-based sequence data are in accordance with the energy model. Some references to methods which learn energy models are given later in Chapter 5.

3.2.3 Adjacent Dinucleotide Model

The *Adjacent Dinucleotide Model* (ADM) [76] is defined as the inhomogeneous Markov chain (of order 1). It has the *Markov property* which tells that the probability of a nucleotide b in position h is independent of the nucleotides in positions $h' < h-1$ on the condition that the nucleotide in position $h-1$ is given. An ADM can be represented as a matrix θ with shape $16 \times \ell$ whose elements $\theta^{ab,h}, a, b \in \Sigma, 1 \leq h \leq \ell$ are the transition probabilities $P(X_h = b | X_{h-1} = a)$. The probability of a sequence $X = X_1 X_2 \cdots X_\ell$ given by the ADM model θ is

$$P(X) = \prod_{1\leq h\leq \ell} P(X_h | X_{h-1}) = \prod_{1\leq h\leq \ell} \theta^{X_{h-1}X_h,h}.$$

Note that we define X_0 to be A, so that the initial probabilities $P(X_1 = b) = P(X_1 = b | X_0 = \mathrm{A}) = \theta^{\mathrm{A}X_1,1}$ get treated symmetrically with the transition probabilities. With the ADM models we use notation $\theta^{b,h} := P(X_h = b)$ for the probability of symbol b in position h, that is,

$$\theta^{b,h} = \sum_{a_1,\ldots,a_{h-1}\in\Sigma} \theta^{\mathrm{A}a_1,1}\theta^{a_1a_2,2}\cdots\theta^{a_{h-1}b,h}$$

for $b \in \Sigma, 1 \leq h \leq \ell$.

Let us fix a position $1 \leq h \leq \ell$ and a nucleotide $a \in \Sigma$. If we sample N sequences from the ADM distribution θ, then the number of nucleotides in the position h on the condition that $P(X_{h-1} = a)$ is multinomially distributed with parameters $n_{a,h-1}$ and $\theta^{a\cdot,h}$. If the sample size N is large enough, then the maximum likelihood estimates

$$\theta^{ab,h}_{\mathrm{MLE}} = n_{ab,h}/n_{a,h-1},$$

for each $b \in \Sigma$, where the $n_{ab,h}$ gives the number of dinucleotide ab ending in position h of the sequences, give the most likely explanation of the nucleotides in the position h on the condition that $P(X_{h-1} = a)$.

3.2.4 Higher order models

Assume $X = X_1X_2\cdots X_\ell$ is a random vector with $X_h \in \Sigma$, for each $1 \leq h \leq \ell$. The probability of X can be written as

$$P(X) = P(X_1)P(X_2|X_1)\cdots P(X_h|X_1X_2\cdots X_{h-1})\cdots P(X_\ell|X_1X_2\cdots X_{\ell-1})$$

by the chain rule. If we further assume that X is a Markov chain of order k, then we can write

$$P(X) = P(X_1)P(X_2|X_1)\cdots P(X_h|X_{h-k}\cdots X_{h-1})\cdots P(X_\ell|X_{\ell-k}\cdots X_{\ell-1}).$$

To be more specific, this is an *inhomogeneous Markov chain of order k*, since the transition probability matrix is allowed to be different at each position of the motif. The sequence $X_{h-k}\cdots X_{h-1}$ is the *context* of position h, for each $1 \leq h \leq \ell$. Using higher order Markov chains allows more precise modeling of the dependencies between positions, but the downside is that more data is needed to learn more complex models. This is because for a Markov chain of order k the number of parameters that need to be learned is $\sum_{h=1}^{\ell} 3\cdot 4^{\min(h-1,k)}$. So, the number of parameters grows exponentially with the order k. In addition, there is a risk of overfitting the model, and therefore the model might not be usable for prediction of new binding sites. Siebert and Söding have, however, developed an adaptive form of higher order Markov chains [72], where the lower order models function as priors to higher order models. This should in principle guard against overfitting.

Eggeling et al [20, 21] provide a model called inhomogeneous Parsimonious Markov Model (iPPM), which enables another way of using higher order Markov models without overfitting. It is based on parsimonious context trees [11], which allow varying context lengths. Each context should be as short as possible.

Another possible extension of the ADM model is to allow dependencies between any pair of positions, not only the adjacent ones. Recently, Omidi et al [55] proposed a model called Dinucleotide Weight Tensor (DWT) for this purpose.

3.2.5 Relative entropy

Relative entropy, also known as the *Kullback–Leibler divergence*, between two discrete distributions p and q is defined by $D(p||q) = \sum_i p_i \log_2 \frac{p_i}{q_i}$. Gibbs' inequality tells us that $D(p||q) \geq 0$, where the equality holds if and only if $p = q$. Relative entropy gives us a way to compare two distributions. In terms of information theory, relative entropy $D(p||q)$ tells the extra amount of information (in bits) that needs to be transmitted, if a message is coded using code optimal to distribution q instead of the correct distribution p. Note that relative entropy is not symmetric with respect to p and q.

If the distributions are over the set of all sequences of length ℓ, that is, $p_X = P_{\theta_1}(X)$ and $q_X = P_{\theta_2}(X)$ for $X \in \Sigma^\ell$, then it can be shown that the relative entropy between distributions p and q, defined by PPM models θ_1 and θ_2, decomposes as

$$D(p||q) = \sum_{h=1}^{\ell} \sum_{a \in \Sigma} \theta_1^{a,h} \log_2 \frac{\theta_1^{a,h}}{\theta_2^{a,h}}.$$

Because of the dependencies between columns in ADM models, the decomposition becomes slightly more complex:

$$D(p||q) = \sum_{h=1}^{\ell} \sum_{a \in \Sigma} P_{\theta_1}(X_{h-1} = a) \sum_{b \in \Sigma} \theta_1^{ab,h} \log_2 \frac{\theta_1^{ab,h}}{\theta_2^{ab,h}}.$$

3.2.6 Information content

If p is a distribution over the set of sequences Σ^ℓ, defined by a PPM motif θ, then the *information content* [67] of the distribution is

$$IC(p) = \sum_{h=1}^{\ell} \sum_{a \in \Sigma} \theta^{a,h} \log_2 \frac{\theta^{a,h}}{0.25} = 2\ell + \sum_{h=1}^{\ell} \sum_{a \in \Sigma} \theta^{a,h} \log_2 \theta^{a,h}.$$

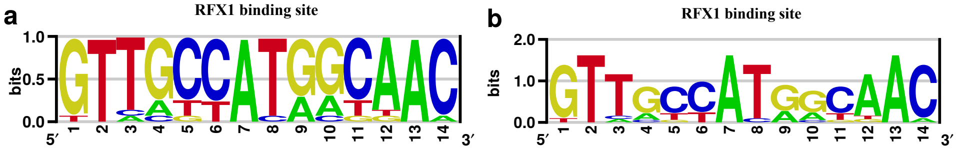


Figure 3.1: Sequence logo for factor RFX1. (a) The letter heights directly show the probabilities of nucleotides. (b) The letter heights are scaled by the information content of the corresponding column. Logos have been produced using enoLOGOS [85].

Similarly, we define the information content for ADMs with

$$\begin{aligned} IC(p) &= \sum_{h=1}^{\ell} \sum_{a\in\Sigma} P_\theta(X_{h-1} = a) \sum_{b\in\Sigma} \theta^{ab,h} \log_2 \frac{\theta^{ab,h}}{0.25} \\ &= 2\ell + \sum_{h=1}^{\ell} \sum_{a\in\Sigma} \theta^{a,h-1} \sum_{b\in\Sigma} \theta^{ab,h} \log_2 \theta^{ab,h}. \end{aligned} \tag{3.1}$$

Since information content is actually the relative entropy between p and the uniform distribution, we immediately see that information content is non-negative. It is also clear by the definition that $IC(p)$ is at most 2ℓ. Information content can be thought to measure the specificity of the TF, that is, how strict it is about the nucleotide content in the binding site.

3.2.7 Visualization

The PPM logo can be visualized as a *sequence logo* [68] as shown in Figure 3.1. In Figure 3.1a the height of each letter is proportional to the probability of the corresponding base. In addition, the nucleotides in each position are ordered according to the probability with the most probable nucleotide at the top and the least probable at the bottom. If we instead scale the probabilities in each column by its information content, we get an alternative visualization of the same model, shown in Figure 3.1b.

In Figure 3.2 an example ADM motif is visualized as a *river-lake logo*. The visualization is a slight modification of the one used in Morgunova et al [52].

3.2.8 Distance between models

In order to assess the success of learning of motif models, it is useful to have a concept of distance between the models. If we have a "ground truth"

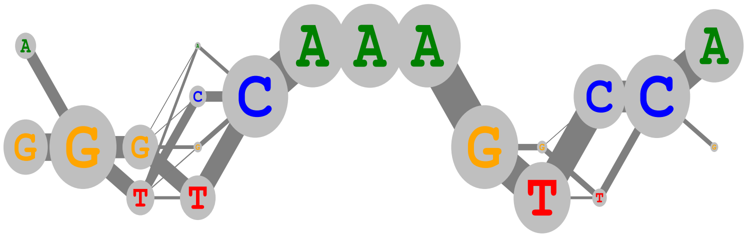


Figure 3.2: Visualization of an ADM motif. The radius of circles reflects the probability of reaching the corresponding state, and the thickness of the edges are proportional to the probabilities of the corresponding dinucleotides. A circle and the edges leaving it (in left-to-right direction) are not drawn, if the probability of the state is less than 0.05. This river-lake logo for factor HNF4A is from Paper III.

of what the target model should be, then our learning algorithm should produce a model which is close to the ground truth model. And even if we do not know the ground truth, the distance measure allows us to reason whether an iterative learning method converges to a model or not. Next we present a few commonly used distance measures for motif models.

Max norm

The max norm distance between two discrete distributions p and q is defined by $d(p,q) = \max_i |p_i - q_i|$. This is an intuitive distance, but is not a very natural distance for distributions. For PPMs θ_1 and θ_2 we define the max norm distance with

$$d(\theta_1, \theta_2) = \max_{a \in \Sigma, 1 \le h \le \ell} |\theta_1^{a,h} - \theta_2^{a,h}|,$$

and for ADMs θ_1 and θ_2 as a distance between dinucleotide probabilities:

$$d(\theta_1, \theta_2) = \max_{a,b \in \Sigma, 1 \le h \le \ell} |\theta_1^{a,h-1}\theta_1^{ab,h} - \theta_2^{a,h-1}\theta_2^{ab,h}|.$$

Symmetric Kullback–Leibler divergence

The *symmetric Kullback–Leibler divergence* is defined as $D_{symmetric}(p,q) = D(p||q) + D(q||p)$. This measure is symmetric, but the triangle-inequality does not hold for it. The square root of the *Jensen–Shannon divergence*

$$\sqrt{D_{JS}(p,q)} := \sqrt{\frac{1}{2}D(p||r) + \frac{1}{2}D(r||q)},$$

where distribution $r := 0.5p + 0.5q$, satisfies all three requirements of a distance measure, for proof see [22].

Total variation distance

The *total variation distance* between two discrete distributions is defined by $\delta(p,q) = \frac{1}{2}\sum_i |p_i - q_i|$. This distance is automatically symmetric and suitable for measuring the distance between distributions. The total variation distance works trivially for both PPM and ADM motifs by defining $p_X = P_{\theta_1}(X)$ and $q_X = P_{\theta_2}(X)$. However, the computation of the total variation distance between two wide models is computationally heavy.

3.3 Co-Operative Binding model

In this section we introduce a representation for the strength of different dimeric binding cases.

As we mentioned in Chapter 2, the binding energies of two transcription factors might not be additive. It seems plausible that as the distance between the two binding sites increases, the probability of co-operative binding should decrease. To analyze the variation of the probability of dimeric binding as the function of the distance between the binding sites, we systematically measure the relative preference of each distance and present these values in array form. We define the distance between sites as the number of bases in the gap between the sites. We allow the distance to be negative, that is, the binding sites are allowed to overlap.

