Discriminative Bayesian principles for predicting sequence signals of gene regulation

Dissertation zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften (Dr. rer. nat.)

der Naturwissenschaftlichen Fakultät III

der Martin-Luther-Universität Halle–Wittenberg

vorgelegt von

Jan Grau

geboren am 12.11.1979 in Bremen

Halle (Saale), April 2010

Gutachter:
1. Prof. Dr.-Ing. Stefan Posch
2. Prof. Dr. Alexander Schliep

Datum der Verteidigung: 15. Juli 2010

Danksagung

Mein besonderer Dank gilt Professor Dr.-Ing. Stefan Posch, der mir einerseits große Freiheit bei der Auswahl der Themen ließ und andererseits durch Rat und kritische Fragen deren Bearbeitung voranbrachte. Der Gedankenaustausch und die Diskussionen, unter anderem bei unseren wöchentlichen Treffen, haben maßgeblich zur Fertigstellung dieser Dissertation beigetragen. Ich hätte mir keine bessere Betreuung wünschen können.

Ich möchte mich weiterhin bei Jens Keilwagen bedanken. Unsere Zusammenarbeit an diversen Fragestellungen war stets inspirierend, gewinnbringend und sehr produktiv. Seine Begeisterung für die Wissenschaft gab mir an vielen Stellen die Motivation, noch einen Schritt weiter zu gehen. Auch für seine Bereitschaft, direkt nach der Abgabe seiner eigenen Dissertation Teile der meinigen korrekturzulesen, möchte ich mich bedanken.

Ebenso danke ich Professor Dr. Ivo Große für Anregungen und Kritik, für seine ansteckende Begeisterungsfähigkeit und für seine Bereitschaft, sich jederzeit in spannende Diskussionen verwickeln zu lassen.

Weiterhin möchte ich mich bei den Mitgliedern der Arbeitsgruppen “Bioinformatik und Mustererkennung” und “Bioinformatik” des Instituts für Informatik und der Arbeitsgruppe “Bioinformatik & Massenspektrometrie” des IPB Halle für Ihre Unterstützung während der Erstellung der Dissertation danken. Insbesondere die Diskussionen in unserem gemeinsamen Forschungsseminar waren dabei sehr fruchtbar. Besonders bedanken möchte ich mich bei Dr. Birgit Möller und Oliver Greß für das Korrekturlesen großer Teile der Dissertation. Mein Dank gilt auch allen übrigen Kolleginnen und Kollegen am Institut für Informatik, die eine Atmosphäre schufen, in der ich gerne gearbeitet habe.

Bei Professor Dr. Alexander Schliep möchte ich mich für die Bereitschaft, das Zweitgutachten zu meiner Dissertation zu erstellen, bedanken.

Meinen Freunden danke ich für die Unterstützung, die Aufmunterungen und die gemeinsamen Zeiten der Entspannung, die mich immer wieder Kraft schöpfen ließen.

Besonders danke ich meinen Eltern und meinen Schwestern für den emotionalen Rückhalt während meines Studiums und der Arbeit an dieser Dissertation.

Abschließend möchte ich mich bei meiner Freundin Yvonne Pöschl für Ihre Unterstützung über die gesamte Zeit der Erstellung dieser Dissertation, insbesondere aber während des letzten halben Jahres bedanken. Sie und unser gemeinsamer Sohn Noah hatten während der Endphase der Dissertation viel Geduld mit mir und ich hoffe, bald die Gelegenheit zu haben, etwas von dieser Geduld und Unterstützung zurückgeben zu können.

Contents

List of Figures

List of Tables

1. Introduction

The fundamental process of gene expression is well-established: the DNA-sequence of a gene is transcribed to messenger RNA (mRNA) by a DNA-dependent RNA polymerase and, in turn, mRNA is translated to a poly-peptide, which may then fold to a protein. Regulatory mechanisms affect each stage of this process, and as a consequence the final amount of product. With advancing research, our picture of gene regulation becomes more complete – and more complex.

The initiation of transcription is mediated by transcription factors, which bind to specific sites on the DNA and enhance or inhibit transcription of a gene to mRNA. Transcription factors often bind coordinately to cis-regulatory modules comprising several transcription factor binding sites (Jeziorska et al., 2009). In eukaryotes, the binding of transcription factors competes with a structural element of chromosomal DNA, namely nucleosomes (Narlikar et al., 2007). In nucleosomes, a stretch of approximately 147 bp of DNA is wound around a histone octamer, which makes this stretch virtually inaccessible to transcription factors due to steric hindrance. Hence, nucleosomes may influence transcriptional regulation besides their primary role in the compaction of chromatin. While originally a single transcription start site (TSS) was assumed, it has become evident during the last years that multiple, alternative TSS may exist for one gene (Mitchell et al., 1986; Roni et al., 2007). Additionally, the TATA box, which is typically located in close vicinity to the TSS, appears to be less important for transcription than assumed in the past. Today, 80 to 90 percent of the promoters of eukaryotic genes are expected to be TATA-less (Gershenzon and Ioshikhes, 2005).

In eukaryotes, the synthesized pre-mRNA is further processed. This co-transcriptional and post-transcriptional processing includes the capping of the 5' end and poly-adenylation of the 3' end, and splicing of the pre-mRNA. Splicing constitutes the excision of introns from the primary transcript, while the remaining exons are joined to form the mature mRNA. Splicing is predominantly accomplished by RNA-protein complexes called spliceosomes, although self-splicing introns are known as well. Introns are terminated by a donor splice site at the 5' end and an acceptor splice site at the 3' end, which are recognized in the splicing process. Today it is known that the splicing process in eukaryotes is not deterministic. So called alternative splicing includes the skipping of complete exons or retention of introns, and the use of alternative donor and acceptor splice sites (Black, 2003). Hence, a single gene may encode for multiple proteins.

Recently, another post-transcriptional mode of gene regulation has gained increased interest, namely microRNAs (Enright et al., 2003; John et al., 2004). MicroRNAs are short endogenous RNA molecules that bind to mRNA in plants and animals, and cause the degradation of the transcript or a repression of its translation. Gene expression may additionally be affected by the rate of translation. Due to the wobble position of codons, multiple species of tRNAs supply

the same amino acids. Since these tRNAs are differently abundant in the cell, the choice of the wobble nucleotide may influence the rate of translation (Man and Pilpel, 2007).

It can be expected that this is not the end of the story. The Encode project found that most bases of the human genome are part of at least one primary transcript (The ENCODE Project Consortium, 2007). This is in stark contrast to the estimation that only approximately 1.5% of the human genome code for proteins, and that approximately 2.5% of the transcripts have a known function, while most bases of the human genome have been considered "junk DNA". Hence, the function of the majority of existing transcripts is still to be elucidated.

Bioinformatics, especially statistical sequence analysis, plays an important role in elucidating all these regulatory processes. Since wet-lab experiments are time-consuming and expensive, *in silico* analyses are often the only feasible way for studying gene regulation on a genomic scale. Bioinformatics approaches have been developed for predicting transcription start sites, for recognizing binding sites of known transcription factors, for de-novo discovery of transcription factor binding sites and cis-regulatory modules, for predicting nucleosome positioning, for recognizing donor and acceptor splice sites and predicting alternative splicing, and for predicting targets of microRNAs.

However, none of these approaches is perfect and the falsification of false-positive predictions by wet-lab experiments is often of little interest and leads to frustration. In the field of statistical sequence analysis, two main directions are investigated to improve computational predictions and reduce the number of false-positives. The first focuses on the development of more sophisticated and appropriate models of the sites of interest, whereas the second concentrates on improved learning principles for determining the parameters of these models. In this work, we mainly follow the second direction, although we adapt the employed statistical models to the characteristics of the given biological phenomena to some extend as well.

Historically, generative learning principles like the maximum likelihood or maximum a-posteriori principle have been applied to problems of bioinformatics very early and are still the prevalent principles of parameter estimation. While generative learning principles focus on an accurate representation of the data, discriminative learning principles are tailored to an accurate classification. For most applications, the model that generated the data is unknown and presumably no statistical model might fully reflect the underlying biological process. In this work, we investigate the utility of a Bayesian discriminative learning principle termed *maximum supervised posterior* for different applications from the field of statistical sequence analysis. We employ the maximum supervised posterior principle for the prediction of transcription factor binding sites, donor splice sites, miRNA target sites, and nucleosome positioning, and for the de-novo discovery of cis-regulatory modules.

For the prediction of transcription factor binding sites and donor splice sites, we adapt existing models, namely Markov models and decision tree models, to discriminative parameter learning. For de-novo discovery of single motifs and cis-regulatory modules comprising binding sites of two motifs, we present a novel approach that combines discriminative learning with an explicit model of the position distribution of binding sites. We propose a new approach using an extended ensemble approach for the prediction of nucleosome positioning, which can

incorporate discrete sequence information as well as numerical properties of DNA. For the prediction of miRNA target sites, we extend profile HMMs to explicitly model the complementary basepairing in the miRNA-mRNA duplex, which is also learned discriminatively.

The maximum supervised posterior principle requires numerical optimization of parameters. However, none of the statistical models and densities employed for the above mentioned approaches is suited for unconstrained numerical optimization in its standard parameterization. Hence, we must derive parameter transformations that map between constrained standard parameters and unconstrained parameters that can directly be optimized numerically. Since the Bayesian maximum supervised posterior principle incorporates a prior on the parameters of the employed models, we must also transform conjugate prior densities according to the parameter transformations to make them applicable to the unconstrained parameters. The corresponding transformation of the parameters of Markov models and the conjugate Dirichlet prior and a subset of the methods for de-novo discovery of single motifs have been developed and published (Keilwagen et al., 2010b,a) in close collaboration with Jens Keilwagen. In this collaboration, we also derived some of the foundations of the discriminative learning of decision tree models.

This work is structured as follows: In the following chapter, we give a more detailed overview to the biological aspects of regulatory mechanisms that are relevant for this work. In the "Methods" chapter, we introduce the discriminative maximum supervised posterior principle, the employed statistical models and densities, and other concepts that are of general importance for all applications. The chapter "Applications" comprises sections for the specific applications of the maximum supervised posterior principle. Each of these sections gives a short overview of the bioinformatic background and related work, describes the methods and data that are specific for the given application, and finally presents and discusses the results of experiments and – except for the prediction of microRNA targets – benchmark studies. After a short chapter describing the implementation of the studied models, learning principles, and algorithms, we conclude the work with an assessment of the utility of the maximum supervised posterior for statistical sequence analysis and classification.

2. From gene to product

This chapter gives a concise overview of biological processes that determine the product of a gene and the rate of its production. Here, we focus on biological processes and aspects of their mechanisms that are relevant for the computational approaches presented in the remainder of this work.

2.1. Transcription factors and cis-regulatory modules

Transcription factors are proteins that bind to specific DNA signals in the promoter region of a gene and enhance or repress the transcription of that gene. Several families of transcription factors with different structural properties exists. Figure 2.1 shows x-ray structures of two different transcription factors binding to a DNA double helix. The structural feature of GATA displayed on the left are two zinc fingers, each including an alpha helix with contact to the DNA and a complexed zinc ion. C/EBP shown on the right belongs to the family of leucine zipper transcription factors which exhibit two zipper-like alpha helices as DNA binding domains. Other families of transcription factors include homeo domain factors, helix-turn-helix factors, or beta-scaffold factors.

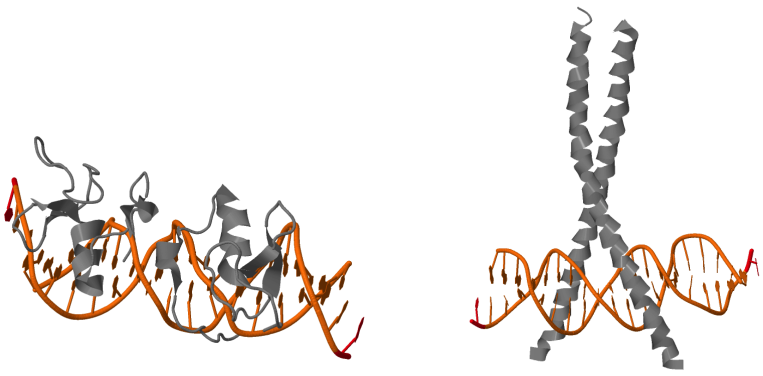


Figure 2.1.: X-ray structures of the zinc finger transcription factor GATA (left) and the leucine zipper transcription factor C/EBP (right) bound to the DNA helix (orange). Both structures are obtained from the protein data bank at `http://www.pdb.org` (Berman et al., 2000) with accessions 1NWQ (C/EBP, Miller et al. (2003)) and 3DFV (GATA, Bates et al. (2008))

Figure 2.2 outlines the organization of a eukaryotic promoter. General transcription factors, like the TATA binding protein, facilitate the formation of the transcription initiation complex including RNA polymerase. These are located within the core promoter in close vicinity to the transcription start. In contrast, specific transcription factors, which are responsible for complex patterns of regulation, may bind in great distance to the transcription start: specific

enhancers are reported several thousand basepairs upstream of the transcription start (Levine and Tjian, 2003). However, most transcription factor binding sites can be found in a maximum distance of 500 to 1000 bp from the transcription start (Kim et al., 2008), which is a reasonable region for computational de-novo discovery with regard to statistics and computation time.

In higher eukaryotes, transcription factor binding sites are often organized in cis-regulatory modules. Cis-regulatory modules comprise several binding sites of a set of transcription factors that bind coordinately to regulate the transcription of a gene. By this means, the number and complexity of specific regulatory patterns that can be achieved by a limited set of transcription factors is greatly increased, which is one of the reasons for the diversity of metazoans despite the surprisingly low number of genes (Levine and Tjian, 2003).

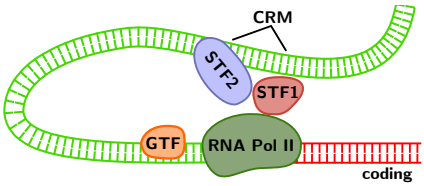


Figure 2.2.: Schematic overview of a eukaryotic promoter. RNA polymerase II binds at the transcription start site. A general transcription factor (GTF) bind in close proximity of the transcription start, while two specific transcription factors (STF1 and STF2) bind coordinately to a cis-regulatory module in great distance to the transcription start site.

Since transcription factors are proteins that are encoded by genes themselves, one transcription factor may activate or repress the expression of other transcription factors. By this means, a regulatory cascade may be initiated by a set of factors, e.g. as a reaction to external stimuli like stress factors or pathogens. This also has the effect that genes that are co-expressed under given conditions may be regulated by different factors that are part of the same regulatory pathway, which complicates the computational de-novo discovery of transcription factor binding sites.

2.2. Nucleosome depletion

The foremost purpose of nucleosomes is the compaction of eukaryotic chromatin. In each nucleosome $\sim$ 147 bp of DNA are wound in 1.67 super-helical turns around a histone octamer as depicted in figure 2.3. The histone octamer consist of one tetramer comprising two copies of each of the core histones H3 and H4, and two dimers of the core histones H2A and H2B. Each histone in the octamer is a protein with a helix-turn-helix-turn-helix motif.

The affinity of DNA to the histone core depends on structural features of the DNA including the specific bases that are in contact with the histones and long-range properties like the bendability of the DNA double helix. Each DNA strand is in contact with the histone core every 10 bp due to the helical turn, which is most likely the origin of $\sim$ 10 bp periodic patterns that are observed for A/T dinucleotides or geometrical properties like tip (Richmond and Davey, 2003; Segal et al., 2006). Such features of DNA can be used by computational approaches to predict the positioning of nucleosomes on a chromosome from DNA sequence (Miele et al., 2008; Segal et al., 2006; Field et al., 2008; Yuan and Liu, 2008).

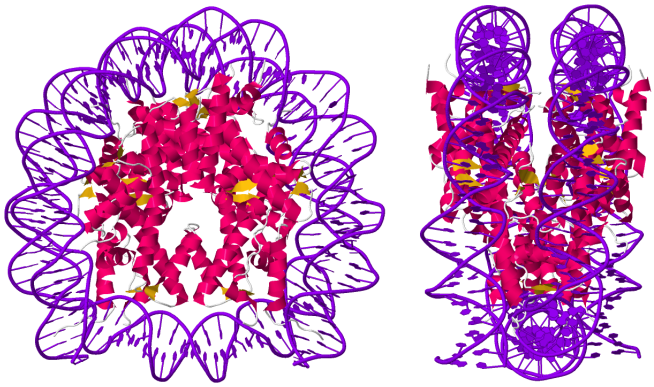


Figure 2.3.: X-ray structure of DNA wound around a histone octamer in a nucleosome. The structure is obtained from the protein data bank, accession 2NZD (Ong et al., 2007).

The arrangement of nucleosomes on the DNA is often visualized as "beads on a string". However, the "beads" are not evenly spaced along chromosomes. The spacing and local clustering of nucleosomes is related to the function of a region of DNA. Protein-coding regions are often highly occupied by nucleosomes and the level of occupancy correlates with the rate of transcription (Lee et al., 2007). In contrast, regulatory active regions in promoters are depleted of nucleosomes and for many genes an exceptionally low nucleosome occupancy can be observed approximately 100 bp upstream of the transcription start site (Lee et al., 2007; Field et al., 2008). Hence, an accurate map of nucleosome positioning can guide the prediction of biologically functional transcription factor binding sites (Narlikar et al., 2007; Ucar et al., 2009).

2.3. Splicing

The spliceosome, which catalyzes splicing in eukaryotes, comprises five small nuclear ribonucleoprotein particles (snRNPs), namely U1, U2, U4, U5, and U6, and additional proteins. Each snRNP is a complex of proteins and one small nuclear RNA (snRNA). The binding between snRNPs and the pre-mRNA is accomplished by complementary basepairing between the snRNA and the pre-mRNA. Three sites on the pre-mRNA are directly involved in the splicing process: the donor splice site at the 5' end of the intron, the acceptor splice site at the 3' end of the intron, and the branch-point located within the intron.

An overview of the splicing process is depicted in figure 2.4. Splicing is initiated by a binding of U1 to the donor splice site at the 5' end of the intron. In this step, the snRNA of U1 recognizes the first six nucleotides of the intron, including the consensus GT at the first two positions of the intron in case of canonical donor splice sites. The branch-point is recognized by the snRNA of U2. After the binding of U2 to the branch-point, a complex of U4, U5, and U6 additionally binds to the donor splice site via a recognition site in the snRNA of U6. This binding depends on the nucleotides at the last two positions of the exons, commonly referred to as position -1 and -2, the consensus G at the first position of the intron, and positions $+4$ to $+6$ on the intron side. Since the binding sites of U6 and U1 overlap, a strong binding

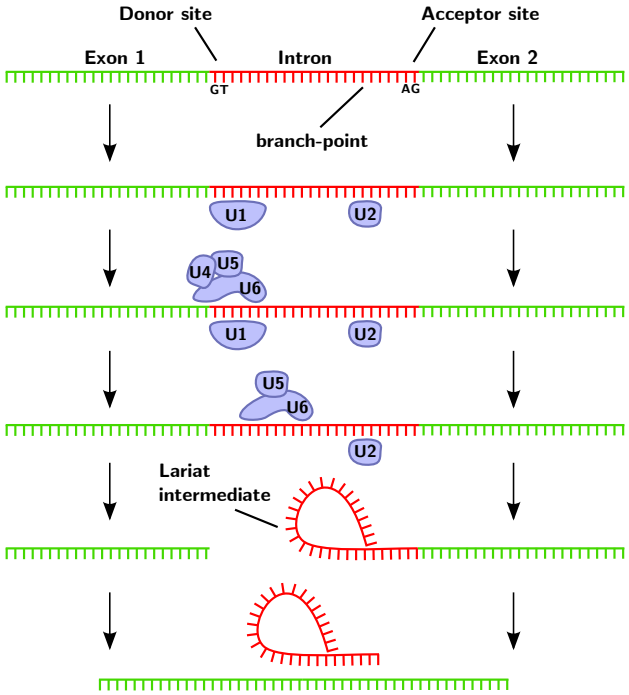


Figure 2.4.: Overview of the splicing process mediated by snRNPs U1, U2, U4, U5, and U6 of the spliceosome.

of U1 may inhibit the binding of the U4/U5/U6 complex and as a consequence the complete splicing process (Brow, 2002).

In the following step, U1 and U4 are released from the spliceosome and U6 is shifted towards the intron. The recognition site of the U6 snRNA, which was previously located at positions -2 to $+1$ of the pre-mRNA, now binds to positions $+4$ to $+6$. Due to an interaction between U6 bound to the donor splice site and U2 bound to the branch-point, both regions are brought into close vicinity, which facilitates a first transesterification resulting in a lariat intermediate. In the lariat intermediate, the bond between exon and intron is replaced by a binding of the 5' end of the intron to a consensus A at the branch-point. In a second transesterification, the two ends of the exons are joined and the lariat intron is released (Brow, 2002).

Against this biological background, the recognition of donor splice sites by computational approaches corresponds to predicting the binding sites of U1 and U6 at positions -2 to $+6$ at the 5' end of the intron.