In addition to the distance between the binding sites, also the relative orientation of the two TFs affects the strength of the binding. If we assume that the TF molecule is not fully symmetric, then the TF bound to DNA has a direction. In case of two TFs bound to DNA, the TFs and their binding sites can either have the same direction or opposite. If we further assume that the two TFs are different, then the TFs can, in addition, be ordered in two ways. This gives in total four possible relative orientations of the binding sites, which we name Head-to-Tail (HT), Head-to-Head (HH), Tail-to-Tail (TT), and Tail-to-Head (TH). These relative orientations are illustrated in Table 3.2. If we have two copies of the same TF, then only three orientations are possible, as the orientations HT and TH indicate the same case. We refer to the dimer formed by different TFs as *heterodimer*, and the dimer formed by copies of the same TF as *homodimer*. In Figure 3.3 four heterodimeric cases are visualized.

For each TF pair, distance and orientation we assign a probability. This is the probability of a (short) sequence, known to have likely been bound by

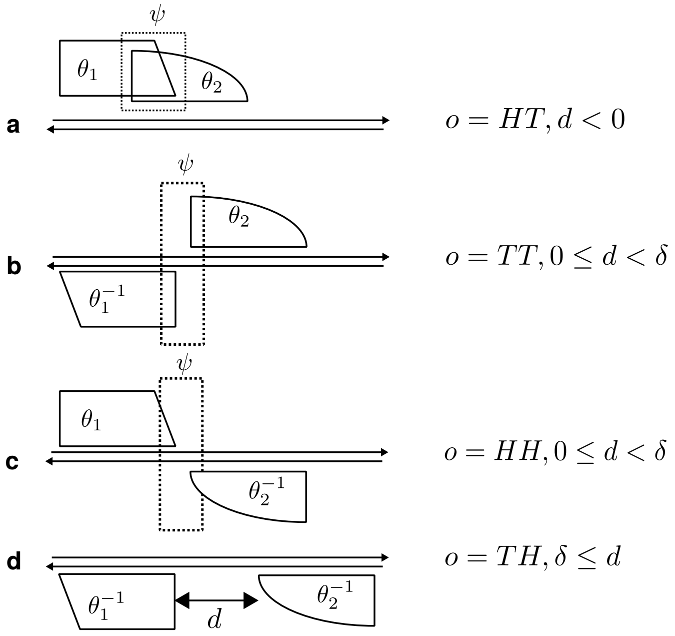
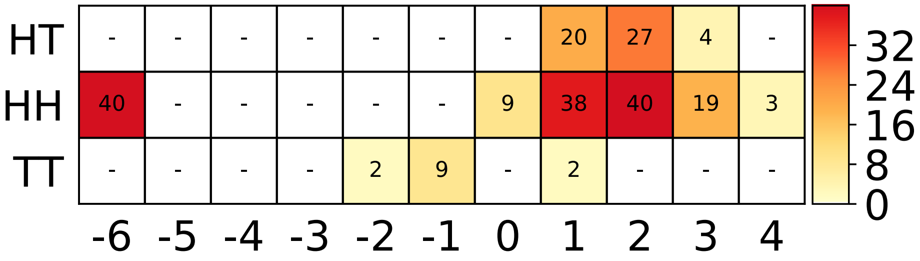


Figure 3.3: Four different heterodimeric cases with distance d and relative orientation o. The threshold δ is the minimum distance when the two binding sites are considered independent. Exponent -1 denotes taking the reverse complement of the binding model. Box ψ indicates the region of the model in which co-operative effects are anticipated to occur.

Table 3.2: Relative orientation of two motif occurrences within a dimer.

Orientation o	Short-hand	
Head-to-Tail	HT	$\rightarrow\rightarrow$
Head-to-Head	HH	$\rightarrow\leftarrow$
Tail-to-Tail	TT	$\leftarrow\rightarrow$
Tail-to-Head	TH	$\leftarrow\leftarrow$



	-6	-5	-4	-3	-2	-1	0	1	2	3	4
HT	-	-	-	-	-	-	-	20	27	4	-
HH	40	-	-	-	-	-	9	38	40	19	3
TT	-	-	-	-	2	9	-	2	-	-	-

Figure 3.4: An example COB table. The rows correspond to orientations o and the columns correspond to distances d between the binding sites, measured as the number of positions between the sites. Negative distance specifies that the sites overlap. The individual values are probabilities of the corresponding dimeric case. The sum of values in the COB table is at most 1. Values below a threshold (by default 0.001) are denoted by a dash character. As this COB table is for a homodimer, only three orientations are possible (by symmetry, orientation HT equals orientation TH).

both factors, to contain exactly this dimeric binding site. These probabilities are represented as *Co-Operative Binding tables* (COB). Formally, a COB table between two factors, θ_{k_1} and θ_{k_2}, is a two-dimensional array $(\lambda_{k_1 k_2 od})$, whose rows are orientations o, columns are distances d, and values are in the interval $[0, 1]$. As the individual values in the array are probabilities, their sum cannot exceed 1. A visualization of an example homodimeric COB table is shown in Figure 3.4. In this thesis we do not consider more complicated binding modes, such as trimers that consist of co-operative binding of three TFs, as the number of interaction models (generalization of COB table) increases exponentially with the number of TFs taking part in the interaction model. We instead assume that a sequence contains either zero, one, or two binding sites.

We have earlier [34] considered a COB table that used the concepts of distance and orientation, but whose entries, instead of probabilities, were log-ratios that measured the over- or under-representation of each dimeric case comparing the number of occurrences observed in a data set against the number of expected occurrences in a background model. There were mainly three problems with this approach. Firstly, an arbitrary threshold for binding score needed to be set in order for us to be able to count the number of occurrences. Secondly, the approach was not flexible enough to handle overlapping dimers well. Thirdly, it did not allow comparison of strengths of monomeric and dimeric binding modes, because it did not form a total probability model. All these problems are handled by our new method, called MODER, that is described in Chapter 5.

Multiple methods have been proposed to discover binding motifs that comprise two parts, that is, dimeric motifs. For instance Bioprospector [46] allowed variable spacing between two half-sites. This enabled finding the motifs for the two half-sites from cases where both factors are bound within a short gap range (for example 1–4 bp). From the predicted binding sites in the input sequences, the observed gap lengths can be listed, but as at that time the number of sequences was typically less than 100, conclusions about preferred distances and possible interactions between the two factors do not have much evidence. Although Bioprospector allows detection of palindromic homodimers of the same TF, different relative orientations are not considered in general.

Bipad [9] and its improved version MaskMinent[47] also allow discovery of motifs consisting of two parts, in addition to contiguous motifs. The distance between the two parts can vary, and all four relative orientations are considered, but the distribution of orientations or the joint distribution of orientations and distances are not discussed in the articles. The distributions of distances are visualized as histograms.

After the introduction of high-throughput TF binding experiments, which provide thousands of sequences expected to be bound by a TF, many new methods appeared that considered the over-representation of motif occurrence pairs. Of these coMOTIF [86] is closest to our method. Both use a total probability model, which allows zero, one, or two motif occurrence per sequence. coMOTIF does not explicitly give the distribution of distances between the two binding sites, but it does consider the distribution of the number of occurrences per sequence and the four relative orientations in the case of two occurrences. The method does not allow consideration of homodimers.

SpaMo [84], iTFs [38], and TACO [33] consider different distances between occurrences and all four relative orientations. Of these only TACO allows overlapping dimers. None of the three methods propose a standard representation for the distribution of different orientation and distance combinations. SpaMo shows the frequencies of distances as a histogram for each relative orientation. iTFs bins distances into the following categories: 0–10 bp, 10–25 bp, 25–50 bp, and 50–100 bp. For each factor pair the *p*-value of the combination of distance category and relative orientation is listed. TACO visualizes for a motif pair the *p*-values as a function of distance as a heatmap.

Chapter 4

Applying motif models

In this chapter we will briefly give three application areas for binding models.

4.1 Scanning genomes for putative binding sites

One application of the binding models of TFs is to scan the whole genome of an organism to find putative binding sites. Clustering of binding sites may indicate a regulatory area of a gene. A threshold for the score must be defined so that a locus whose score by the binding model is equal or above this threshold is declared as a binding site. This threshold is often based on a p-value threshold. For example, consider a p-value threshold 0.001 and a background model θ_0. For the significance threshold 0.001 there corresponds a score threshold T such that the probability of a sequence distributed according to the background model getting score at least T is (about) 0.001. For example the MOODS software [39, 40] uses this method to efficiently locate putative binding sites from a genome. FIMO is another example of a binding site searching software [28]. Instead of p-values, it uses q-values [75] to decide the score threshold in order to handle multiple hypothesis testing. Note that when scanning the human genome for binding sites of length ℓ, the number of tests performed is very large; roughly equal to the size of the genome $3 \cdot 10^9$. So, the multiple hypothesis testing correction, such as the Bonferroni correction, where the significance threshold is divided by the number of the tests, cannot be neglected. The Bonferroni correction may however be too conservative as the probabilities of overlapping binding sites are not independent.

The well known genome browser Ensembl [89] uses MOODS and sets of published PPMs [35, 36, 53] to annotate the human genome with the putative TF binding sites [23, 88].

High-affinity binding sites of a TF can also appear in a genome just by chance. Therefore, to avoid false positives, it may be useful to restrict the search to the parts of genome that contain open chromatin. These cell type and development phase-dependent areas of a genome can be obtained, for example, using the DNase-seq experiment [12].

4.2 Classifying factors using motif similarity

Binding models have also been used to create a network representation of their similarity. Jolma et al [34] produced a set of PPM models for 411 TFs (human and mouse) and using a similarity measure for the models a network representation was created. Since many of the models were very similar to each other, a representative model was chosen among the similar ones. This resulted in a dominating set of 239 representative PPMs. This offers an alternative view to the set of TFs, complementing the ones obtained through protein sequence similarity or 3D structure similarity. The subnetworks in the representation seemed to be in accordance with the established TF families.

To measure the divergence of binding specificities of TFs between human, mouse, and fruit fly, Nitta et al [53] obtained accurate binding models for orthologous TFs of the three species using identical methods. Using a motif similarity measure, the divergence of the binding sites was noted to be surprisingly small.

4.3 Predicting the effect of mutations in binding sites on binding strength

Single Nucleotide Polymorphism (SNP) is the most common type of genetic variation in human genome [14]. The 1000 Genomes Project [14] listed 84.7 million SNPs. Only a fraction of these are functional. Besides SNPs in coding regions, the regulatory regions can also contain SNPs that can affect the regulation of genes, called regulatory SNPs (rSNPs). Since testing the function of rSNPs experimentally is expensive, many computational methods have been developed to predict them [4, 48, 63, 78, 92]. Most of them test whether an SNP has a significant effect on the binding affinity of a TF.

As an example of how different variants of a regulatory SNP can affect the development of a disease, two studies [58, 81] have shown that the G-variant in rSNP rs6983267 can increase the risk of colorectal cancer. Each of the studies describe a different mechanism for how the risk variant can lead to the disease by changing the binding probability of different TFs.

Chapter 5

Learning models

In this chapter we go through the main results of this thesis, namely, the learning algorithms of monomeric and dimeric motifs for both 0th and 1st order Markov chains, and the COB tables. In Section 5.1 we briefly describe some biological experiments that can provide data for learning algorithms. In Section 5.2 we introduce the combinatorial SeedHam method, and in Section 5.3 we introduce the EM-based learning algorithm called MODER. Learning of COB tables is discussed in Section 5.4, and Section 5.5 is about learning ADM models. In Section 5.6 some other approaches that have previously been used to learn binding models are reviewed. Section 5.7 discusses the evaluation of the goodness of the learned models.

5.1 Experimental data for learning binding models

Molecular biological experiments can be divided into two classes: *in vitro* and *in vivo* methods. In vitro experiments study some feature in isolation, whereas in vivo experiments study a feature as part of a cell line, a living tissue or an organism. The data, from which one can try to learn binding models, can be obtained from several different biological experiments. In vitro methods consider binding of TFs to a set of short DNA sequences, called *oligonucleotides*. In the case of in vivo methods, however, the binding of TFs to DNA happens to full length chromosomes, which are folded into packed chromatin, in the presence of other proteins that a living cell contains. Since the structure of chromatin and the set of proteins contained in a cell depends on its cell and tissue type, in vivo methods must be performed separately for each tissue type and cell line. In the rest of this section we briefly cover two in vitro methods and one in vivo method.