2.4. Degradation and translational silencing induced by microRNAs

MicroRNAs (miRNAs) are transcribed from DNA and undergo maturation before they become functional. Transcription of miRNAs is most likely accomplished by RNA polymerase II, which is also responsible for transcribing protein-coding genes. The maturation of the transcript is illustrated in figure 2.5. In animals, the primary transcript called pri-miRNA folds into a stem-loop structure – also referred to as hairpin – flanked by unpaired RNA sequences on both sides of the stem. These are removed by an RNA endonuclease called Drosha yielding

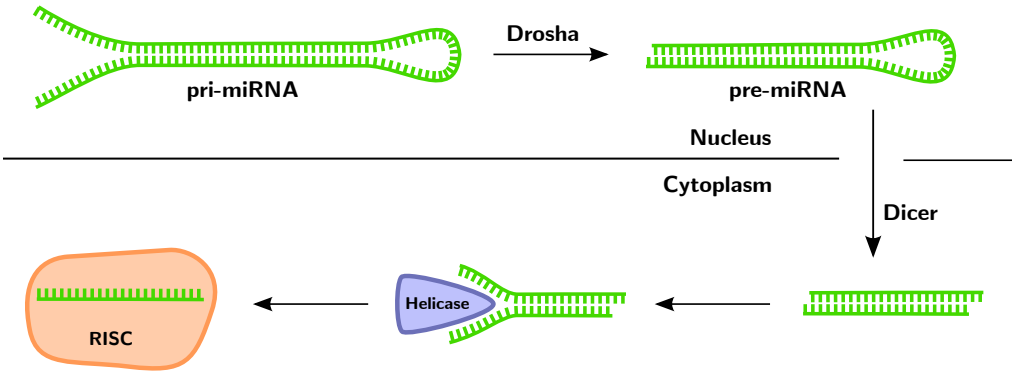


Figure 2.5.: Maturation of pri-miRNA to the final miRNA in metazoans. The trimming of the pri-miRNA is accomplished by two enzymes, Drosha and Dicer. After the double-stranded miRNA duplex is separated by Helicase, the miRNA is loaded into RISC.

the stem-loop structure of the pre-miRNA. The pre-miRNA is transported from the nucleus into the cytoplasm, where the loop structure is removed by another endonuclease called Dicer. The two strands of the remaining RNA duplex, which corresponds to the stem of the original stem-loop structure, are then separated by a helicase yielding the mature miRNA of $\sim$ 22 nt length. The strand of the miRNA duplex that corresponds to the mature miRNA depends on the side of the duplex which can be separated with less effort by the helicase. The maturation of miRNAs in plants proceeds in similar steps but requires other enzymes than for animals (Bartel, 2004; Ghosh et al., 2007).

The mature miRNA is loaded into the RNA-induced silencing complex (RISC), which is required for the function of the miRNA. RISC may down-regulate the expression of a target gene either by cleavage of the mRNA or by translational repression. The target gene is determined by complementarity to the loaded miRNA. If the miRNA is highly complementary to the target site on the target gene, the mRNA will be cleaved in a mechanism assumed to be similar to that of small interfering RNAs (siRNAs). Otherwise, RISC represses translation of the target gene (Bartel, 2004). However, the exact mechanism of RISC is still unknown.

Target sites of miRNAs in plants and animals show different characteristics. In plants, the miRNA sequence is highly complementary to the target site and target sites are predominantly found in coding regions of genes. In contrast, miRNA targets in animals are often located in the 3' UTRs of target genes and require perfect complementarity only in a seed region of $\sim$ 7 nt at the 5' end of the miRNA. Hence, translational repression instead of mRNA cleavage seems to be the prevalent mechanism of miRNAs in animals (Bartel, 2004; Lewis et al., 2005). As Bartel (2004) points out, the preference for 3' UTRs observed in animals could also be an artifact, since the first miRNA target site was found in a 3' UTR and guided research to this region.

3. Methods

This chapter presents the foundations of discriminative learning of statistical models. It defines discriminative learning principles in section 3.2, Markov models for the representation of discrete sequences in section 3.3.1, and densities for modelling continuous data in section 3.3.2. As we use Bayesian approaches for all of the applications, we also define priors on the parameters of Markov models and densities in section 3.4.

3.1. Statistical foundation and notation

All problems treated in the following can be considered as classification problems: We obtain a data set $\boldsymbol{X}$ of N sequences $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, i.e. $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$. Each of these sequences $\boldsymbol{x}_n$ is defined over an alphabet Σ, which in case of DNA sequences is $\Sigma = \{A, C, G, T\}$. For a sequence of length L this can be formalized by $\boldsymbol{x}_n \in \Sigma^L$. The goal is to assign each sequence $\boldsymbol{x}_n$ to the *correct* class c from a pre-defined set of admissible classes $\mathcal{C}, |\mathcal{C}| = K$. For instance, sets of such classes could be transcription factor binding sites and non-binding sites, splice donor sites and non-donor sites, or sequences bound in nucleosomes and linker sequences.

A common setting in statistical sequence classification is to learn probabilistic models with parameters $\boldsymbol{\theta}$ from a given *training data set* of sequences $\boldsymbol{x}_n$ and associated class labels c_n. We denote by $\boldsymbol{c} = (c_1, c_2, \ldots, c_N)$ the vector of the correct classes for each of the sequences in the data set $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$. For classification, the learned model is applied to new sequences and each sequence is commonly assigned to that class yielding the maximum class posterior $P(c \,|\, \boldsymbol{x}, \boldsymbol{\theta})$

$$\begin{aligned} c^* &= \operatorname*{argmax}_c P(c \,|\, \boldsymbol{x}, \boldsymbol{\theta}) \\ &= \operatorname*{argmax}_c P(\boldsymbol{x}, c \,|\, \boldsymbol{\theta}), \end{aligned} \tag{3.1}$$

where $P(\boldsymbol{x}, c \,|\, \boldsymbol{\theta})$ denotes the *likelihood* of sequence $\boldsymbol{x}$ and class c given parameters $\boldsymbol{\theta}$. The functional form of the likelihood depends on the chosen statistical model, distribution, or density.

Results of classification and classification accuracy highly depend on the principle which is used to estimate the parameters $\boldsymbol{\theta}$. One of the most prevalent principles of parameter estimation in bioinformatics applications is *maximum likelihood* (ML) estimation, which has been used in a variety of applications, e.g. the computational prediction of transcription factor binding sites and splice sites (Staden, 1984; Zhang and Marr, 1993; Salzberg, 1997; Burge, 1998; Yeo

and Burge, 2004). Among all possible parameter values, ML chooses those that maximize the likelihood $P(\boldsymbol{X}, \boldsymbol{c} \,|\, \boldsymbol{\theta})$

$$\boldsymbol{\theta}^*_{\mathrm{ML}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} P(\boldsymbol{X}, \boldsymbol{c} \,|\, \boldsymbol{\theta}), \tag{3.2}$$

which decomposes to the product of independent likelihoods for each of the sequences in case of independent, identically distributed (i.i.d.) data, i.e.

$$\overset{i.i.d.}{=} \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{n=1}^{N} P(\boldsymbol{x}_n, c_n \,|\, \boldsymbol{\theta}). \tag{3.3}$$

Despite its popularity, ML estimation entails certain disadvantages from a Bayesian perspective as well as under practical considerations. First, it disregards uncertainty in parameter estimation induced by the limited size of the training data set, which is the case for many bioinformatics applications. And it is prone to over-fitting if the training data are too limited, e.g. if certain events cannot be observed in the training data although they might be possible in general. Second, it does not allow for including a-priori knowledge about the parameters into parameter estimation. These disadvantages may be overcome – at least to some extent – by imposing a prior $p(\boldsymbol{\theta} \,|\, \boldsymbol{\alpha})$ on the parameters, where $\boldsymbol{\alpha}$ denotes the hyper-parameters of the prior density, leading to the posterior $P(\boldsymbol{\theta} \,|\, \boldsymbol{X}, \boldsymbol{c}, \boldsymbol{\alpha})$ of the parameters $\boldsymbol{\theta}$ given the training data $\boldsymbol{X}$ and $\boldsymbol{c}$, and hyper-parameters $\boldsymbol{\alpha}$:

$$P(\boldsymbol{\theta} \,|\, \boldsymbol{X}, \boldsymbol{c}, \boldsymbol{\alpha}) \;=\; \frac{P(\boldsymbol{X}, \boldsymbol{c} \,|\, \boldsymbol{\theta})\, p(\boldsymbol{\theta} \,|\, \boldsymbol{\alpha})}{P(\boldsymbol{X}, \boldsymbol{c})} \tag{3.4}$$

Maximum a-posteriori (MAP) estimation optimizes the parameters with respect to this posterior

$$\begin{aligned} \boldsymbol{\theta}^*_{\mathrm{MAP}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} P(\boldsymbol{\theta} \,|\, \boldsymbol{X}, \boldsymbol{c}, \boldsymbol{\alpha}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} P(\boldsymbol{X}, \boldsymbol{c} \,|\, \boldsymbol{\theta})\, p(\boldsymbol{\theta} \,|\, \boldsymbol{\alpha}), \end{aligned} \tag{3.5}$$

hence, searching for the most likely parameter values when considering the training data as well as the a-prior knowledge represented by the hyper-parameters $\boldsymbol{\alpha}$. If the parameters of the likelihoods for the different classes are independent, MAP and ML estimation of these parameters can be carried out independently on the data stemming from the corresponding class.

ML and MAP estimation are called *generative* principles (Bishop, 2006), as they aim at an accurate description of the distribution of the training sequences $\boldsymbol{X}$ and the associated classes $\boldsymbol{c}$. Hence, both principles are appropriate if the goal is to obtain a model that can generate new data which are most similar to the original training data under the constraints of the chosen statistical model.

However, neither of the generative approaches does directly optimize the parameters with respect to the classification task. Hence, the classification accuracy achieved by generative

approaches may stay behind possibilities when training data are limited. This is the motivation for defining *discriminative* principles as presented in the next section.

3.2. Discriminative objective functions

Support vector machines (SVMs) (Cortes and Vapnik, 1995; Smola and Schölkopf, 1998) are probably the most widespread discriminative learning method in bioinformatics. SVMs have been applied to many problems of sequence classification, e.g. the recognition of transcription start sites (Sonnenburg et al., 2006) and translation initiation sites (Meinicke et al., 2004), gene finding (Schweikert et al., 2009), the prediction of transcription factor binding sites (Jiang et al., 2007), or de-novo motif discovery (Schultheiss et al., 2009). The power of SVMs highly depends on the chosen *kernel*, which maps the input data to a, usually higher dimensional, feature space, where samples of the two classes can be separated by a linear *hyper-plane*. In case of sequence classification, SVMs – depending on the application – may exhibit two potential disadvantages: First, SVMs have originally been defined for two-class problems, and multi-class problems must be mapped to a number of two-class problems to fit the SVM framework. Second, the weights of an SVM are less easy to interpret than e.g. the probabilities of a probabilistic model, even though recent approaches like POIMs (Sonnenburg et al., 2008) have improved the interpretability of SVMs.

Another discriminative learning method, namely *logistic regression* (Ng and Jordan, 2002), has gained increased attention in bioinformatics during the last years. Like SVMs, logistic regression was originally proposed for general machine learning problems. In the field of bioinformatics, it has been applied to the prediction of nucleosome positioning (Yuan and Liu, 2008), the regulation of genes by sigma transcription factors (de Hoon et al., 2004), or the analysis of microarray data (Liao and Chin, 2007). Logistic regression defines the class posterior $P\left(c=1 \,|\, \boldsymbol{x}, \boldsymbol{\beta}\right), c \in \{1,2\}$ by the logistic function applied to the dot product of a real-valued parameter vector $\boldsymbol{\beta}$ and the input features $\boldsymbol{x}$:

$$P\left(c=1 \,|\, \boldsymbol{x}, \boldsymbol{\beta}\right) \;=\; \frac{1}{1+\exp\left(-\boldsymbol{\beta}^T \boldsymbol{x}\right)} \tag{3.6}$$

Logistic regression in its original definition entails disadvantages similar to those of SVMs: Again, the parameters are real-valued and hence not always intuitively interpretable, and it is only applicable to two-class problems. *Soft max* (Heckerman and Meek, 1997) (also called *multinomial logistic regression* (Cawley et al., 2007) or *multiclass logistic regression* (Bishop, 2006)) extends logistic regression to multiple classes but retains real-valued parameters:

$$P\left(c \,|\, \boldsymbol{x}, \boldsymbol{\beta}\right) \;=\; \frac{\exp\left(\boldsymbol{\beta}_c^T \boldsymbol{x}\right)}{\sum_{\tilde{c}} \exp\left(\boldsymbol{\beta}_{\tilde{c}}^T \boldsymbol{x}\right)} \tag{3.7}$$

The definition of *conditional likelihood* presented in the next section can be understood as a generalization of soft max, where the dot product may be replaced by more complex and potentially non-linear functions of parameters $\boldsymbol{\beta}$ defined by different classes of statistical models. The *supervised posterior* presented in section 3.2.2 additionally imposes a prior on the

parameters and hence can be seen as a Bayesian variant of conditional likelihood and the discriminative analogon to the posterior.

3.2.1. Maximum conditional likelihood for class-conditional statistical models

The *maximum conditional likelihood* (MCL) principle (Greiner and Zhou, 2001; Wettig et al., 2003; Grossman and Domingos, 2004; Roos et al., 2005) aims at finding those parameters $\boldsymbol{\theta}$ that maximize the *conditional likelihood* (CL) $P(\boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\theta})$ of the correct class labels $\boldsymbol{c}$ given the training sequences $\boldsymbol{X}$ and parameters $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^*_{\mathrm{MCL}} \quad = \quad \operatorname*{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\theta}) \tag{3.8}$$

Again, we assume that all data points $(\boldsymbol{x}_n, c_n)$ are independent and identically distributed. Therefore, the conditional likelihood $P(\boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\theta})$ can be expressed as the product of independent class posteriors $P(c_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$, i.e.

$$P(\boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\theta}) \quad = \quad \prod_n^N P(c_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) \,. \tag{3.9}$$

Comparing equation (3.9) to the classification criterion (3.1), we observe that MCL is closely linked to the classification task. Hence, we anticipate that parameters $\boldsymbol{\theta}^*_{\mathrm{MCL}}$ learned by MCL may lead to a more accurate classification than those learned by generative principles.

In contrast to logistic regression or soft max, we now define the class posterior $P(c \mid \boldsymbol{x}, \boldsymbol{\theta})$ based on the likelihoods $P(\boldsymbol{x}, c \mid \boldsymbol{\theta})$ for each of the classes.

$$P(c \mid \boldsymbol{x}, \boldsymbol{\theta}) \quad = \quad \frac{P(\boldsymbol{x}, c \mid \boldsymbol{\theta})}{P(\boldsymbol{x} \mid \boldsymbol{\theta})} = \frac{P(\boldsymbol{x}, c \mid \boldsymbol{\theta})}{\sum_{\tilde{c}} P(\boldsymbol{x}, \tilde{c} \mid \boldsymbol{\theta})} \tag{3.10}$$

The definition of the class posterior in terms of class-dependent likelihoods allows for using any statistical model in the discriminative conditional likelihood principle that can also be employed for the generative principles. This includes popular discrete sequence models, like position weight matrices (Staden, 1984; Stormo et al., 1982), weight array models (Zhang and Marr, 1993), or higher order Markov models (Yakhnenko et al., 2005; Keilwagen et al., 2007; Grau et al., 2007b) – introduced in section 3.3.1 – with multinomial likelihood.

In analogy to models working on discrete input, we may apply this definition of conditional likelihood to densities $p(\boldsymbol{y}, c \mid \boldsymbol{\theta})$ for sequences $\boldsymbol{y} = y_1 y_2 \ldots y_L$ of continuous values $y_\ell \in \mathbb{R}$, yielding

$$P(c \mid \boldsymbol{y}, \boldsymbol{\theta}) \quad = \quad \frac{p(\boldsymbol{y}, c \mid \boldsymbol{\theta})}{\sum_{\tilde{c}} p(\boldsymbol{y}, \tilde{c} \mid \boldsymbol{\theta})}. \tag{3.11}$$

All results derived in this section about the conditional likelihood for discrete data are readily transferred to continuous data as well.

The optimization of the parameters according to equation (3.8) cannot be carried out analytically for any of the statistical models considered in this work. Hence, we must resort to

numerical optimization techniques, e.g. gradient ascent, conjugate gradients, or second-order quasi-Newton methods (Wallach, 2004). However, these optimization techniques work on unconstrained parameter values, and methods like log-barrier functions (Guo et al., 2005) would be necessary to limit the allowed values of a parameter to e.g. $[0, 1]$. For large numbers of parameters, this would on the one hand complicate the optimization problem, and on the other hand potentially abolish useful properties of the objective function as for instance concavity.

Hence, we choose an alternative approach by deriving unconstrained optimization problems: For all parameters, including those that must be limited to some interval, we define a transformation $\boldsymbol{t}(\boldsymbol{\beta})$ from parameters $\boldsymbol{\beta} \in \mathbb{R}^D$ that are allowed to sweep over the reals to the constrained parameters $\boldsymbol{\theta} \in \mathbb{P} \subsetneq \mathbb{R}^D$, i.e. $\boldsymbol{t} : \mathbb{R}^D \to \mathbb{P}$. As a result, we can now optimize the objective function with respect to the unconstrained parameters $\boldsymbol{\beta}$. If we need the original parameters $\boldsymbol{\theta}$, e.g. because these are easier to interpret, they can be obtained by applying the transformation $\boldsymbol{\theta} = \boldsymbol{t}(\boldsymbol{\beta})$. In case of the MCL principle defined in equation (3.8), we can replace the parameters $\boldsymbol{\theta}$ by the transformed parameters $\boldsymbol{t}(\boldsymbol{\beta})$, and obtain

$$\begin{aligned} \boldsymbol{\beta}^*_{\mathrm{MCL}} &= \operatorname*{argmax}_{\boldsymbol{\beta}} P\left(\boldsymbol{c} \,|\, \boldsymbol{X}, \boldsymbol{t}(\boldsymbol{\beta})\right) \\ &=: \operatorname*{argmax}_{\boldsymbol{\beta}} P\left(\boldsymbol{c} \,|\, \boldsymbol{X}, \boldsymbol{\beta}\right) \end{aligned} \tag{3.12}$$

We call $\boldsymbol{\beta}^*_{\mathrm{MCL}}$ the MCL estimate of the parameters $\boldsymbol{\beta}$. As for any objective function, we can equivalently optimize the parameters with respect to the *log* CL

$$\begin{aligned} \boldsymbol{\beta}^*_{\mathrm{MCL}} &= \operatorname*{argmax}_{\boldsymbol{\beta}} \log P\left(\boldsymbol{c} \,|\, \boldsymbol{X}, \boldsymbol{\beta}\right), \\ &= \operatorname*{argmax}_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log P\left(c_n \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}\right) \end{aligned} \tag{3.13}$$

which often results in more tractable gradients and less numerical problems for optimization.

Like ML estimation, the MCL principle tends to over-fit to the training data if these are of limited size and it does not allow for including a-priori knowledge into parameter estimation. The problem of over-fitting may be even more severe for MCL than it is for the generative principles (Ng and Jordan, 2002). In part, this can be explained by the comparably large number of parameters used to model a small domain, i.e. the possible classes. In the ML case, the number of parameters is the same, but the modelled domain is significantly larger being all classes *and* all possible sequences in these classes.

This problem is addressed by a Bayesian approach called *maximum supervised posterior* (Wettig et al., 2002; Grünwald et al., 2002; Cerquides and de Mántaras, 2005) presented in the next section.

3.2.2. Maximum supervised posterior

The maximum supervised posterior (MSP) principle is defined in analogy to the step from ML to MAP estimation. Instead of CL alone, we now optimize the parameters with respect to a

product of CL and a prior $q(\boldsymbol{\beta} \mid \boldsymbol{\alpha})$ on the parameters $\boldsymbol{\beta}$ with hyper-parameters $\boldsymbol{\alpha}$.

$$\begin{aligned} \boldsymbol{\beta}^*_{\mathrm{MSP}} &= \underset{\boldsymbol{\beta}}{\operatorname{argmax}}\, P(\boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\beta})\, q(\boldsymbol{\beta} \mid \boldsymbol{\alpha}) \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left[\prod_n^N P(c_n \mid \boldsymbol{x}_n, \boldsymbol{\beta}) \right] q(\boldsymbol{\beta} \mid \boldsymbol{\alpha}) \end{aligned} \tag{3.14}$$

Priors $q(\boldsymbol{\beta} \mid \boldsymbol{\alpha})$ that are originally defined on real-valued parameters $\boldsymbol{\beta} \in \mathbb{R}^D$ are Gaussian and Laplace priors, which will be presented in detail in section 3.4.1. However, a number of popular and often conjugate priors for many families of distribution are defined on parameters $\boldsymbol{\theta} \in \mathbb{P}^D \subsetneq \mathbb{R}^D$, which cannot be optimized by unconstrained numerical optimization techniques. For instance, in case of Markov models, this is the widely used Dirichlet prior (section 3.4.2), while in case of Gaussian likelihoods, these are normal-gamma or – in the multi-variate case – normal-Wishart densities (section 3.4.3).

It is worthwhile to use these priors for the discriminative MSP principle as well for two reasons: First, the use of equivalent priors for MAP and MSP allows for an unbiased comparison of the classification accuracy observed for both principles, since the influence of different priors on the classification result is eliminated (Keilwagen et al., 2010b). Second, these priors are known to be conjugate to the likelihood of the employed model. Hence, the priors and their hyper-parameters can often be interpreted intuitively as additional observations from a set of pseudo-data.

In section 3.2.1, we introduced a transformation $\boldsymbol{t}(\boldsymbol{\beta})$ from $\boldsymbol{\beta}$ to $\boldsymbol{\theta}$. We can use $\boldsymbol{t}(\boldsymbol{\beta})$ to transform a prior $p(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$ to a prior $q(\boldsymbol{\beta} \mid \boldsymbol{\alpha})$ defined on parameters $\boldsymbol{\beta} \in \mathbb{R}^D$. Following the substitution rule for integrals, we obtain

$$q(\boldsymbol{\beta} \mid \boldsymbol{\alpha}) = p(\boldsymbol{t}(\boldsymbol{\beta}) \mid \boldsymbol{\alpha}) \left| \det \left(\frac{\partial \boldsymbol{t}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \right) \right|, \tag{3.15}$$

where $det\left(\frac{\partial \boldsymbol{t}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right)$ is the Jacobian of $\boldsymbol{t}$.

3.2.3. Soft-labelling

In some applications, we have no exact knowledge about the class-membership of the input data. Rather, we assign each input sequence $\boldsymbol{x}_n$ a probability $w_{n,c} \in [0, 1]$ of belonging to class c, where $\forall n = 1, \ldots, N : \sum_{\tilde{c}} w_{n,\tilde{c}} = 1$. The goal is, to learn parameters $\boldsymbol{\beta}$ such that for all sequences $\boldsymbol{x}_n$ in the training data and all classes c, the class posterior $P(c \mid \boldsymbol{x}_n, \boldsymbol{\beta})$ is as close as possible to the corresponding probability $w_{n,c}$.