5.1.1 In vitro methods

SELEX is an in vitro method that was originally introduced in the beginning of the 1990s [54, 80], and was later converted into a high-throughput method [36, 91]. The HT-SELEX experiment starts with a large set of fixed length (typically 14–40 bp) random DNA sequences. Then copies of a TF are added, and those oligonucleotides that were bound by the TF are selected. The selected oligonucleotides are amplified using PCR (Polymerase Chain Reaction), and a fraction of the oligonucleotides are sequenced, while for the rest the procedure is repeated. Several rounds can be performed, and from each round a sample is sequenced. In each round the relative abundances of different sequences are enriched according to the affinity of the binding sites of the TF they contain. HT-SELEX can result in hundreds of thousands or even millions of sequences per round. CAP-SELEX [35] is a modified version of SELEX, which allows selection of oligonucleotides that were simultaneously bound by two different TFs. This gives information about the heterodimeric binding sites.

Protein Binding Microarray (PBM) contains oligonucleotides of length L attached from the other end to the array. Each single spot on the array has several copies of the same oligonucleotide. The oligonucleotides are designed to contain all possible k-mers. Typically the length L is 60 bp and the value of k is 10 bp [8]. Then, fluorescently marked TFs are allowed to bind to the array, and the intensity of each spot is measured using laser. The intensity is proportional to the amount of TFs bound to that spot. The result of the experiment is a list of oligonucleotides with corresponding intensities. For longer binding sites, especially for dimeric sites, PBM might not produce enough different sequences to obtain accurate models.

5.1.2 In vivo method

Chromatin ImmunoPrecipitation followed by sequencing (ChIP-seq) [64] can give information about the binding sites of a TF in the tissue type under consideration. First, the bound proteins are fixed to the chromatin using formaldehyde. Then the nuclei of the cells are extracted, and the genome is broken into short pieces. The pieces that were bound by the TF are selected using an appropriate antibody. The selected pieces are amplified using PCR and then sequenced. The resulting sequence reads are mapped to the genome, and over-represented regions in the genome are likely to contain binding site(s) of the TF. The degree of enrichment can give a score for the region. As the width of the enriched regions can be hundreds of base pairs, the resolution of the binding site location is not very good. Versions of ChIP-seq, such as ChIP-exo [61] and ChIP-nexus [29], have been developed to improve the resolution.

5.2 SeedHam method

In this section we give a high-level overview of the SeedHam algorithm, presented in Paper I. It can be used for obtaining a PPM for a TF from count-based data, such as SELEX or ChIP-seq data. The idea can be extended to ADM models as shown in Paper III. In addition to the set of sequences in which we know the binding sites of the TF are enriched, the length ℓ of the binding model/sites needs to be known. The most common ℓ-mers can be assumed to have high affinity towards the TF under investigation. We choose one such common subsequence as the *seed*, from which we build the binding model. Furthermore, the ℓ-mers that are within a small Hamming distance from the seed are likely to have relatively high affinity towards the TF as well. So, all the occurrences of the small Hamming neighbourhood of the seed are declared as binding sites, and by aligning them in a special way, we obtain a binding model. If the positions in the binding sites of the TF are assumed to be independent, or if dependencies are assumed to appear only between the adjacent positions, a small Hamming neighbourhood should contain all the information necessary to obtain accurate models, provided the data is large enough.

So, the SeedHam algorithm consists of three phases:

- Find a seed s of length ℓ;
- Locate all occurrences of the Hamming neighbourhood of the seed s in the data;
- Align the occurrences to obtain a binding model.

However, all these phases have details which complicate the algorithm. We will address the problems of each phase in turn in the subsequent subsections.

5.2.1 Finding the seed

Although the most common ℓ-mer in the data is a more or less unique sequence, it might not always be the one we want to use. There are some TFs, such as HOXB13, that seem to have two different binding profiles [51]. There should be two seeds corresponding to the two profiles. When the SeedHam algorithm is started using a seed of a profile, the result should be the binding model for the profile corresponding to that seed.

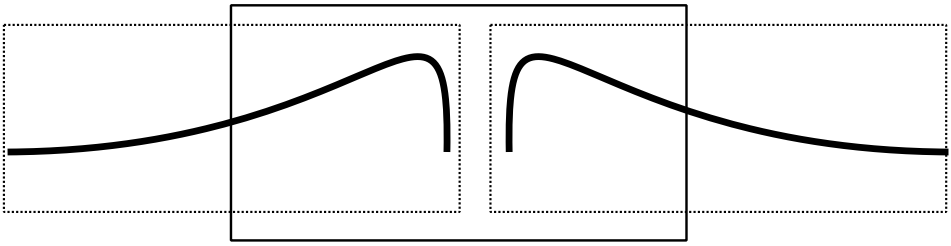


Figure 5.1: The curved lines indicate the information content of the dimeric motif as the function of position. The most common ℓ-mer corresponds to the middle part of the dimer (solid line box) instead of either of the half-sites (dotted line boxes).

Even if a TF does not have multiple monomeric binding profiles, there may still be in the data enriched subsequences that belong either to monomeric binding sites of the TF or to different dimeric binding sites. The most common ℓ-mer of the data could be, for example, just a middle section of a dimeric binding site instead of a monomeric binding site, see Figure 5.1. Some kind of global analysis of the enriched subsequences in the data could be used to decide which subsequences are monomeric or dimeric binding sites, and which are just shifts of the full binding sites.

The data might also contain repetitive low-complexity sequences (like ACACACAC), especially in ChIP-seq data, that are very frequent, but might not have anything to do with TF binding sites. These could first be masked, for example with RepeatMasker [74], before finding the common ℓ-mers in the data. Or alternatively, low-complexity sequences could be rejected as seeds, by requiring that all four bases be included in the seed, as has been done in BEESEM [65].

Finally, the selection of the seed may affect the ambiguity between selecting a binding site or its reverse complement, in case both belong to the Hamming neighbourhood of the seed. This will be further discussed in the next subsection.

5.2.2 Locating the occurrences of the Hamming neighbourhood

As the SeedHam algorithm is designed for learning monomeric models only, we assume that the data does not contain overlapping dimeric occurrences. However, even if we assume that the TFs had not bound overlapping sites in the data, there may still be occurrences of Hamming neighbours of the seed partly overlapping the real binding sites. These occurrences appear out of

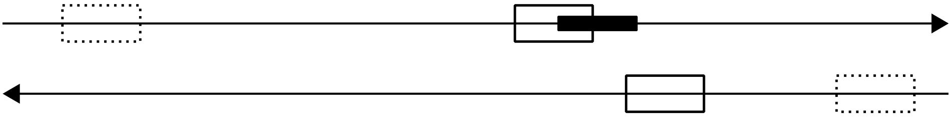


Figure 5.2: A true occurrence (solid black bar) induces two artefact occurrences (hollow box) in separate strands. Two false occurrences rise purely from the background (dotted hollow box).

randomness, and can occur in either DNA strand, see Figure 5.2. For example, if we assume that in positions 4–9 of sequence GCTACTACTGTAGT there is the true occurrence ACTACT and the other positions were distributed randomly, then, with high probability, there are occurrences of the Hamming neighbourhood of ACTACT with radius 1 in positions 1–6 and, as reverse complement, in positions 9–14. The probability of these artefact occurrences depends on the self-similarity of the binding model θ. Some occurrences of the Hamming neighbours of the seed can also come purely from the background.

Let us assume that we know the binding model θ, the background model θ_0, and the fraction λ of the sites that are true binding sites. Assume further that there are N ℓ-windows in the data X, that is, there are N possible locations where the TF could bind in theory. The relationship between the total counts of ℓ-mers and the counts of artefact occurrences and the real binding sites is described by the equation

$$\begin{aligned} count(u) =& \lambda N(P_\theta(u) + P_\theta(u^{-1})) \\ &+ (1 - \lambda(2\ell - 1))N(P_{\theta_0}(u) + P_{\theta_0}(u^{-1})) \\ &+ \lambda N \sum_{|j|=1}^{\ell-1} (P_{\theta_j}(u) + P_{\theta_j}(u^{-1})), \end{aligned} \tag{5.1}$$

where $u \in \Sigma^\ell$. The models θ_j are formed by prepending (appending) the prefix (suffix) of length $\ell - j$ of θ with j copies of background model θ_0, if j is negative (positive). The first term counts the real occurrences (type i), the second term of the right-hand side corresponds to ℓ-mers arising purely from the background distribution θ_0 (type ii), the third term corresponds to ℓ-mers arising partly from the background and partly from the real binding site (type iii). When locating the binding sites of the TF for alignment, the sites of type (ii) and (iii) must be ignored, as they are not distributed according to the binding model θ. False occurrences of type (ii) are easy to remove, if we have an estimate for the signal fraction λ and the background model θ_0. However, subtracting false occurrences of type (iii) would require knowing the binding model θ as well, which is what we are trying to learn. An ad hoc iterative approach where an estimate for θ could be used to obtain a better estimate. But there are no guarantees that this would converge to the correct binding model. Another alternative approach is presented in Paper I, where an occurrence of a Hamming neighbour is selected if all overlapping occurrences are farther away from the seed in Hamming distance than the occurrence under consideration. This heuristic method may cause bias to the selected occurrences, though.

There is also a problem with choosing the correct orientation of the site. Two things can go wrong here:

- If both u and u^{-1} are Hamming neighbours of the seed, then only one of them should be selected, as the TF cannot bind both ways simultaneously.
- Even if only one of the orientations, say u, give an ℓ-mer which is a Hamming neighbour of the seed, we might still want to exclude it. This is because the correct orientation could be u^{-1}, which is not a Hamming neighbour of the seed, and should therefore be excluded.

The choice of the seed has an effect on the uniqueness of the orientation, as is further discussed in Paper I. In brief, the seed s should be such that the Hamming neighbourhoods of s and s^{-1} intersect as little as possible.

The choice of the radius of the Hamming neighbourhood should be based on the size of the data, the signal strength, and the length of the model. If the data is small but the signal is strong then a higher radius gives more accurate results. Otherwise a small radius should be preferred, because a high radius may introduce noise to the model. For PPM models the radius needs to be at least 1, and for ADM models it needs to be at least 2. The effect of different Hamming radii on learning PPM and ADM models is analyzed in more detail in Papers I and III.

5.2.3 Aligning the occurrences of the Hamming neighbourhood

Locating the putative binding sites using the Hamming neighbourhood of a seed has the unfortunate effect that the predicted sites cannot be directly aligned anymore. In more detail, assume that we have a multiset A of binding sites that are correctly oriented and that are sampled according to the distribution given by the binding model θ. So, aligning the multiset A in the standard way should approximately give the binding model θ. Let $s \in A$ be a relatively frequent ℓ-mer and ρ be a Hamming radius. Now we can define a subset $B = \{u \in A | h(s, u) \leq \rho\} \subset A$. The standard alignment of this multiset does not give the binding model θ, as B is a biased sample of A. This bias can be corrected by using a special alignment. Let us first assume that the binding model is a PPM. For each column j of the binding model we have to use a corresponding subset C_j of B, whose alignment gives correct distribution for column j. Specifically, each element $u \in C_j$ can have at most $\rho - 1$ mismatches against the seed s outside the column j. (In the special case that $s_j = \mathrm{N}$, we have $C_j = B$.) This way each nucleotide in position j is treated equally. Had we instead allowed the full set B for the alignment of position j, then for nucleotide s_j more mismatches outside position j would have been allowed than for other nucleotides. In the case of an ADM model the correction is more complicated due to the dependencies between adjacent positions. The correct way to align Hamming neighbourhoods is derived in Paper III.

5.2.4 Time and space complexities

The basic form of the SeedHam algorithm for learning a PPM has time and space requirement $O(|X| + \ell \cdot \min(|X|, K))$, where K is the number of distinct sequences in the Hamming neighbourhood of the seed. But if the artefact subtraction is used, there will be an additional term exponential in ℓ.

The SeedHam algorithm for learning an ADM model, essentially Algorithm S1 in Paper III, has time-complexities $O(\ell\rho)$ and $O(\ell \min(\rho, \ell))$ for initialization (lines 1–2) and bias correction (lines 11–27) phases, respectively. Since the counting phase (lines 3–10) takes the same time as for the PPM version, the total time-complexity is again $O(|X| + \ell \cdot \min(|X|, K))$.