A common choice for classification problems with soft-labelling is to minimize the mean squared error (MSE) between the probabilities $w_{n,c}$ and the corresponding class posteriors $P(c \mid \boldsymbol{x}_n, \boldsymbol{\beta})$:

$$\boldsymbol{\beta}^*_{\mathrm{MSE}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{KN} \sum_{n=1}^{N} \sum_{c=1}^{K} \left(P(c \mid \boldsymbol{x}_n, \boldsymbol{\beta}) - w_{n,c} \right)^2 \tag{3.16}$$

When using logistic regression (equation (3.6)) for the class posterior, this model is equivalent to a neural network composed of a single neuron with a logistic activation function. Since MSE is a common cost function for neural networks, learning the parameters β could be accomplished by standard algorithms for neural networks. For a single neuron this would amount to gradient descent or similar methods.

However, MSE is not directly linked to the probabilistic nature of our formulation of the class posterior (equation (3.10)), and it might be worthwhile to search for other, probabilistic objective functions that can incorporate soft-labelling. One desirable property of such an objective function might be, that it degrades to conditional likelihood if the probabilities $w_{n,c}$ are either 0 or 1, i.e. in case of hard-labelling. Another property might be, that it utilizes that the vector $\boldsymbol{w}_n = (w_{n,1}, \ldots, w_{n,K})$ is a probability vector , i.e. $w_{n,k} \in [0,1]$ and $\sum_k w_{n,k} = 1$, and $P(c \,|\, \boldsymbol{x}_n, \boldsymbol{\beta})$ is a discrete probability distribution on the classes c.

Here, we propose an objective function which exhibits both properties and which can be optimized by the same algorithms that we use for standard MCL. We define

$$\begin{aligned} \boldsymbol{\beta}^*_{\mathrm{MCL}} &= \operatorname*{argmax}_{\boldsymbol{\beta}} \sum_{n=1}^{N} \sum_{\tilde{c}=1}^{K} w_{n,\tilde{c}} \log P(\tilde{c} \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}) \,. && (3.17) \\ &=: \operatorname*{argmax}_{\boldsymbol{\beta}} \log \mathrm{CL}(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{\beta}), && (3.18) \end{aligned}$$

where $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N)$ in analogy to the definition of $\boldsymbol{c}$. If we know the correct class labels exactly, i.e. if the probabilities are defined as

$$w_{n,\tilde{c}} = \begin{cases} 1 & \text{if } \tilde{c} = c_n \\ 0 & \text{otherwise} \end{cases},$$

we can replace the weights $w_{n,\tilde{c}}$ by Kronecker deltas $\delta_{a,b}$, which are 1 if $a = b$ and 0 otherwise, resulting in

$$\begin{aligned} \boldsymbol{\beta}^*_{\mathrm{MCL}} &= \operatorname*{argmax}_{\boldsymbol{\beta}} \sum_{n=1}^{N} \sum_{\tilde{c}=1}^{K} \delta_{\tilde{c},c_n} \log P(\tilde{c} \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}) && (3.19) \\ &= \operatorname*{argmax}_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log P(c_n \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}) && (3.20) \\ &= \operatorname*{argmax}_{\boldsymbol{\beta}} \log P(\boldsymbol{c} \,|\, \boldsymbol{X}, \boldsymbol{\beta}) \,. && (3.21) \end{aligned}$$

We see from the last line that we obtain the original definition of MCL of equation (3.13) in this case, and, hence, fulfill the first property.

If we augment equation (3.17) by a constant with respect to the parameters $\boldsymbol{\beta}$

$$\boldsymbol{\beta}^*_{\mathrm{MCL}} = \operatorname*{argmax}_{\boldsymbol{\beta}} \sum_{n=1}^{N} \sum_{\tilde{c}=1}^{K} \left(w_{n,c} \log P(\tilde{c} \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}) - w_{n,c} \log w_{n,c} \right), \quad (3.22)$$

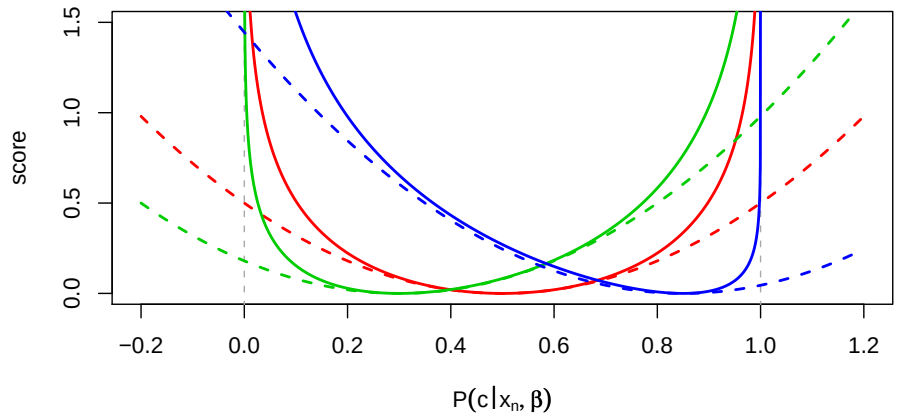


Figure 3.1.: Comparison of MSE (dashed lines) and Kullback-Leibler divergence for a two-class problem with $w_{n,1} = 0.5$ (red), $w_{n,1} = 0.3$ (green), and $w_{n,1} = 0.85$ (blue).

and invert the sign

$$\boldsymbol{\beta}^*_{\mathrm{MCL}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{n=1}^{N} \sum_{\tilde{c}=1}^{K} \left(-w_{n,c} \log P\left(\tilde{c} \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}\right) + w_{n,c} \log w_{n,c}\right), \tag{3.23}$$

we find that this objective function actually minimizes the sum of the Kullback-Leibler divergences (Kullback and Leibler, 1951) D_{KL} between $\boldsymbol{w}_n$ and the probability distribution $P\left(c \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}\right)$ for sequence $\boldsymbol{x}_n$:

$$\boldsymbol{\beta}^*_{\mathrm{MCL}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{n=1}^{N} D_{\mathrm{KL}}(\boldsymbol{w}_n || P\left(c \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}\right)) \tag{3.24}$$

Kullback-Leibler divergence is a common choice (Ben-Gal et al., 2005; Gunewardena and Zhang, 2008; Ellrott et al., 2002; Keles et al., 2003) to measure the divergence of two probability distributions, which are $\boldsymbol{w}_n$ and $P\left(c \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}\right)$ in this case, and accordingly fulfills the second property. Hence, it appears to be an appropriate measure for learning with soft labels as well.

We illustrate Kullback-Leibler divergence for a two-class problem and a single input sequence $\boldsymbol{x}_n$ in figure 3.1, and we compare Kullback-Leibler divergence to MSE for different values of the probability of the first class $w_{n,1}$. From figure 3.1, we observe that Kullback-Leibler divergence is defined only for admissible values of $P\left(c \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}\right)$ and values approaching the limits of 0 or 1 are strongly disfavored. In contrast, the parabolic characteristic of MSE disregards the constraints on $P\left(c \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}\right)$. However, both objective functions show a similar characteristic in the vicinity of the optima.

Considering equation (3.17) and comparing it to equation (3.13), we observe that this objective function corresponds to MCL estimation of the parameters $\boldsymbol{\beta}$ using weighted input data. Each sequence $\boldsymbol{x}_n$ serves as an input for each of the classes $c = 1, \ldots, K$ weighted by the probability

$w_{n,c}$ for class c. The only modification that is necessary to utilize the probabilities $w_{n,c}$ in the numerical optimization is to extend conditional likelihood (and its gradients) to weighted data.

This weighted variant of conditional likelihood can be multiplied by a prior to obtain the supervised posterior in the same manner as for conditional likelihood without weights. We define the weighted variant of the maximum supervised posterior principle as

$$\boldsymbol{\beta}^*_{\mathrm{MSP}} \quad = \quad \operatorname{argmax} \left[\log \mathrm{CL}(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{\beta}) + \log q\left(\boldsymbol{\beta} \,|\, \boldsymbol{\alpha}\right) \right]. \tag{3.25}$$

3.3. Statistical models and densities

Up to now, the likelihood $P\left(\boldsymbol{x}, c \,|\, \boldsymbol{\beta}\right)$ has been utilized as an abstract placeholder, which can be replaced by different probability distributions or densities depending on the assumptions made about the statistical characteristics of the data. In this section, we introduce Markov models, which define a family of likelihoods on discrete sequences. In case of continuous random variables, we consider univariate and multivariate Gaussian densities as well as Gamma densities. Since we want to use these for the discriminative MSP principle, we also define specific parameter transformations that allow for unconstrained numerical optimization of the parameters.

3.3.1. Discrete random variables: Markov models

We start the derivation of Markov models with the general decomposition of the likelihood $P\left(\boldsymbol{x}, c \,|\, \boldsymbol{\phi}\right)$ defined on constrained parameters $\boldsymbol{\phi}$. Following Bayes, we can decompose the likelihood as

$$P\left(\boldsymbol{x}, c \,|\, \boldsymbol{\phi}\right) \quad = \quad P(c|\boldsymbol{\phi}) P_1(x_1|c, \boldsymbol{\phi}) \prod_{\ell=2}^{L} P_\ell(x_\ell|x_1, \ldots, x_{\ell-1}, c, \boldsymbol{\phi}), \tag{3.26}$$

where the (conditional) probability distributions $P_\ell(x_\ell|x_1, \ldots, x_{\ell-1}, c, \boldsymbol{\phi})$ may depend on the current position ℓ.

A Markov model of order d_c assumes that for class c the probability of symbol x_ℓ at position ℓ in the sequence $\boldsymbol{x}$ does not depend on all preceding symbols $x_1, \ldots, x_{\ell-1}$ but only on the last d_c symbols called the *context*:

$$P\left(\boldsymbol{x}, c \,|\, \boldsymbol{\phi}\right) \quad = \quad P(c|\boldsymbol{\phi}) P_1(x_1|c, \boldsymbol{\phi}) \prod_{\ell=2}^{L} P_\ell(x_\ell|x_{\ell-d_c}, \ldots, x_{\ell-1}, c, \boldsymbol{\phi}), \tag{3.27}$$

where x_i is the empty string iff $i \leq 0$.