5.2.5 Related work

Berger et al [8] have used a similar alignment method for learning PWM models (using fixed Hamming radius 1) from PBM data without any justification for correctness. Later this method has been dubbed Seed-and-wobble.

There have been many methods that tried to find over-represented motifs from data, where the motifs were (restricted) IUPAC sequences [10, 24, 73]. In all these methods the over-representation is defined by the Z-score, and the artefact motifs are the main problem. The difference between these methods and SeedHam (in addition to using different types of motif) is that SeedHam tries to learn a single model and handles artefacts at the occurrence level, whereas the other methods try to learn multiple motifs and handle the artefacts at motif level. DECOD [31] learns multiple PWM models by masking the previously found motifs from the data before finding the next motif. The problem of shifted motifs is handled using deconvolution.

5.3 EM algorithm

The EM algorithm (Expectation Maximization) [16] is a framework of algorithms whose instances can be used to find a maximum likelihood parameters for model η and data X specific to that instance. We want to maximize

$$P(X|\eta) \tag{5.2}$$

over all distributions given by parameters η. In case finding the maximum by derivating this likelihood function is hard, one may try to apply the EM algorithm instead. It assumes there are hidden variables Z which make the maximization easier if they are known. Then the likelihood of the model is considered on the condition that the additional hidden data is given: $P(X, Z|\eta)$. The single maximization task is replaced by a sequence of simpler maximization tasks. If we assume we have an estimate for the parameter η, say $\eta^{(t)}$, then we first try to compute the expectation $E_{Z|X,\eta^{(t)}} P(X, Z|\eta)$ using the estimated parameters and the data. This is the *expectation phase* of the algorithm. Then in the *maximization phase* we compute the next estimate $\eta^{(t+1)} := \arg\max_\eta E_{Z|X,\eta^{(t)}} P(X, Z|\eta)$. If we start from some initial parameter $\eta^{(0)}$, possibly chosen randomly, we get a sequence of parameters $\eta^{(0)}, \eta^{(1)}, \eta^{(2)}, \ldots$. It can be shown that in this sequence the next parameter gives better likelihood in Expression 5.2 than the previous parameter, unless the algorithm already converged [16].

If the likelihood function is bounded from above, then the EM algorithm must converge. Note that there are families of distributions, such as the exponential families, for which the expectation and maximization phases are of particularly simple form. For an EM algorithm to work, however, it is enough that we can compute the expectation and maximization, as we have done below.

There is a risk that the algorithm converges only to a local maximum, not global. This risk can be reduced by starting the EM randomly from several different starting points. Or alternatively one can try to use a single good starting point, which is expected to be close to the global maximum.

As a more concrete instance of the EM algorithm we will discuss estimating a probabilistic mixture model composed of a background model θ_0 and p PPM models $\theta_1, \ldots, \theta_p$. The observed data X consists of n nucleotide sequences of length ℓ. For simplicity, we assume first that each PPM motif has width ℓ, then later we show how to generalize this to the case where the motif width can be shorter than the length of the sequences. Each sequence X_i is assumed to be generated randomly by one of the models of the mixture. The probability of each model is given by the non-negative weights $\lambda_0, \lambda_1, \ldots, \lambda_p$ (mixing parameters) that sum up to 1. In this example we define the hidden variables so that variable Z_{ik} is 1 if the ith sequence was generated by model k, and 0 otherwise, for $i = 1, \ldots, n, k = 0, 1, \ldots, p$. We have $\lambda_k = P(Z_{ik} = 1)$ for any $i = 1, \ldots, n, k = 0, 1, \ldots, p$. The total model η is then $(\lambda_0, \lambda_1, \ldots, \lambda_p, \theta_0, \theta_1, \ldots, \theta_p)$. If the sequence X_i was generated by the background model θ_0, then its probability is

$$P(X_i|Z_{i0} = 1, \theta_0) = \prod_{h=1}^{\ell} \theta_0^{X_{ih}}. \tag{5.3}$$

If on the other hand the sequence was generated by PPM θ_k, then its probability is

$$P(X_i|Z_{ik} = 1, \theta_k) = \prod_{h=1}^{\ell} \theta_k^{X_{ih},h}. \tag{5.4}$$

In the EM algorithm, instead of maximizing the likelihood, we maximize the logarithm of the likelihood to ease the computations. Since the logarithm is a monotonically increasing function, the maximum of the log likelihood occurs in the same point as the maximum of the likelihood. The log likelihood of model η given complete data X and Z is

$$\log P(X, Z|\eta) = \log \prod_{i=1}^{n} \sum_{k=0}^{p} Z_{ik} \lambda_k P(X_i|Z_{ik} = 1, \eta). \tag{5.5}$$

Since, for any $i = 1, \ldots, n$, only one of the variables Z_{ik} has value 1, we get

$$\log P(X, Z|\eta) = \sum_{i=1}^{n} \sum_{k=0}^{p} Z_{ik} \log(\lambda_k P(X_i|Z_{ik} = 1, \eta)). \tag{5.6}$$

When computing the expectation of the log likelihood, by linearity of expectation, we only need to compute the expectation of the hidden variables. We denote these by lowercase z. By Bayes' rule we get

$$z_{ik}^{(t)} := E[Z_{ik}|X, \eta^{(t)}] = P(Z_{ik} = 1|X_i, \eta^{(t)}) = \frac{\lambda_k^{(t)} P(X_i|Z_{ik} = 1, \eta^{(t)})}{P(X_i|\eta^{(t)})}. \tag{5.7}$$

The denominator can be computed as

$$P(X_i|\eta^{(t)}) = \sum_{k=0}^{p} \lambda_k^{(t)} P(X_i|Z_{ik} = 1, \eta^{(t)}). \tag{5.8}$$

In the maximization phase we now need to compute

$$\arg\max_{\eta} \sum_{i=1}^{n} \sum_{k=0}^{p} z_{ik} (\log \lambda_k + \log P(X_i | Z_{ik} = 1, \eta)). \tag{5.9}$$

By reordering the terms we get

$$\begin{aligned}\eta^{(t+1)} := \arg\max_{\eta} \Big[& \sum_{i=1}^{n} \sum_{k=0}^{p} z_{ik} \log \lambda_k \\ & + \sum_{i=1}^{n} z_{i0} \log P(X_i | Z_{i0} = 1, \theta_0) \\ & + \sum_{i=1}^{n} \sum_{k=1}^{p} z_{ik} \log P(X_i | Z_{ik} = 1, \theta_k) \Big].\end{aligned} \tag{5.10}$$

As the three parts above only depend on parameters λ, θ_0, and θ_k, respectively, they can be maximized separately.

Let us first compute $\arg\max_\eta \sum_{i=1}^{n} \sum_{k=0}^{p} z_{ik} \log \lambda_k$. By reordering we get $\arg\max_\eta \sum_{k=0}^{p} (\sum_{i=1}^{n} z_{ik}) \log \lambda_k$. We define $e_k := \sum_{i=1}^{n} z_{ik}$ and claim that $\frac{e_k}{\sum_{k'} e_{k'}}$, for $k = 0, 1, \ldots, p$, is the maximum likelihood estimate for parameter λ_k. We proof this with Gibbs' inequality, which states that for distributions p and q

$$\sum_{k=0}^{p} p_k \log q_k \leq \sum_{k=0}^{p} p_k \log p_k, \tag{5.11}$$

where equality holds if and only if $p_k = q_k$ for all $k = 0, \ldots, p$. If we now set $p_k = \frac{e_k}{\sum_{k'} e_{k'}}$ and $q_k = \lambda_k$, we get

$$\sum_{k=0}^{p} \frac{e_k}{\sum_{k'} e_{k'}} \log \lambda_k \leq \sum_{k=0}^{p} \frac{e_k}{\sum_{k'} e_{k'}} \log \frac{e_k}{\sum_{k'} e_{k'}}. \tag{5.12}$$

By multiplying both sides by $\sum_{k'} e_{k'}$ we get

$$\sum_{k=0}^{p} e_k \log \lambda_k \leq \sum_{k=0}^{p} e_k \log \frac{e_k}{\sum_{k'} e_{k'}}. \tag{5.13}$$

So, by Gibbs' inequality the maximum is attained when $\lambda_k = \frac{e_k}{\sum_{k'} e_{k'}}$.

Maximizing the second part gives new estimate $\theta_0^{(t+1)}$ for the background model parameters. The second part is

$$\sum_{i=1}^{n} z_{i0} \log \prod_{a \in \Sigma} (\theta_0^a)^{N_{ia}} = \sum_{i=1}^{n} z_{i0} \sum_{a \in \Sigma} N_{ia} \log \theta_0^a = \sum_{a \in \Sigma} (\sum_{i=1}^{n} z_{i0} N_{ia}) \log \theta_0^a, \tag{5.14}$$

where N_{ia} is the count of nucleotide a in sequence X_i. If we denote $e_a = \sum_{i=1}^{n} z_{i0} N_{ia}$, then using the same reasoning as above with the mixing parameters, we get new estimates $\theta_0^a = \frac{e_a}{\sum_{a' \in \Sigma} e_{a'}}$ for all $a \in \Sigma$.

Finally, in the third part we have

$$\begin{aligned}\sum_{i=1}^{n} \sum_{k=1}^{p} z_{ik} \log \prod_{h=1}^{\ell} \theta_k^{X_{ih},h} &= \sum_{i=1}^{n} \sum_{k=1}^{p} z_{ik} \sum_{h=1}^{\ell} \log \theta_k^{X_{ih},h} \\ &= \sum_{k=1}^{p} \sum_{h=1}^{\ell} \sum_{a \in \Sigma} (\sum_{i=1}^{n} [X_{ih} = a] z_{ik}) \log \theta_k^{a,h},\end{aligned} \tag{5.15}$$

where $[X_{ih} = a]$ is 1 or 0 depending on whether the equality holds or not, respectively. Let us write $e_{kah} := \sum_{i=1}^{n} [X_{ih} = a] z_{ik}$. Since the columns of PPM models are independent, we get new estimates for each $k = 1, \ldots, p$ and $h = 1, \ldots, \ell$ by $\theta_k^{ah} = \frac{e_{kah}}{\sum_{a' \in \Sigma} e_{ka'h}}$ using the same reasoning as earlier.

The more general mixture model where the length ℓ of models can be shorter than the sequence length L can be obtained by using hidden variables Z_{ikj} that have value 1 only when the model θ_k has generated the occurrence starting in position j of sequence X_i. The sum $Z_{i0} + \sum_{k=1}^{p} \sum_j Z_{ikj}$ of hidden variables for a sequence X_i is required to be 1. Then Equation 5.4 is replaced by equation

$$P(X_i | Z_{ikj} = 1, \theta_k, \theta_0) = \prod_{h<j \text{ or } j+\ell \leq h} \theta_0^{X_{ih}} \prod_{h=j}^{j+\ell-1} \theta_k^{X_{ih},h-j+1}. \tag{5.16}$$

And in the maximization phase the terms corresponding to the background model are collected together for estimation of parameter θ_0.

In addition to this relaxation, the MODER implementation has several other generalizations. For example, the sequences do not need to have the same length, so we denote the length of sequence X_i with L_i. Also, we have included the SeedHam-style restriction of learning of a model to the Hamming neighbourhood of a seed to prevent close-by models from interfering with each other, and to speed-up the implementation. Furthermore, in Paper III the EM algorithm is extended to include learning of ADMs, see also Section 5.5.

Since the data takes nL space, the model takes $1+p+4+4p\ell$ space, and the hidden variables take space $np(L-\ell+1)$, the total space complexity is $O(nL + p\ell + np(L-\ell+1))$. Computing the probabilities in the expectation phase takes $O(np(L-\ell+1)\ell)$ time, and maximizing the models also takes time $O(np(L-\ell+1)\ell)$ per iteration.

Method	Time (hh:mm:ss)	Iterations
PPM monomer	00:03:11	7.24
ADM monomer	00:31:57	69.99
PPM with COB	09:49:11	27.73
ADM with COB	19:16:26	100.39

Table 5.1: Average running times and numbers of iterations over runs of 314 SELEX data sets of 95 485–1 294 346 sequences of length 40 bp. The reported time is the CPU time.

Lawrence and Reilly [44] first presented an EM algorithm for learning a motif model from a set of sequences. The algorithm assumed that every sequence contained at least one occurrence of the motif (OOPS, One Occurrence Per Sequence). Bailey and Elkan [7] extended this to a ZOOPS model (Zero or One Occurrence Per Sequence) in their widely-used program MEME by using a mixture model of two components: background sequence and motif sequence. Above we have extended this mixture model to include multiple motif models.

The main problem in MEME is its slowness. MEME takes all possible nucleotide sequences of given length, then builds an initial seed model for each of these sequences and runs EM for a few rounds to see if it starts to converge. Then the best seeds are run to convergence. Secondly, after convergence, several significance tests are run to decide, which binding sites to align to get the final model. As MODER does neither of these steps, it can handle also large data sets in reasonable time, see Table 5.1.

SEME [90] tries to improve this by using importance sampling to use only a small portion of the data to learn the model. They claim this improves both the speed and accuracy of the algorithm. SEME also allows modifying the binding model length during the run. Also EXTREME [59] tries to reduce the time requirement of MEME. It uses an online EM algorithm in which at each iteration only a single new sequence is considered. It also uses over-represented subsequences as seeds for the algorithm. STEME [60] is yet another MEME variant aiming for faster operation. It uses a suffix tree to implement an approximation of the EM algorithm.

Omidi et al [55] use an EM algorithm to learn the parameters of their Dinucleotide Weight Tensor (DWT), which models all pairwise dependencies. InMoDe [20] learns an inhomogeneous Parsimonious Markov Model (iPMM), which allows a Markov model with varying context lengths. It uses a variant of the EM algorithm, which optimizes the BIC-score [69] (see Subsection 5.7.4) instead of likelihood.

5.4 Co-Operative Binding model

We can use the EM algorithm to estimate COB tables. For each dimeric case k_1k_2od we add to our mixture model the weight $\lambda_{k_1k_2od}$ and the corresponding binding model $\tau_{k_1k_2od}$. In the case where the distance $d \geq 0$ the dimeric binding model can be though to be built from the pair $(\theta_{k_1}, \theta_{k_2})$ of monomeric binding models. The parameters θ_k are then derived from the monomeric cases of the kth model and from the dimeric cases $kk'od$ and $k'kod$ for any $k' = 1, \ldots, p$, orientation o, and distance $d \geq 0$. In the overlapping case ($d < 0$) the situation is more complicated and is discussed in Paper II, for PPMs, and in Paper III for ADMs.

The coMOTIF method [86] uses a similar mixture model and an EM algorithm to learn it, but as its interest is mainly in refining the monomeric models and finding out which TFs like to bind together, it does not output anything similar to our COB table, even though it could.

Bioprospector [46] uses Gibbs sampling to learn two PPMs and a list of two-block sites. But no representation for distribution of distances and orientations is given, possibly since at that time the binding data was scarce. Bipad/MaskMinent [9, 47] first chooses a random multiple alignment of bipartite sites, and then tries to greedily improve the information content of the multiple alignment by modifying it a sequence at a time. This is repeated until the information content does not improve anymore. This procedure is started from several random initial points to avoid local maximum. The algorithm represents the distribution of orientation and distance only in an ad hoc manner, as histograms. SPAMO [84] gets as inputs a ChIP-seq data for a TF, a PPM model (primary motif) for the TF, and a database of PPMs (putative secondary motifs) whose co-occurrences with the primary motif are tested. The frequencies of distances between binding sites of primary and secondary motifs are displayed as separate histograms for each orientation. The statistical significance of a fixed distance is tested using a binomial test based on the null model, which assumes each distance is equally probable. iTFs [38] uses a given database of binding motifs and FIMO scanner [28] to find occurrences of the motifs in genomic data. The distance and orientation of adjacent occurrences are recorded, and Fisher's exact test is used for finding significant biases in (i) orientation, (ii) distance, and (iii) distance specific to an orientation. Distances are binned into the following bins 0–10 bp, 10–25 bp, 25–50 bp, and 50–100 bp. TACO [33] gets a set of regulatory regions from a genome and a database of motifs as inputs. Over-represented dimeric motifs are ranked according to their p-values.

5.5 Learning ADM models

Paper III extends the EM algorithm from Paper II so that it can handle both inhomogeneous zeroth and first order Markov chains as binding models, that is, PPMs and ADMs. As the algorithm framework is similar for both model types, we list here only the parts that change in the ADM model case.

The joint likelihood of the model parameters η, given data X and missing information Z, has exactly the same form as with PPMs:

$$L(\eta|X,Z) = P(X,Z|\eta) = \prod_{i=1}^{n} \Big(Z_{i0} \cdot \lambda_0 \cdot P(X_i|Z_{i0}=1,\eta) + \sum_{k \in M \cup D^+ \cup D^-} \sum_{j \in S_{ik}} Z_{ikj} \cdot \frac{\lambda_k}{|S_{ik}|} \cdot P(X_i|Z_{ikj}=1,\eta) \Big).$$

However, the probabilities of sequences are computed differently, as the model η now contains ADMs instead of PPMs. In Section 3.2.3 we represented the ADM model as a matrix θ with shape $16 \times \ell$ whose elements $\theta^{ab,h}, a, b \in \Sigma, 1 \leq h \leq \ell$ are the transition probabilities $P(X_h = b|X_{h-1} = a)$. The probability of a sequence $X = X_1 X_2 \cdots X_\ell$ given by the ADM model θ is

$$P(X) = \prod_{1 \leq h \leq \ell} P(X_h|X_{h-1}) = \prod_{1 \leq h \leq \ell} \theta^{X_{h-1}X_h,h} = \prod_{1 \leq h \leq \ell} \theta[X_{h-1}X_h, h].$$

Note that we define X_0 to be A, so that the initial probabilities $P(X_1 = b) = P(X_1 = b|X_0 = \mathrm{A}) = \theta^{\mathrm{A}X_1,1}$ get treated symmetrically with the transition probabilities. With the ADM models we use notation $\theta^{b,h} := P(X_h = b)$ for the probability of symbol b in position h, that is,

$$\theta^{b,h} = \sum_{a_1,\ldots,a_{h-1} \in \Sigma} \theta^{\mathrm{A}a_1,1}\theta^{a_1a_2,2} \cdots \theta^{a_{h-1}b,h}$$

for $b \in \Sigma, 1 \leq h \leq \ell$.

The reverse complement θ_k^{-1} of θ_k is an ADM such that

$$\theta_k^{-1}[ab,h] = \frac{\theta_k[\overline{ba}, \ell_k - h + 2]\theta_k[\bar{b}, \ell_k - h + 1]}{\theta_k[\bar{a}, \ell_k - h + 2]},$$

where $\overline{ab}$ denotes the complementary dinucleotide $\bar{a}\bar{b}$ (e.g., $\overline{\mathrm{AC}} = TG$). If the denominator is zero, we define $\theta_k^{-1}[ab,h] = 0$ for all $b \in \Sigma$.

The SeedHam-style restriction of learning of a binding model to the Hamming neighbourhood of a seed is slightly more complicated in the ADM case because of the dependencies between adjacent positions. The full algorithm that corrects the seed bias caused by the Hamming sample is given in detail in the Supplement of Paper III. The text and the pseudo-code algorithm also show how the pseudo-counts need to be added in this case.

Again, if the COB table and the associated dimers are learned as well, then the case of overlapping ADM models (distance $d < 0$) is more complicated than with the PPM models. This is handled in the supplement of Paper III.

Similarly as with learning the PPMs, the data takes nL space, the model takes $1+p+4+16p\ell$ space, and the hidden variables take space $np(L-\ell+1)$, the total space complexity is $O(nL + p\ell + np(L - \ell + 1))$. Computing the probabilities in the expectation phase takes $O(np(L - \ell + 1)\ell)$ time, and maximizing the models also takes time $O(np(L - \ell + 1)\ell)$ per iteration. Table 5.1 shows examples of running times and numbers of iterations for learning ADM models.

5.6 Review of other existing methods to optimize binding models

Gibbs sampling is another common method for learning TF binding models. Lawrence et al originally introduced it for motifs in protein sequences [43], but it has later been used for TF binding site motifs, see for example [26, 79]. Gibbs sampling can be though of as a stochastic version of EM algorithm: instead of using all putative sites from sequence X_i according to weights z_{ikj}, it samples just one site proportionally to weights z_{ikj}. From these sampled sites the model is estimated. The algorithm is iterated until convergence. However, Gibbs sampling can get stuck in a local mode. This can be avoided, for example, by running the algorithm several times, starting from different initial points.

RPMCMC (Repulsive Parallel Markov Chain Monte Carlo) [32] is also based on Gibbs sampling, but it uses several Gibbs samplers to learn multiple models simultaneously. To avoid the different samplers converging to the same model, a repulsive force is used to keep the motifs separate.

Laurila et al [42] predict binding sites in a genome while taking into account the simultaneous competition between multiple TFs over possibly overlapping binding sites. They use a Metropolis–Hastings algorithm to compute the posterior probability of a non-overlapping subset of binding sites by a given set of TFs. The PWM models are random variables having Dirichlet priors, whose hyperparameters are specified by PWMs from TRANSFAC [49]. The protein–protein interactions available from existing databases can be used to specify the prior probabilities of each set of TF and binding site pairs. Wasson et al [83] give another method to compute the posterior probabilities of binding sites when competition between multiple factors is taken into account. Their method includes other factors besides TFs, such as nucleosomes, in the competitive model. The concentrations of TFs and nucleosomes are also included in the model.

GADEM [45] combines several techniques to find multiple dimeric motifs. It first finds over-represented k-mers ($k = 3, 4, 5, 6$) and forms dimeric patterns. Patterns of a randomly selected sample are then refined using the EM algorithm to PPM models. Significant sites of these motifs are located in the data, and the E-value of each alignment is computed. The E-value is used as a fitness score for a genetic algorithm, which tries to iteratively improve the models. Weak motifs are modified by either a mutation or a recombination.

Alipanahi et al [2] predict sequence specificities of TFs and RNA binding proteins using deep learning. Here deep learning produces a weighted set of PWMs. They showed that deep learning can outperform 26 other common discovery algorithms. Problems with deep learning include the possibility of overfitting and complicated binding models. This deep learning approach has later been been improved for example by Zeng et al [87] and Shen et al [71].

Colombo and Vlassis [13] use spectral methods to learn a PPM model. The idea is to first form a multi-dimensional array, which tells the observed relative frequencies of all ℓ-mers in the data. Then matrix decomposition is used to represent this as a mixture of p PPM models. Then one of these components is chosen as the result based on the p-values of the PPMs. The method is fast and performs well compared to our early version of SeedHam [36] and DREME [6].

Annala et al [5] describe a linear regression method to predict PBM probe intensities. All 4–8-mers of the PBM array are used as features in the design matrix $H = (h_{s,k})$, where $h_{s,k}$ has value 1 if the K-mer k is contained in the probe sequence s, and 0 otherwise. The solution vector of feature affinities can be used to predict intensities of sequences of the same length as the original probes. They have later extended the method to handle SELEX data [37]. In the case of SELEX data the probes are replaced by SELEX reads, and the probe intensities are replaced by binary variables, which have value one for bound reads and zero for unbound reads.

Pelossof et al [57] use regression techniques to try to predict TF binding intensities of PBM probes based on the TF's amino acid sequence. An interaction model matrix is learned from the k-mer feature matrix of the probes of the PBM and from the k'-mer feature matrix of the amino acid sequences. All the amino acid sequences must belong to the same TF family as does the TF whose binding intensities we are trying to learn. Because the regression includes two feature matrices, it is called bilinear regression.

Sharon et al [70] propose a binding model, which considers both mononucleotide and dinucleotide (not necessarily adjacent) features. The probability of a sequence is based on the sum of the frequencies of its statistically significant features. To keep the model complexity low, L_1-regularization is used. A discriminative ad hoc method is used to find aligned k-mers that are enriched in a positive set over a negative set.