If the probabilities $P_\ell(x_\ell|x_{\ell-d_c}, \ldots, x_{\ell-1}, c, \boldsymbol{\phi})$ are position-independent, i.e. $\forall \ell, k = (d_c + 1), \ldots, L : P_\ell(a|\boldsymbol{b}, c, \boldsymbol{\phi}) = P_k(a|\boldsymbol{b}, c, \boldsymbol{\phi}), a \in \Sigma, \boldsymbol{b} \in \Sigma^{d_c}$, we call the Markov models *homogeneous* and *inhomogeneous* otherwise. Markov models belong to the class of graphical models. The dependencies assumed by Markov models can be represented by a directed acyclic graph

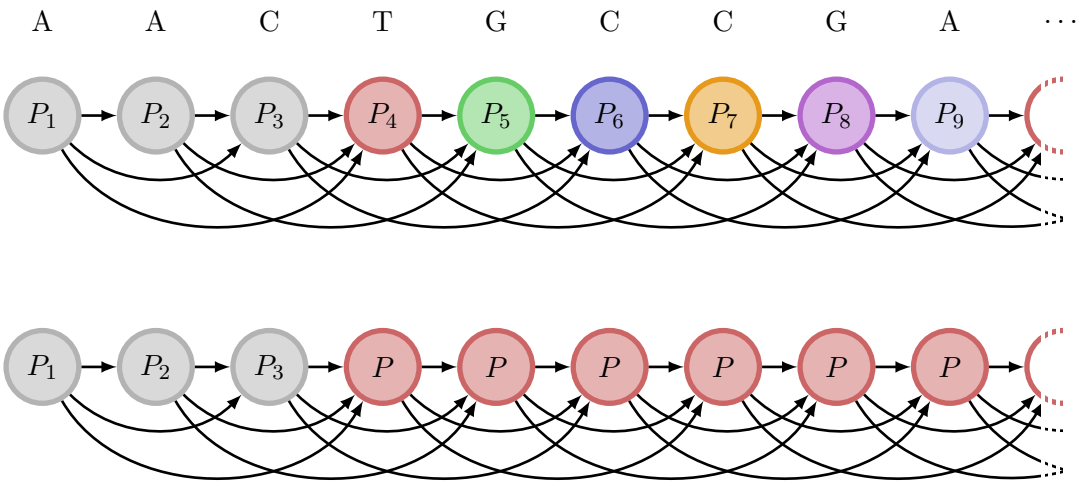


Figure 3.2.: DAG structure representing an inhomogeneous (top) and a homogeneous (bottom) Markov model of order $d_c = 3$. Both models exhibit a position-dependent initial distribution at the first three positions. While the conditional distributions at the remaining positions remain position-dependent for the inhomogeneous Markov model as indicated by the coloring of nodes, we use the same conditional probability distribution for the rest of the sequences in case of the homogeneous Markov model.

(DAG) with nodes representing the random variables at positions ℓ and directed edges representing the possible dependencies. Figure 3.2 depicts the graph structure of an inhomogeneous and a homogeneous Markov model of order $d_c = 3$. Both models exhibit a position-dependent initial distribution at the first three positions. The coloring of nodes indicates the (in-) homogeneity of the conditional probability distributions at positions 4 and above. The probability distributions of the inhomogeneous model remain positions dependent for the other positions, whereas the homogeneous model assumes that the distributions at positions $\ell \geq 4$ are identical. In both cases, the conditional probability distribution at position $\ell \geq 4$ depends on the symbols observed at positions $\ell - 3$ through $\ell - 1$.

The class of Markov models includes two models, which are widely used in bioinformatics applications, namely the *position weight matrix* (PWM) (Stormo et al., 1982; Staden, 1984), and the *weight array* model (WAM) (Zhang and Marr, 1993; Salzberg, 1997). The PWM model is an inhomogeneous Markov model of order $d_c = 0$ and assumes that the nucleotides at all positions in the sequence are drawn independently. The WAM model is an inhomogeneous Markov model of order $d_c = 1$ and assumes that the probability of symbol x_ℓ at position ℓ depends only on the symbol observed at position $\ell - 1$ and the class c. PWM models and WAM models have been used for the prediction of transcription factor binding sites (Staden, 1984; Kel et al., 2003; Chekmenev et al., 2005) and cis-regulatory modules (Berman et al., 2002; Pape et al., 2009), translation initiation sites (Stormo et al., 1982), nucleosome positions (Segal et al., 2006), and splice sites (Staden, 1984; Zhang and Marr, 1993; Salzberg, 1997). PWM models are also used for de-novo motif discovery (Bailey and Elkan, 1994; Thompson et al., 2003; Redhead and Bailey, 2007).

3.3.1.1. Inhomogeneous Markov models

In the following, we formalize Markov models using constrained parameters $\boldsymbol{\phi}$. We define parameters ϕ_c representing the probability of the classes c, parameters $\phi_{1,a|c}$ for the probability of observing symbol a at position 1 in the sequence given class c, and parameters $\phi_{\ell,a|\boldsymbol{b},c}$ for observing symbol a at position ℓ given the observation of $\boldsymbol{b}$ at positions $\ell - d_c$ to $\ell - 1$ and class c.

The parameters define proper probability distributions and, hence,

- $\forall c \in \mathcal{C} : \phi_c \in [0,1]$ and $\sum_{\tilde{c}} \phi_{\tilde{c}} = 1$,
- $\forall c \in \mathcal{C}, \forall a \in \Sigma : \phi_{1,a|c} \in [0,1]$ and $\sum_{\tilde{a}} \phi_{1,\tilde{a}|c} = 1$, and
- $\forall c \in \mathcal{C}, \forall \ell = 2, \ldots, L, \forall a \in \Sigma, \forall \boldsymbol{b} \in \Sigma^{\min\{d_c, \ell-1\}} : \phi_{\ell,a|\boldsymbol{b},c} \in [0,1]$ and $\sum_{\tilde{a}} \phi_{\ell,\tilde{a}|\boldsymbol{b},c} = 1$.

Let $\boldsymbol{\phi}_{1|c} = (\phi_{1,a_1|c}, \ldots, \phi_{1,a_{|\Sigma|}|c})$, and $\boldsymbol{\phi}_{\ell|\boldsymbol{b},c} = (\phi_{\ell,a_1|\boldsymbol{b},c}, \ldots, \phi_{\ell,a_{|\Sigma|}|\boldsymbol{b},c})$. We assume *parameter independence* (Heckerman et al., 1995), i.e. the parameter vectors $\boldsymbol{\phi}_{1|c}$, and $\boldsymbol{\phi}_{\ell|\boldsymbol{b},c}$ at all positions $\ell = 2, \ldots, L$ and for all contexts $\boldsymbol{b}$ are pair-wise independent.

We can re-write equation (3.27) as

$$P(\boldsymbol{x}, c \,|\, \boldsymbol{\phi}) \quad = \quad P(c|\phi_c) P_1(x_1|c, \boldsymbol{\phi}_{1|c}) \prod_{\ell=2}^{L} P_\ell(x_\ell | x_{\ell-d_c}, \ldots, x_{\ell-1}, c, \boldsymbol{\phi}_{\ell|x_{\ell-d_c},\ldots,x_{\ell-1},c})$$

and finally denote the likelihood directly using the parameters

$$P(\boldsymbol{x}, c \,|\, \boldsymbol{\phi}) \quad = \quad \phi_c \phi_{1,x_1|c} \prod_{\ell=2}^{L} \phi_{\ell,x_\ell|x_{\ell-d_c},\ldots,x_{\ell-1},c}. \tag{3.28}$$

If we insert this definition of the likelihood into the definition of the class posterior (equation (3.10), p. 12), we obtain

$$P(c \,|\, \boldsymbol{x}, \boldsymbol{\phi}) \quad = \quad \frac{\phi_c \phi_{1,x_1|c} \prod_{\ell=2}^{L} \phi_{\ell,x_\ell|x_{\ell-d_c},\ldots,x_{\ell-1},c}}{\sum_{\tilde{c}} \phi_{\tilde{c}} \phi_{1,x_1|\tilde{c}} \prod_{\ell=2}^{L} \phi_{\ell,x_\ell|x_{\ell-d_{\tilde{c}}},\ldots,x_{\ell-1},\tilde{c}}}. \tag{3.29}$$

As noted in section 3.2.1, the parameters in $\boldsymbol{\phi}$, which are constrained to the interval $[0,1]$, are not suited for numerical optimization. Hence, we seek a parameterization of the class posterior in terms of real-valued parameters. Wettig et al. (2003) propose such a parameterization by defining

$$P(c \,|\, \boldsymbol{x}, \boldsymbol{\xi}) \quad = \quad \frac{\exp\left(\xi_c + \xi_{1,x_1|c} + \sum_{\ell=2}^{L} \xi_{\ell,x_\ell|x_{\ell-d_c},\ldots,x_{\ell-1},c}\right)}{\sum_{\tilde{c}} \exp\left(\xi_{\tilde{c}} + \xi_{1,x_1|\tilde{c}} + \sum_{\ell=2}^{L} \xi_{\ell,x_\ell|x_{\ell-d_{\tilde{c}}},\ldots,x_{\ell-1},\tilde{c}}\right)}, \tag{3.30}$$

where $\xi_c \in \mathbb{R}$ is the parameter for class c, $\xi_{1,a|c} \in \mathbb{R}$ is the parameter for symbol a at position 1 given class c, and $\xi_{\ell,a|\boldsymbol{b},c} \in \mathbb{R}$ is the parameter for symbol a at position ℓ given class c and context $\boldsymbol{b}$.

We want to find a transformation (Wettig et al., 2003; Keilwagen et al., 2010b) from unconstrained parameters $\boldsymbol{\xi}$ to the constrained parameters $\boldsymbol{\phi}$ to show that, using this transformation,

the two definitions of the class posterior of equations (3.29) and (3.30) are equivalent. We define the transformation $\boldsymbol{t}$ as

$$\phi_c = t_c(\boldsymbol{\xi}) := \frac{\exp(\xi_c)\, Z_c(\boldsymbol{\xi})}{\sum_{\tilde{c}} \exp(\xi_{\tilde{c}})\, Z_{\tilde{c}}(\boldsymbol{\xi})} \tag{3.31}$$

$$\phi_{1,a|c} = t_{1,a|c}(\boldsymbol{\xi}) := \frac{\exp\left(\xi_{1,a|c}\right) Z_{1,a|c}(\boldsymbol{\xi})}{\sum_{\tilde{a}} \exp\left(\xi_{1,\tilde{a}|c}\right) Z_{1,\tilde{a}|c}(\boldsymbol{\xi})} \tag{3.32}$$

$$\phi_{\ell,a|\boldsymbol{b},c} = t_{\ell,a|\boldsymbol{b},c}(\boldsymbol{\xi}) := \frac{\exp\left(\xi_{\ell,a|\boldsymbol{b},c}\right) Z_{\ell,a|\boldsymbol{b},c}(\boldsymbol{\xi})}{\sum_{\tilde{a}} \exp\left(\xi_{\ell,\tilde{a}|\boldsymbol{b},c}\right) Z_{\ell,\tilde{a}|\boldsymbol{b},c}(\boldsymbol{\xi})}, \tag{3.33}$$

where the normalization terms $Z(\boldsymbol{\xi})$ are defined as

$$Z_{L,a|\boldsymbol{b},c}(\boldsymbol{\xi}) = 1, \tag{3.34}$$

$$Z_{\ell,a|b_1\ldots b_{d_c},c}(\boldsymbol{\xi}) = \sum_{\tilde{a}} \exp\left(\xi_{\ell+1,\tilde{a}|b_2\ldots b_{d_c}a,c}\right) Z_{\ell+1,\tilde{a}|b_2\ldots b_{d_c}a,c}(\boldsymbol{\xi}) \tag{3.35}$$

$$Z_{1,a|c}(\boldsymbol{\xi}) = \sum_{\tilde{a}} \exp\left(\xi_{2,\tilde{a}|a,c}\right) Z_{2,\tilde{a}|a,c}(\boldsymbol{\xi}) \tag{3.36}$$

$$Z_c(\boldsymbol{\xi}) = \sum_{\tilde{a}} \exp\left(\xi_{1,\tilde{a}|c}\right) Z_{1,\tilde{a}|c}(\boldsymbol{\xi}) \tag{3.37}$$

and the last parameters are fixed to zero, i.e. $\xi_K := 0$, $\forall c \in \mathcal{C} : \xi_{1,|\Sigma||c} := 0$, and $\xi_{\ell,|\Sigma||\boldsymbol{b},c} := 0$.

We can also define an inverse transformation from $\boldsymbol{\phi}$-parameters to $\boldsymbol{\xi}$-parameters as

$$\xi_c = t_c^{-1}(\boldsymbol{\phi}) := \log\left(\frac{\phi_c}{\phi_K}\right) \tag{3.38}$$

$$\xi_{1,a|c} = t_{1,a|c}^{-1}(\boldsymbol{\phi}) := \log\left(\frac{\phi_{1,a|c}}{\phi_{1,|\Sigma||c}}\right) \tag{3.39}$$

$$\phi_{\ell,a|\boldsymbol{b},c} = t_{\ell,a|\boldsymbol{b},c}^{-1}(\boldsymbol{\phi}) := \log\left(\frac{\phi_{\ell,a|\boldsymbol{b},c}}{\phi_{\ell,|\Sigma||\boldsymbol{b},c}}\right) \tag{3.40}$$

Are more detailed derivation of this parameterization is given in (Keilwagen et al., 2010b). In (Wettig et al., 2003; Keilwagen et al., 2010b), this transformation is extended to moral Bayesian networks, which are a generalization of Markov models with an arbitrary moral structure of the underlying DAG.

If we insert the transformation into equation (3.28), we find that many of the normalization terms cancel and we obtain

$$P(\boldsymbol{x}, c|\boldsymbol{\xi}) = \frac{1}{Z(\boldsymbol{\xi})} \exp\left(\xi_c + \xi_{1,x_1|c} + \sum_{\ell=2}^{L} \xi_{\ell,x_\ell|x_{\ell-d_c},\ldots,x_{\ell-1},c}\right), \tag{3.41}$$

where

$$Z(\boldsymbol{\xi}) = \sum_c \exp(\xi_c)\, Z_c(\boldsymbol{\xi}), \tag{3.42}$$

which, using equations (3.34) through (3.37), can be expanded to

$$Z(\boldsymbol{\xi}) = \sum_{c} \sum_{\boldsymbol{x} \in \Sigma^L} \exp\left(\xi_c + \xi_{1,x_1|c} + \sum_{\ell=2}^{L} \xi_{\ell,x_\ell|x_{\ell-d_c},\ldots,x_{\ell-1},c} \right). \tag{3.43}$$

Using the likelihood of equation (3.41) for the class posterior of equation (3.10) (p. 12), the normalization terms $Z(\boldsymbol{\beta})$ cancel as well, and we obtain the definition of the class posterior of equation (3.30) without explicit use of any of the normalization terms. However, we will need the normalization terms, when we define a transformed product-Dirichlet prior on the parameters $\boldsymbol{\xi}$ in section 3.4.2.

It can be proven that MCL is a concave optimization problem (Wettig et al., 2003) for Markov models parameterized in terms of $\boldsymbol{\xi}$ when stated in terms of the *log* conditional likelihood. Thus, the values of the parameters $\boldsymbol{\xi}^*_{\mathrm{MCL}}$ obtained by numerical optimization do not depend on the initialization and one run of the optimization is sufficient to reliably obtain the globally optimal parameters. Since the priors on the parameters of Markov models introduced in section 3.4 are log concave functions of the parameters as well, the same holds true when optimizing the parameters according to the discriminative MSP principle. A proof of concavity of conditional likelihood for inhomogeneous Markov models and of the transformed Dirichlet prior is given in appendix A.1.

When we consider two-class problems, i.e. $c \in \{1, 2\}$, we can slightly re-write equation (3.30) as

$$P\left(c \,|\, \boldsymbol{x}, \boldsymbol{\xi}\right) = \frac{1}{1 + \exp\left(\xi_2 - \xi_1 + \xi_{1,x_1|2} - \xi_{1,x_1|1} + \sum_{\ell=2}^{L} \left[\xi_{\ell,x_\ell|x_{\ell-d_2},\ldots,x_{\ell-1},2} - \xi_{\ell,x_\ell|x_{\ell-d_1},\ldots,x_{\ell-1},1} \right] \right)}. \tag{3.44}$$

We can interpret the differences of parameter values $\xi_{\ell,x_\ell|x_{\ell-d_2},\ldots,x_{\ell-1},2} - \xi_{\ell,x_\ell|x_{\ell-d_1},\ldots,x_{\ell-1},1}$ as alternative, real-valued parameters of the class posterior. The length of the context considered by these alternative parameters is then equal to the maximum of the two orders d_1 and d_2. This elucidates that the expressiveness of the class posterior is determined by the maximum order of d_1 and d_2 or – in other words – we can choose d_1 arbitrarily if $d_1 \leq d_2$ without affecting the expressiveness of the class posterior, and vice versa. We could even omit the parameters of the model having the lower order without losing expressiveness of the class posterior. Considering MSP instead of MCL, i.e. imposing a prior on the parameters $\boldsymbol{\xi}$, may abolish this property, depending on the employed prior and its hyper-parameters (Grau et al., 2007a).

Equation (3.44) also shows the close relation between MCL for Markov models and logistic regression (see equation (3.6)) if we consider two-class problems. If we encode the sequence $\boldsymbol{x}$ as a binary vector of length $L \cdot |\Sigma|$ and use the differences $\xi_1 - \xi_2$, $\xi_{1,a|1} - \xi_{1,a|2}$, and $\xi_{\ell,a|\boldsymbol{b},1} - \xi_{\ell,a|\boldsymbol{b},2}$ as entries of the parameter vector $\boldsymbol{\beta}$ of equation (3.6) (p. 11), we obtain the same functional form as in equation (3.44).

3.3.1.2. Homogeneous Markov models

In analogy to equation (3.28), we define the likelihood of a homogeneous Markov model of order d_c in terms of $\boldsymbol{\phi}$-parameters

$$P(\boldsymbol{x}, c \,|\, \boldsymbol{\phi}) \quad = \quad \phi_c \prod_{\ell=1}^{d_c} \phi_{\ell,x_\ell|x_{\ell-d_c},\ldots,x_{\ell-1},c} \prod_{\ell=d_c+1}^{L} \phi_{x_\ell|x_{\ell-d_c},\ldots,x_{\ell-1},c}, \tag{3.45}$$

where the parameters $\phi_{\ell,x_\ell|x_{\ell-d_c},\ldots,x_{\ell-1},c}$ are responsible for the position-dependent initial distribution of the first d_c symbols in $\boldsymbol{x}$, whereas the parameters $\phi_{x_\ell|x_{\ell-d_c},\ldots,x_{\ell-1},c}$ representing the conditional probabilities at the remaining positions $d_c + 1$ through L positions are position-independent, i.e. homogeneous.

We cannot use the parameter transformation defined for inhomogeneous models to obtain unconstrained parameters $\boldsymbol{\xi}$ in case of homogeneous Markov models, because the normalization terms $Z(\boldsymbol{\xi})$ depend on the parameters at subsequent positions. Hence, the homogeneity of parameters would lead to cyclic dependencies between the homogeneous parameters. Additionally, homogeneous Markov models can model sequences of arbitrary length and the interpretation of the $\boldsymbol{\xi}$-parameters in terms of probabilities would depend on the length of the considered sequences. Hence, we define independent local transformations (Meila-Predoviciu, 1999) for the parameters of homogeneous Markov models, which involve those parameters living on the same simplex. Let $\boldsymbol{\xi}_{\ell|\boldsymbol{g},c} = (\xi_{\ell,a_1|\boldsymbol{g},c}, \ldots, \xi_{\ell,a_{|\Sigma|}|\boldsymbol{g},c}),\ \boldsymbol{g} \in \Sigma^{\ell-1}$ denote the vectors of the parameters of the initial distribution. Let $\boldsymbol{\xi}_{\boldsymbol{b},c} = (\xi_{a_1|\boldsymbol{b},c}, \ldots, \xi_{a_{|\Sigma|}|\boldsymbol{b},c})$ denote the vectors of homogeneous parameters conditional on context $\boldsymbol{b} \in \Sigma^{d_c}$ and class c. We define the transformations $\boldsymbol{t}_{\text{hom}}$ from $\boldsymbol{\xi}$-parameters to $\boldsymbol{\phi}$-parameters as

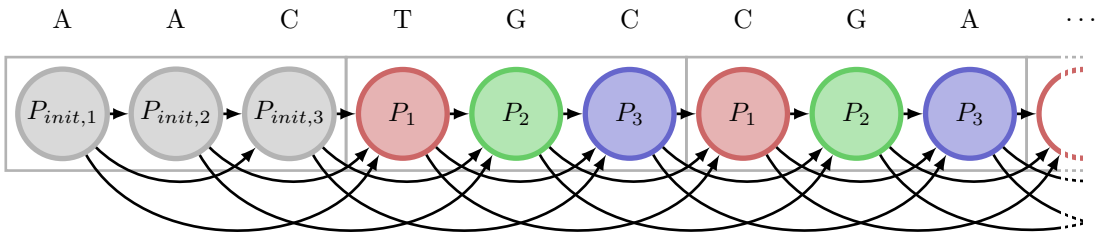
$$\phi_c = t_{\text{hom},c}(\boldsymbol{\xi}) \quad := \quad \frac{\exp(\xi_c)}{\sum_{\tilde{c}} \exp(\xi_{\tilde{c}})} \tag{3.46}$$

$$\phi_{\ell,a|\boldsymbol{g},c} = t_{\text{hom},\ell,a|\boldsymbol{g},c}(\boldsymbol{\xi}_{\ell|\boldsymbol{g},c}) \quad := \quad \frac{\exp\left(\xi_{\ell,a|\boldsymbol{g},c}\right)}{\sum_{\tilde{a}} \exp\left(\xi_{\ell,\tilde{a}|\boldsymbol{g},c}\right)} \tag{3.47}$$

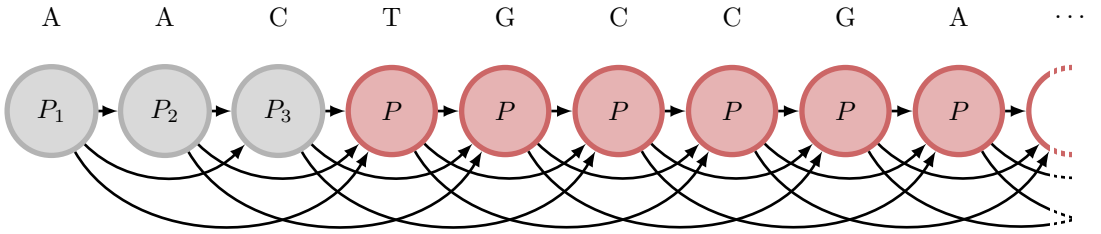
$$\phi_{a|\boldsymbol{b},c} = t_{\text{hom},a|\boldsymbol{b},c}(\boldsymbol{\xi}_{\boldsymbol{b},c}) \quad := \quad \frac{\exp\left(\xi_{a|\boldsymbol{b},c}\right)}{\sum_{\tilde{a}} \exp\left(\xi_{\tilde{a}|\boldsymbol{b},c}\right)} \tag{3.48}$$

The inverse transformation of equation (3.38) remains valid for homogeneous Markov models (Meila-Predoviciu, 1999).