Biophysical methods try to learn the real affinities, that is, the Gibbs free energy between a DNA and a TF, based on thermodynamical principles. Djordjevic et al [17] present a biophysical approach to discover TF binding sites, which should give accurate binding models also in the case of high TF concentration when the high-affinity binding sites are saturated. Their algorithm QPMEME uses quadratic programming to learn the parameters of a biophysical model from a set of known binding sites. Foat et al [27] gave an algorithm called MatrixREDUCE, which used regression methods to learn the parameters of a biophysical model from BPM data. They later [62] extended their method to incorporate more complicated sequence features besides the positional mononucleotides. The extended method, called FeatureREDUCE, could include, e.g., dinucleotide features, and robust regression methods were used to learn the model parameters. Biophysical methods have also been applied to SELEX data: Ruan et al [65] present an algorithm called BEESEM that uses an EM algorithm to learn the model parameters.

5.7 Measuring goodness of the models

In this section four measures for comparing the goodness of the learned models are discussed.

5.7.1 Distance to ground truth

If we know the correct model beforehand, then we can use various distance measures to compare our estimated model to the correct one. In case we have generated some test data using a model, then we obviously know what the correct result is. But also in case there is a good model available, obtained possibly using some other more accurate and expensive method, we can use this as a *ground truth* against which to compare a model obtained from our own method. We have mainly used the max norm of parameter differences to compare two binding models, as it is an intuitive distance measure. On the downside, it is not specifically meant for probability distributions, and it can be too conservative. Total variation distance on the other hand is meant for comparing probability distributions, but it is expensive to compute for longer models, and it is less intuitive than the max norm. Finally, the square root of JS-divergence is a distance between distributions, although it does not have very intuitive interpretation. Papers I, II, and III use max norm distance to compare the ground truth model and the learned model.

5.7.2 Correlation

If a ground truth model is not available, then one way of measuring the goodness of a model is to check how well the model explains the k-mer counts in a data. This is achieved by computing the correlation (coefficient of determination, R^2) between the counts of a k-mer in the data and the score of the k-mer given by our binding model. The value of k should be high enough to cover all the important features of the model, but not so long that the counts are low and only a small fraction of all possible k-mers are present in the data, i.e. the expected count of a k-mer according to the background model should be at least 1.

The score is computed as the maximum sum of log ratios $\log \frac{p(x)}{q(x)}$ over aligned positions x over all possible alignments of the k-mer and the model, where p and q are the probabilities given by the binding model and the background model, respectively. Figure 5.3 gives an example of a correlation plot between the logarithm of observed counts and the expected scores. Logarithm of counts is used to make sure they are on the same scale as the scores. Correlations are used in Papers II and III to assess goodness of learned models.

5.7.3 Receiver operating characteristic

If we choose some threshold T for the score given by motif, we can then use the model to classify sequences into positive or negative sequences, depending on whether the maximum score given by the model to the sequence is above the threshold T or not. If we additionally have two sets of sequences, one whose sequences are known to be positive (for example, they contain a binding site of a TF), and another whose sequences are known to be negative (do not contain any binding sites), then we can use these data

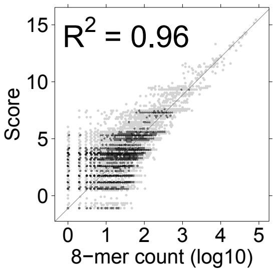


Figure 5.3: Correlation of observed counts and scores by a binding model of 8-mers. The coefficient of determination (R^2) is shown in the left upper corner. Least squares line fitted to the data points is plotted in red.

sets to check our model's performance as a binary classifier. If we correctly predict a sequence to be positive (negative), then we call it a true positive (true negative). If we however fail to predict the class of the sequence, then we call it either false positive (FP) or false negative (FN). The count ratio $\frac{\#(TP)}{\#(P)}$ is the true positive rate, and the ratio $\frac{\#(FP)}{\#(N)}$ is the false positive rate.

When we slide the threshold T from maximum possible to minimum possible score, the points (FPR, TPR) draw the Receiver Operating Characteristic (ROC) curve. An example ROC curve is shown in Figure 5.4. The performance of a model as a classifier is the better the closer the curve goes to the upper left point $(0, 1)$. The diagonal line TPR=FPR is the worst possible classifier. ROC curves are used in Paper II to evaluate goodness of the learned models.

5.7.4 Bayesian information criterion

If one increases the number of parameters in a model so that the original model can be considered as a submodel of the new model, then the maximum likelihood with the new model should be at least as high as with the original model. This may however lead to overfitting as the larger model may get higher likelihood just by modeling some random features of the data. One way of avoiding this overfitting is to use the Bayesian Information Criterion (BIC) [69], which penalizes the use of a large number of parameters. The BIC is defined by

$$\text{BIC} = \ln(n)k - 2\ln(L),$$

where L is the maximum likelihood, n is the number of data points and k is the number of model parameters. This measure is used in Paper III to compare ADMs to PPMs.

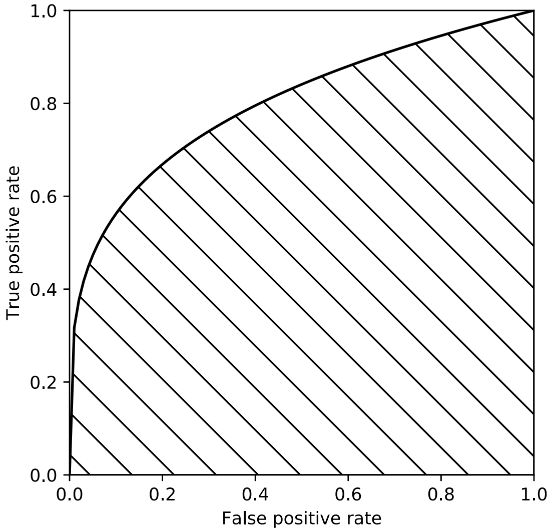


Figure 5.4: Example ROC curve, where the area under the curve is shown in a pattern.

Chapter 6

Experimental evaluation of the new methods

Paper I introduces the SeedHam method and its improved version SeedHam+. The relative strengths of each variant were discussed, and the effect of the Hamming radius and the seed length on learning accuracy was demonstrated in Figure 2 of Paper I using generated data. Briefly, optimal Hamming radius increases as the motif length ℓ increases, but decreases when the number of motif instances increases. We also compared SeedHam with two similar seed-driven PPM learning methods DREME [6] and DECOD [31]. The results of the comparison are presented in Table 2 of Paper I which shows that in majority of cases SeedHam relearned the PPM from generated data more accurately than the two other methods.

In Paper II a probabilistic framework where dimeric binding models are built modularly from monomeric models is presented, and an EM algorithm for learning the model parameters is given. The utility of the learned mixture models is demonstrated on SELEX data using correlation plots shown in Figures 3–10 for TFs HOXB13, HNF4A, TFAP2A, FLI1, FOXC1, and PKNOX2. The ability of MODER allowing deviations from the expected dimeric models is also shown to have importance. For example, for factors FLI1 and PKNOX2 MODER detects directional dimeric models where the expected models have palindromic symmetry. For factor HNF4A we have shown how different binding modes can be represented as dimeric cases of two short PPM models. Again, this required that our model allowed deviations in the overlapping or gap region.

In Figure 9 of Paper II the models learned by MODER from SELEX data are shown to generalize for classifying ChIP-seq peaks. MODER can learn mixtures of binding models also from ChIP-seq/ChIP-exo data sets, see Figures S2–S4 for examples with factors NRSF, CTCF, and RXRA.

Finally, in comparison to other methods, MODER is shown to be faster than the popular motif discoverer MEME [7] and to give models that better explain the data. In comparison between MODER and MaskMinent [47] on 40 ENCODE ChIP-seq data sets, MODER wins in 35 out of 40 cases.

In Paper III, when we repeated the modularity analysis of HNF4A from Paper II for ADM models, it was discovered that, instead of four separate models, a single ADM model could explain the data with R^2 value of 0.96, see Figures 3, 4, S3, and S4 of Paper III. When learning both PPM and ADM models from 230 SELEX data sets, the ADM models gave slightly but consistently better R^2 values than the PPM models, see Table 1 of Paper III. Next we compared MODER2 to BaMM [72], which can learn higher order binding models, using data sets from four TF families: ETS (20 TFs), bHLH (24), bZIP (14), and homeodomain containing family (172). When comparing ADMs by MODER2 and by BaMM, the difference is small, with MODER2 giving the best average in two families of TFs and BaMM in the remaining two. As for the number of wins, MODER2 is best in three families, while it seems to have difficulties with the ETS family. Table S8 of Paper III shows the value of including strong dimers into the mixture model over using only monomer models.

Chapter 7

Concluding discussion

This thesis studied models of monomeric and dimeric transcription factor binding sites and methods for learning them efficiently and accurately.

In Papers I and III we have shown that the introduced SeedHam method can be used to locate putative binding sites from a data and align them in unbiased manner to obtain either a PPM or ADM model. SeedHam takes advantage of the assumption that the binding model is a product of independent categorical distributions. The radius of the Hamming neighbourhood one should use was shown to depend on the size of the data, the fraction of signal in the data, and the length of the binding model. The method is simple and fast, but the method may in some cases produce inaccurate models due to inclusion of background into the model. As the heuristic method we used in Paper I to exclude the background did not fully solve this issue, and it considerably complicated and slowed the algorithm, we chose to use another approach to this problem.

In Paper II we introduced an EM algorithm, called MODER, that can be used for learning binding models from a set of enriched nucleotide sequences. It also prevents mixing of the background and the binding sites, as it tries to find a maximum likelihood model for both the background and the signal. If a mixture of several distinct binding models and a background model is used, then EM can learn several models simultaneously and divide the data between the models. As a next step, a probabilistic representation for strengths of different dimeric cases of a pair of TFs, the COB table, was given. It was shown that MODER can be used for learning COB tables accurately.

In experiments it was detected that frequently one or two strong dimeric cases were present. Often these dimers could be corroborated by previous research. In Paper II PPM binding models were used, and in Paper III the method was extended, and implemented as MODER2, to allow learning of

ADM models as well. This allows finding out whether PPM is a sufficient model for TF binding sites or is the more complicated ADM required in some cases. In Paper III it was noted that often PPM models were accurate enough, with ADM models giving only slightly, but consistently, better R^2 values.

In Papers II and III we assumed that dimeric binding models can be constructed from two copies of monomeric binding models. We call this property modularity. As we allowed the two monomeric instances to overlap, we had to compute the expected dimeric model in each overlapping scenario. Especially in the case of ADM models, computing the expected model proved to be non-trivial due to the dependencies between the adjacent positions. To test whether interactions between the two proteins of the dimer can modify the expected dimeric binding motif, we allowed deviation from the expected overlapping dimeric model in the overlapping area. However, we allowed the overlapping area to be at most half of the lengths of the monomeric binding models, otherwise the modular structure of the total model would collapse. This is because if the overlapping area is nearly the full length of the dimer, then the deviation would allow nearly complete independence from the monomeric model. Experiments on real data revealed many deviations from the expected model.

We restricted the learning of a model to a Hamming neighbourhood of the consensus sequence of the model even in MODER, since that can remove noise and prevent two similar models from mixing up. In addition, it can also make the implementation more efficient as only part of the data needs to be considered in the maximization phase. MODER is efficient as its time-complexity depends linearly on the data size, but if the number of monomeric models and the COB tables increases, then also the required time increases fast.

We have tested our methods both on generated data and on biological data from SELEX and ChIP-seq experiments. We have given binding models and COB tables for several different TFs. These models can easily confirm previous observations as well as gain new biological insights.

A possible future direction of our work is a scanner that could read an entire genome and predict regulatory areas. The scanner would use monomeric binding models of all TFs, and dimeric binding models of all possible TF pairs. The system would also include a binding model for nucleosomes that could help define areas of DNA accessible to TFs.

References

[1] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular biology of the cell*. Garland Science, New York, sixth edition, 2015.

[2] Babak Alipanahi, Andrew Delong, Matthew T. Weirauch, and Brendan J. Frey. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature Biotech.*, 33(8):831–838, 2015.

[3] Takanori Amano, Tomoko Sagai, Hideyuki Tanabe, Yoichi Mizushina, Hiromi Nakazawa, and Toshihiko Shiroishi. Chromosomal dynamics at the Shh locus: limb bud-specific differential regulation of competence and active transcription. *Developmental Cell*, 16(1):47–57, 2009.

[4] Malin C. Andersen, Pär G. Engström, Stuart Lithwick, David Arenillas, Per Eriksson, Boris Lenhard, Wyeth W. Wasserman, and Jacob Odeberg. In silico detection of sequence variations modifying transcriptional regulation. *PLOS Computational Biology*, 4(1):1–12, 2008.

[5] Matti Annala, Kirsti Laurila, Harri Lähdesmäki, and Matti Nykter. A linear model for transcription factor binding affinity prediction in protein binding microarrays. *PLOS ONE*, 6(5):1–13, 2011.

[6] Timothy L. Bailey. DREME: motif discovery in transcription factor ChIP-seq data. *Bioinformatics*, 27(12):1653–1659, 2011.

[7] Timothy L. Bailey and Charles Elkan. The value of prior knowledge in discovering motifs with MEME. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology, Cambridge, United Kingdom, July 16-19, 1995*, pages 21–29, 1995.

[8] Michael F. Berger, Anthony A. Philippakis, Aaron M. Qureshi, Fangxue S. He, Preston W. Estep, and Martha L. Bulyk. Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nature Biotech.*, 24(11):1429–1435, 2006.

[9] Chengpeng Bi and Peter K. Rogan. Bipartite pattern discovery by entropy minimization-based multiple local alignment. *Nucleic Acids Res.*, 32(17):4979–4991, 2004.

[10] Mathieu Blanchette and Saurabh Sinha. Separating real motifs from their artifacts. *Bioinformatics*, 17(SUPPL. 1), 2001.

[11] Pierre-Yves Bourguignon and David Robelin. Modeles de Markov parcimonieux: sélection de modele et estimation. In *JOBIM, 5êmes Journées Ouvertes en Biologie, Informatique et Mathématiques*, 2004.

[12] Alan P. Boyle, Sean Davis, Hennady P. Shulha, Paul Meltzer, Elliott H. Margulies, Zhiping Weng, Terrence S. Furey, and Gregory E. Crawford. High-resolution mapping and characterization of open chromatin across the genome. *Cell*, 132(2):311–322, 2008.

[13] Nicoló Colombo and Nikos Vlassis. Fastmotif: spectral sequence motif discovery. *Bioinformatics*, 31(16):2623–2631, 2015.

[14] The 1000 Genomes Project Consortium et al. A global reference for human genetic variation. *Nature*, 526:68–74, 2015.

[15] The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489:57–74, 2012.

[16] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[17] Marko Djordjevic, Anirvan M. Sengupta, and Boris I. Shraiman. A biophysical approach to transcription factor binding site discovery. *Genome Research*, 13(11):2381–2390, 2003.

[18] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[19] Denitsa Eckweiler, Christian-Alexander Dudek, Juliane Hartlich, David Brötje, and Dieter Jahn. PRODORIC2: the bacterial gene regulation database in 2018. *Nucleic acids research*, 46(D1):D320–D326, 2018.

[20] Ralf Eggeling, Ivo Grosse, and Jan Grau. InMoDe: tools for learning and visualizing intra-motif dependencies of DNA binding sites. *Bioinformatics (Oxford, England)*, 33(4):580–582, 2017.

[21] Ralf Eggeling, Teemu Roos, Petri Myllymäki, and Ivo Grosse. Inferring intra-motif dependencies of DNA binding sites from ChIP-seq data. *BMC Bioinformatics*, 16(1):375, 2015.

[22] Dominik M. Endres and Johannes E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003.

[23] Ensembl. The Ensembl regulatory build. 2015. URL: `http://www.ensembl.org/info/genome/funcgen/regulatory_build.html`.

[24] Laurence Ettwiller, Benedict Paten, Marcel Souren, Felix Loosli, Jochen Wittbrodt, and Ewan Birney. The discovery, positioning and verification of a set of transcription-associated motifs in vertebrates. *Genome biology*, 6(12):R104–R104, 2005.

[25] Iakes Ezkurdia, David Juan, Jose Manuel Rodriguez, Adam Frankish, Mark Diekhans, Jennifer Harrow, Jesus Vazquez, Alfonso Valencia, and Michael L. Tress. Multiple evidence strands suggest that there may be as few as 19 000 human protein-coding genes. *Human Molecular Genetics*, 23(22):5866–5878, 2014.

[26] Alexander V. Favorov, Mikhail S. Gelfand, Anna V. Gerasimova, Dmitry A. Ravcheev, Alexandre A. Mironov, and Vsevolod J. Makeev. A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length. *Bioinformatics*, 21(10):2240–2245, 2005.

[27] Barrett C. Foat, Alexandre V. Morozov, and Harmen J. Bussemaker. Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE. *Bioinformatics*, 22(14):e141–e149, 2006.

[28] Charles E. Grant, Timothy L. Bailey, and William Stafford Noble. FIMO: scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, 2011.

[29] Qiye He, Jeff Johnston, and Julia Zeitlinger. ChIP-nexus enables improved detection of in vivo transcription factor binding footprints. *Nature biotechnology*, 33(4):395–401, 2015.

[30] John M. Heumann, Alan S. Lapedes, and Gary D. Stormo. Neural networks for determining protein specificity and multiple alignment of binding sites. In *ISMB*, pages 188–194, 1994.

[31] Peter Huggins, Shan Zhong, Idit Shiff, Rachel Beckerman, Oleg Laptenko, Carol Prives, Marcel H. Schulz, Itamar Simon, and Ziv Bar-Joseph. DECOD: fast and accurate discriminative DNA motif finding. *Bioinformatics*, 27(17):2361–2367, 2011.

[32] Hisaki Ikebata and Ryo Yoshida. Repulsive parallel MCMC algorithm for discovering diverse motifs from large sequence sets. *Bioinformatics*, 31(10):1561–1568, 2015.

[33] Aleksander Jankowski, Shyam Prabhakar, and Jerzy Tiuryn. TACO: a general-purpose tool for predicting cell-type–specific transcription factor dimers. *BMC Genomics*, 15(1):1–12, 2014.

[34] Arttu Jolma, Jian Yan, Thomas Whitington, Jarkko Toivonen, et al. DNA-binding specificities of human transcription factors. *Cell*, 152(1–2):327–339, 2013.

[35] Arttu Jolma, Yimeng Yin, Kazuhiro R. Nitta, Kashyap Dave, Alexander Popov, Minna Taipale, Martin Enge, Teemu Kivioja, Ekaterina Morgunova, and Jussi Taipale. DNA-dependent formation of transcription factor pairs alters their binding specificity. *Nature*, 527(7578): 384–388, 2015.

[36] Arttu Jolma et al. Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. *Genome Res.*, 20(6):861–873, 2010.

[37] Juhani Kähärä and Harri Lähdesmäki. Evaluating a linear k-mer model for protein-DNA interactions using high-throughput SELEX data. *BMC Bioinformatics*, 14(10):S2, 2013.

[38] Majid Kazemian, Hannah Pham, Scot A. Wolfe, Michael H. Brodsky, and Saurabh Sinha. Widespread evidence of cooperative DNA binding by transcription factors in Drosophila development. *Nucleic Acids Res.*, 41(17):8237–8252, 2013.

[39] Janne H. Korhonen, Kimmo Palin, Jussi Taipale, and Esko Ukkonen. Fast motif matching revisited: high-order PWMs, SNPs and indels. *Bioinformatics*, 33(4):514–521, 2017.

[40] Janne Korhonen, Petri Martinmäki, Cinzia Pizzi, Pasi Rastas, and Esko Ukkonen. MOODS: fast search for position weight matrix matches in DNA sequences. *Bioinformatics*, 25(23):3181–3182, 2009.

[41] Samuel A. Lambert, Arttu Jolma, Laura F. Campitelli, Pratyush K. Das, Yimeng Yin, Mihai Albu, Xiaoting Chen, Jussi Taipale, Timothy R. Hughes, and Matthew T. Weirauch. The human transcription factors. *Cell*, 172(4):650–665, 2018.

[42] Kirsti Laurila, Olli Yli-Harja, and Harri Lähdesmäki. A protein–protein interaction guided method for competitive transcription factor binding improves target predictions. *Nucleic acids research*, 37(22): e146–e146, 2009.

[43] Charles E. Lawrence, 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(5131):208–214, 1993.

[44] Charles E. Lawrence 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(1):41–51, 1990.

[45] Leping Li. GADEM: a genetic algorithm guided formation of spaced dyads coupled with an EM algorithm for motif discovery. *Journal of Computational Biology*, 16(2):317–329, 2009.

[46] Xiaole Liu, Douglas L. Brutlag, Jun S. Liu, et al. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. In *Pacific symposium on biocomputing*, volume 6, pages 127–138, 2001.

[47] Ruipeng Lu, Eliseos J. Mucaki, and Peter K. Rogan. Discovery and validation of information theory-based transcription factor and cofactor binding site motifs. *Nucleic Acids Res.*, 45(5):e27, 2017.

[48] Geoff Macintyre, James Bailey, Izhak Haviv, and Adam Kowalczyk. is-rSNP: a novel technique for in silico regulatory SNP detection. *Bioinformatics*, 26(18):i524–i530, 2010.

[49] Volker Matys et al. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, 34:D108–D110, 2006.

[50] Judith A. McClarin, Christin A. Frederick, Bi-Cheng Wang, Patricia Greene, Herbert W. Boyer, John Grable, and John M. Rosenberg. Structure of the DNA-Eco RI endonuclease recognition complex at 3 Å resolution. *Science*, 234(4783):1526–1541, 1986.

[51] Ekaterina Morgunova, Yimeng Yin, Pratyush K Das, Arttu Jolma, Fangjie Zhu, Alexander Popov, You Xu, Lennart Nilsson, and Jussi Taipale. Two distinct DNA sequences recognized by transcription factors represent enthalpy and entropy optima. *eLife*, 7:e32963, 2018.

[52] Ekaterina Morgunova, Yimeng Yin, Arttu Jolma, Kashyap Dave, Bernhard Schmierer, Alexander Popov, Nadejda Eremina, Lennart Nilsson, and Jussi Taipale. Structural insights into the DNA-binding specificity of E2F family transcription factors. *Nature Communications*, 6:10050, 2015.

[53] Kazuhiro R. Nitta et al. Conservation of transcription factor binding specificities across 600 million years of bilateria evolution. *eLife*, 4: e04837, 2015.

[54] Arnold R. Oliphant, Christopher J. Brandl, and Kevin Struhl. Defining the sequence specificity of DNA-binding proteins by selecting binding sites from random-sequence oligonucleotides: analysis of yeast GCN4 protein. *Mol. Cell. Biol.*, 9(7):2944–2949, 1989.

[55] Saeed Omidi, Mihaela Zavolan, Mikhail Pachkov, Jeremie Breda, Severin Berger, and Erik van Nimwegen. Automated incorporation of pairwise dependency in transcription factor binding site prediction using dinucleotide weight tensors. *PLOS Computational Biology*, 13(7): 1–22, 2017.

[56] Bioinformatics Organization. IUPAC codes. 2000. URL: `https://www.bioinformatics.org/sms/iupac.html`.

[57] Raphael Pelossof, Irtisha Singh, Julie L. Yang, Matthew T. Weirauch, Timothy R. Hughes, and Christina S. Leslie. Affinity regression predicts the recognition code of nucleic acid–binding proteins. *Nature Biotechnology*, 33:1242–1249, 2015.

[58] Mark M. Pomerantz et al. The 8q24 cancer risk variant rs6983267 shows long-range interaction with MYC in colorectal cancer. *Nature Genetics*, 41:882–884, 2009.

[59] Daniel Quang and Xiaohui Xie. EXTREME: an online EM algorithm for motif discovery. *Bioinformatics*, 30(12):1667–1673, 2014.

[60] John E. Reid and Lorenz Wernisch. STEME: a robust, accurate motif finder for large data sets. *PLoS One*, 9(3):e90735, 2014.

[61] Ho Sung Rhee and B. Franklin Pugh. Comprehensive genome-wide protein-DNA interactions detected at single-nucleotide resolution. *Cell*, 147(6):1408–1419, 2011.

[62] Todd R. Riley, Allan Lazarovici, Richard S. Mann, and Harmen J. Bussemaker. Building accurate sequence-to-affinity models from high-throughput in vitro protein-DNA binding data using FeatureREDUCE. *eLife*, 4:e06397, 2015.

[63] Alberto Riva. Large-scale computational identification of regulatory SNPs with rSNP-MAPPER. *BMC Genomics*, 13(4):S7, 2012.

[64] Gordon Robertson et al. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods*, 4:651–657, 2007.

[65] Shuxiang Ruan, S. Joshua Swamidass, and Gary D. Stormo. BEESEM: estimation of binding energy models using HT-SELEX data. *Bioinformatics*, 33(15):2288–2295, 2017.

[66] M. Santillan and M. C. Mackey. Dynamic regulation of the tryptophan operon: a modeling study and comparison with experimental data. *Proceedings of the National Academy of Sciences of the United States of America*, 98(4):1364–1369, 2001.

[67] Thomas D. Schneider, Gary D. Stormo, Larry Gold, and Andrzej Ehrenfeucht. Information content of binding sites on nucleotide sequences. *Journal of Molecular Biology*, 188(3):415–431, 1986.

[68] Thomas Schneider and R. Michael Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res.*, 18(20):6097–6100, 1990.

[69] Gideon Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 1978.

[70] Eilon Sharon, Shai Lubliner, and Eran Segal. A feature-based approach to modeling protein-DNA interactions. *PLOS Computational Biology*, 4(8):1–17, 2008.

[71] Zhen Shen, Wenzheng Bao, and De-Shuang Huang. Recurrent neural network for predicting transcription factor binding sites. *Scientific Reports*, 8(1):15270, 2018.

[72] Matthias Siebert and Johannes Söding. Bayesian Markov models consistently outperform PWMs at predicting motifs in nucleotide sequences. *Nucleic acids research*, 44(13):6055–6069, 2016.

[73] Saurabh Sinha and Martin Tompa. A statistical method for finding transcription factor binding sites. In *ISMB*, volume 8, pages 344–354, 2000.

[74] Adrian F. A. Smit, Robert Hubley, and Phil Green. RepeatMasker Open-4.0. 2015. URL: `http://www.repeatmasker.org`.

[75] John D. Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3): 479–498, 2002.

[76] Gary D. Stormo, Thomas D. Schneider, and Larry Gold. Quantitative analysis of the relationship between nucleotide sequence and functional activity. *Nucleic acids research*, 14(16):6661–6679, 1986.

[77] Gary D. Stormo, Thomas D. Schneider, Larry Gold, and Andrzej Ehrenfeucht. Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli. *Nucleic Acids Res.*, 10(9):2997–3011, 1982.

[78] Morgane Thomas-Chollier, Andrew Hufton, Matthias Heinig, Sean O'Keeffe, Nassim El Masri, Helge G. Roider, Thomas Manke, and Martin Vingron. Transcription factor binding predictions using TRAP for the analysis of ChIP-seq data and regulatory SNPs. *Nature Protocols*, 6:1860–1869, 2011.

[79] William Thompson, Eric C. Rouchka, and Charles E. Lawrence. Gibbs Recursive Sampler: finding transcription factor binding sites. *Nucleic acids research*, 31(13):3580–3585, 2003.

[80] Craig Tuerk and Larry Gold. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science*, 249(4968):505–510, 1990.

[81] Sari Tuupanen et al. The common colorectal cancer predisposition SNP rs6983267 at chromosome 8q24 confers potential to enhanced Wnt signaling. *Nature Genetics*, 41:885–890, 2009.

[82] Juan M. Vaquerizas, Sarah K. Kummerfeld, Sarah A. Teichmann, and Nicholas M. Luscombe. A census of human transcription factors: function, expression and evolution. *Nature Reviews Genetics*, 10(4): 252–263, 2009.

[83] Todd Wasson and Alexander J. Hartemink. An ensemble model of competitive multi-factor binding of the genome. *Genome research*, 19(11):2101–2112, 2009.

[84] Tom Whitington, Martin C. Frith, James Johnson, and Timothy L. Bailey. Inferring transcription factor complexes from ChIP-seq data. *Nucleic Acids Res.*, 39(15):e98, 2011.

[85] Christopher T. Workman, Yutong Yin, David L. Corcoran, Trey Ideker, Gary D. Stormo, and Panayiotis V. Benos. enoLOGOS: a versatile web tool for energy normalized sequence logos. *Nucleic Acids Research*, 33(suppl_2):W389–W392, 2005.

[86] Mengyuan Xu, Clarice R. Weinberg, David M. Umbach, and Leping Li. coMOTIF: a mixture framework for identifying transcription factor and a coregulator motif in ChIP-seq data. *Bioinformatics*, 27(19): 2625–2632, 2011.

[87] Haoyang Zeng, Matthew D. Edwards, Ge Liu, and David K. Gifford. Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics*, 32(12):i121–i127, 2016.

[88] Daniel R. Zerbino, Steven P. Wilder, Nathan Johnson, Thomas Juettemann, and Paul R. Flicek. The Ensembl regulatory build. *Genome biology*, 16(1):56, 2015.

[89] Daniel R. Zerbino et al. Ensembl 2018. *Nucleic Acids Research*, 46(D1): D754–D761, 2017.

[90] ZhiZhuo Zhang, Cheng Wei Chang, Willy Hugo, Edwin Cheung, and Wing-Kin Sung. Simultaneously learning DNA motif along with its position and sequence rank preferences through expectation maximization algorithm. *Journal of Computational Biology*, 20(3):237–248, 2013.

[91] Yue Zhao, David Granas, and Gary D. Stormo. Inferring binding energies from selected binding sites. *PLOS Computational Biology*, 5(12):1–8, 2009.

[92] Chandler Zuo, Sunyoung Shin, and Sündüz Keleş. atSNP: transcription factor binding affinity testing for regulatory SNP detection. *Bioinformatics*, 31(20):3353–3355, 2015.

TIETOJENKÄSITTELYTIETEEN OSASTO
PL 68 (Pietari Kalmin katu 5)
00014 Helsingin yliopisto

JULKAISUSARJA **A**

DEPARTMENT OF COMPUTER SCIENCE
P.O. Box 68 (Pietari Kalmin katu 5)
FI-00014 University of Helsinki, Finland

SERIES OF PUBLICATIONS **A**

Reports are available on the e-thesis site of the University of Helsinki.

A-2014-1 J. Korhonen: Graph and Hypergraph Decompositions for Exact Algorithms. 62+66 pp. (Ph.D. Thesis)

A-2014-2 J. Paalasmaa: Monitoring Sleep with Force Sensor Measurement. 59+47 pp. (Ph.D. Thesis)

A-2014-3 L. Langohr: Methods for Finding Interesting Nodes in Weighted Graphs. 70+54 pp. (Ph.D. Thesis)

A-2014-4 S. Bhattacharya: Continuous Context Inference on Mobile Platforms. 94+67 pp. (Ph.D. Thesis)

A-2014-5 E. Lagerspetz: Collaborative Mobile Energy Awareness. 60+46 pp. (Ph.D. Thesis)

A-2015-1 L. Wang: Content, Topology and Cooperation in In-network Caching. 190 pp. (Ph.D. Thesis)

A-2015-2 T. Niinimäki: Approximation Strategies for Structure Learning in Bayesian Networks. 64+93 pp. (Ph.D. Thesis)

A-2015-3 D. Kempa: Efficient Construction of Fundamental Data Structures in Large-Scale Text Indexing. 68+88 pp. (Ph.D. Thesis)

A-2015-4 K. Zhao: Understanding Urban Human Mobility for Network Applications. 62+46 pp. (Ph.D. Thesis)

A-2015-5 A. Laaksonen: Algorithms for Melody Search and Transcription. 36+54 pp. (Ph.D. Thesis)

A-2015-6 Y. Ding: Collaborative Traffic Offloading for Mobile Systems. 223 pp. (Ph.D. Thesis)

A-2015-7 F. Fagerholm: Software Developer Experience: Case Studies in Lean-Agile and Open Source Environments. 118+68 pp. (Ph.D. Thesis)

A-2016-1 T. Ahonen: Cover Song Identification using Compression-based Distance Measures. 122+25 pp. (Ph.D. Thesis)

A-2016-2 O. Gross: World Associations as a Language Model for Generative and Creative Tasks. 60+10+54 pp. (Ph.D. Thesis)

A-2016-3 J. Määttä: Model Selection Methods for Linear Regression and Phylogenetic Reconstruction. 44+73 pp. (Ph.D. Thesis)

A-2016-4 J. Toivanen: Methods and Models in Linguistic and Musical Computational Creativity. 56+8+79 pp. (Ph.D. Thesis)

A-2016-5 K. Athukorala: Information Search as Adaptive Interaction. 122 pp. (Ph.D. Thesis)

A-2016-6 J.-K. Kangas: Combinatorial Algorithms with Applications in Learning Graphical Models. 66+90 pp. (Ph.D. Thesis)

A-2017-1 Y. Zou: On Model Selection for Bayesian Networks and Sparse Logistic Regression. 58+61 pp. (Ph.D. Thesis)

A-2017-2 Y.-T. Hsieh: Exploring Hand-Based Haptic Interfaces for Mobile Interaction Design. 79+120 pp. (Ph.D. Thesis)

A-2017-3 D. Valenzuela: Algorithms and Data Structures for Sequence Analysis in the Pan-Genomic Era. 74+78 pp. (Ph.D. Thesis)

A-2017-4 A. Hellas: Retention in Introductory Programming. 68+88 pp. (Ph.D. Thesis)

A-2017-5 M. Du: Natural Language Processing System for Business Intelligence. 78+72 pp. (Ph.D. Thesis)

A-2017-6 A. Kuosmanen: Third-Generation RNA-Sequencing Analysis: Graph Alignment and Transcript Assembly with Long Reads. 64+69 pp. (Ph.D. Thesis)

A-2018-1 M. Nelimarkka: Performative Hybrid Interaction: Understanding Planned Events across Collocated and Mediated Interaction Spheres. 64+82 pp. (Ph.D. Thesis)

A-2018-2 E. Peltonen: Crowdsensed Mobile Data Analytics. 100+91 pp. (Ph.D. Thesis)

A-2018-3 O. Barral: Implicit Interaction with Textual Information using Physiological Signals. 72+145 pp. (Ph.D. Thesis)

A-2018-4 I. Kosunen: Exploring the Dynamics of the Biocybernetic Loop in Physiological Computing. 91+161 pp. (Ph.D. Thesis)

A-2018-5 J. Berg: Solving Optimization Problems via Maximum Satisfiability: Encodings and Re-Encodings. 86+102 pp. (Ph.D. Thesis)

A-2018-6 J. Pyykkö: Online Personalization in Exploratory Search. 101+63 pp. (Ph.D. Thesis)

A-2018-7 L. Pivovarova: Classification and Clustering in Media Monitoring: from Knowledge Engineering to Deep Learning. 78+56 pp. (Ph.D. Thesis)

A-2019-1 K. Salo: Modular Audio Platform for Youth Engagement in a Museum Context. 97+78 pp. (Ph.D. Thesis)

A-2019-2 A. Koski: On the Provisioning of Mission Critical Information Systems based on Public Tenders. 96+79 pp. (Ph.D. Thesis)

A-2019-3 A. Kantosalo: Human-Computer Co-Creativity - Designing, Evaluating and Modelling Computational Collaborators for Poetry Writing. 74+86 pp. (Ph.D. Thesis)

A-2019-4 O. Karkulahti: Understanding Social Media through Large Volume Measurements. 116 pp. (Ph.D. Thesis)

A-2019-5 S. Yaman: Initiating the Transition towards Continuous Experimentation: Empirical Studies with Software Development Teams and Practitioners. 81+90 pp. (Ph.D. Thesis)

A-2019-6 N. Mohan: Edge Computing Platforms and Protocols. 87+69 pp. (Ph.D. Thesis)

A-2019-7 I. Järvinen: Congestion Control and Active Queue Management During Flow Startup. 87+48 pp. (Ph.D. Thesis)

A-2019-8 J. Leinonen: Keystroke Data in Programming Courses. 56+53 pp. (Ph.D. Thesis)

A-2019-9 T. Talvitie: Counting and Sampling Directed Acyclic Graphs for Learning Bayesian Networks. 70+54 pp. (Ph.D. Thesis)