Although this transformation results in unconstrained parameters $\boldsymbol{\xi}$, which can be optimized numerically, we do not obtain a log concave conditional likelihood for homogeneous Markov models of orders greater than zero. Hence, we must restart numerical optimization from different initializations to obtain globally optimal parameters $\boldsymbol{\xi}^*_{\text{MSP}}$ with high probability. However, we may use the parameter values obtained by the generative MAP principle as initialization, which assures that the parameters obtained by numerical optimization yield a supervised posterior that is not lower than for MAP parameters. This strategy is also referred to as using *plug-in* parameters.



(a) Periodic Markov model of order 3 with period 3 for phase 1.



(b) Homogeneous Markov model of order 3

Figure 3.3.: Graphical representation of a periodic and a homogeneous Markov model. The homogeneous Markov models employs the same conditional probability distribution for positions 4 and above, whereas the periodic Markov model re-uses probability distributions with a period of 3 as indicated by the coloring of nodes.

3.3.1.3. Periodic Markov models

We use another class of Markov models, namely periodic Markov models, for discriminating coding and non-coding sequences in section 4.3. On the one hand, periodic Markov models are similar to homogeneous Markov models in re-using identical probability distributions for different positions. On the other hand, these probability distributions are re-used only with a given period and, similar to inhomogeneous Markov models, are not position independent. Since we use the periodic Markov model for its capability to represent codons, we consider only periodic Markov models with a period and order of 3 in the following. This also simplifies formalization, although the extension to other periods and orders is straightforward.

The 3-periodic Markov model explicitly models codons, where the probability of a certain nucleotide x_ℓ at position ℓ depends on its localization within the codon. In a periodic Markov model of order 3 this probability also depends on the three preceding nucleotides, i.e. the probability of the first nucleotide in a codon depends on all three nucleotides of the preceding codon, the probability of the second nucleotide depends on the first nucleotide in the current codon and the last two nucleotides of the preceding codon, and the probability of the third nucleotide depends on the previous two nucleotides in the current codon and the last nucleotide of the preceding codon.

Figure 3.3 shows a graphical representation of a 3-periodic Markov model of order 3 in comparison to a homogeneous Markov model of order 3. While we assume the same conditional probability distribution for positions 4 and above in case of the homogeneous Markov model, the periodic Markov model re-uses conditional probability distributions with a period of 3, i.e. the conditional probability distribution at position ℓ is the same as at position $\ell + 3$, but

potentially different from those at positions $\ell + 1$ and $\ell + 2$. Figure 3.3 visualizes the case of phase 1 for which the reading frame starts at position 1, whereas the complete model is a mixture model over the three possible frames on one strand. The differentiation between forward and complementary strand could be accomplished by a surrounding mixture model over the two strands.

In this case, we skip the definition in terms of constrained parameters $\boldsymbol{\phi}$ and first define the 3-periodic Markov model in terms of probabilities which are then expressed in terms of real-valued parameters $\boldsymbol{\xi}$. We define

$$P_{\text{pMM}}(\boldsymbol{x}|c, \boldsymbol{\xi}) = \sum_{f=1}^{3} \Big[P(f|c, \boldsymbol{\xi}) P_{\text{init},f}(x_1, x_2, x_3|c, \boldsymbol{\xi}) \cdot \prod_{\ell=3}^{L} P_{(f+\ell) \bmod 3}(x_\ell | x_{\ell-1}, \ldots, x_{\ell-3}, c, \boldsymbol{\xi}) \Big], \tag{3.49}$$

where the mixture probabilities $P(f|c, \boldsymbol{\xi})$ of the three phases $f \in \{1, 2, 3\}$ are parameterized as

$$P(f|c, \boldsymbol{\xi}) = \frac{\exp(\xi_{f|c})}{\sum_{\tilde{f}=0}^{2} \exp(\xi_{\tilde{f}|c})}, \tag{3.50}$$

the initial probability distribution of phase f is defined as

$$P_{\text{init},f}(x_1, x_2, x_3|c, \boldsymbol{\xi}) = \frac{\exp(\xi_{\text{pMM},f,x_1|c})}{\sum_{a\in\Sigma} \exp(\xi_{\text{pMM},f,a|c})} \frac{\exp(\xi_{\text{pMM},f,x_2|x_1,c})}{\sum_{a\in\Sigma} \exp(\xi_{\text{pMM},f,a|x_1,c})} \frac{\exp(\xi_{\text{pMM},f,x_3|x_1,x_2,c})}{\sum_{a\in\Sigma} \exp(\xi_{\text{pMM},f,a|x_1,x_2,c})}, \tag{3.51}$$

and the periodic conditional probabilities of the i-th nucleotide, $i \in \{1, 2, 3\}$, of a codon are defined as

$$P_i(x_\ell|x_{\ell-1}, x_{\ell-2}, x_{\ell-3}, c, \boldsymbol{\xi}) = \frac{\exp(\xi_{\text{pMM},i,x_\ell|x_{\ell-1},x_{\ell-2},x_{\ell-3},c})}{\sum_{a\in\Sigma} \exp(\xi_{\text{pMM},i,a|x_{\ell-1},x_{\ell-2},x_{\ell-3},c})}. \tag{3.52}$$

As for homogeneous Markov models of higher order, we do not obtain a concave conditional likelihood for periodic Markov models. Hence, we use plug-in parameters using uniform $P(f|c, \boldsymbol{\xi})$ for periodic Markov models as well.

3.3.2. Continuous random variables

Although we consider (discrete) nucleotide sequences in this work, we need to handle continuous random variables and corresponding densities in some cases as well. Examples for continuous values derived from sequence are physico-chemical properties like melting temperature or free energy, geometrical properties like twist or shift, and probabilistic measures like the entropy of k-mer compositions. These measures are presented in detail in section 4.3.2.6 (p. 99). In this work, we consider Gaussian and gamma densities for modelling continuous

random variables. We define the likelihood $P(\boldsymbol{y}, c \,|\, \boldsymbol{\beta})$ of a continuous sequence $\boldsymbol{y}$ and class c given parameters $\boldsymbol{\beta}$ as

$$p(\boldsymbol{y}, c \,|\, \boldsymbol{\beta}) \quad = \quad P(c|\boldsymbol{\beta}) p(\boldsymbol{y}|c, \boldsymbol{\beta}), \tag{3.53}$$

where $P(c|\boldsymbol{\beta})$ denotes the a-priori probability of class c given parameters $\boldsymbol{\beta}$, and $p(\boldsymbol{y}|c, \boldsymbol{\beta})$ denotes the likelihood of $\boldsymbol{y}$ given class c and parameters $\boldsymbol{\beta}$. In contrast to Markov models (cf. equation (3.31), we parameterize the a-priori class probabilities $P(c|\boldsymbol{\beta})$ and the densities $p(\boldsymbol{y}, c \,|\, \boldsymbol{\beta})$ independently. We define $P(c|\boldsymbol{\beta})$ as

$$P(c|\boldsymbol{\beta}) \quad = \quad \frac{\exp(\boldsymbol{\beta}_c)}{\sum_{\tilde{c}} \exp(\boldsymbol{\beta}_{\tilde{c}})}. \tag{3.54}$$

Since the parameters of the densities responsible for different classes are parameterized independently as well, we waive denoting the dependency of $p(\boldsymbol{y}|c, \boldsymbol{\beta})$ on the class explicitly in the following sections.

3.3.2.1. (Bivariate) Gaussian density

Let Y be a Gaussian distributed random variable with values $y \in \mathbb{R}$. The Gaussian density with mean $\mu \in \mathbb{R}$ and precision $\lambda \in \mathbb{R}^+$ of values y is defined as

$$\mathcal{N}(y|\mu, \lambda) \quad = \quad \sqrt{\frac{\lambda}{2\pi}} \exp\left(-\frac{1}{2}\lambda(y-\mu)^2\right). \tag{3.55}$$

More commonly, the Gaussian density is parameterized by a variance parameter $\sigma^2 = \frac{1}{\lambda}$. However, the parameterization by the precision λ is more convenient for numerical optimization, the definition of a conjugate prior, and the multivariate Gaussian distribution.

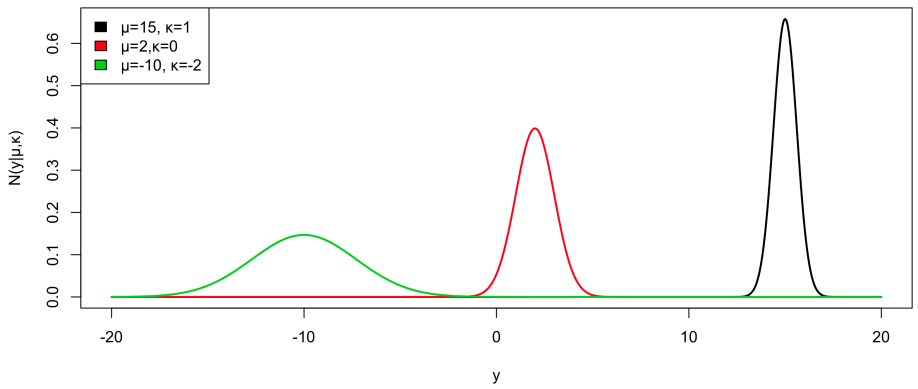


Figure 3.4.: Gaussian density for different values of μ and κ. The location of the maximum of the density is controlled by the parameter μ. The parameter κ influences the variance of the density, where smaller values κ lead to greater variances.

Since the precision λ is limited to positive values, we define a transformation $\lambda = \exp(\kappa), \kappa \in \mathbb{R}$ for unconstrained numerical optimization. The resulting Gaussian density in terms of $\mu \in \mathbb{R}$

and $\kappa \in \mathbb{R}$ is defined as

$$\mathcal{N}(y|\mu,\kappa) = \sqrt{\frac{\exp(\kappa)}{2\pi}} \exp\left(-\frac{1}{2}\exp(\kappa)(y-\mu)^2\right). \tag{3.56}$$

The Gaussian density for different values of μ and κ is depicted in figure 3.4.

For the position distribution employed for de-novo discovery of cis-regulatory modules (see section 4.2), we also need the multivariate, in this case bivariate, generalization of the Gaussian density. Let $\boldsymbol{Y} = (Y_1, \ldots, Y_D)$ be a vector of random variables assuming values $\boldsymbol{y} \in \mathbb{R}^D$. The multivariate Gaussian density with mean vector $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{\Lambda}$ of values $\boldsymbol{y}$ is defined as

$$\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu},\boldsymbol{\Lambda}) = \frac{\det(\boldsymbol{\Lambda})^{1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\boldsymbol{\Lambda}(\boldsymbol{y}-\boldsymbol{\mu})\right) \tag{3.57}$$

In case of the bivariate Gaussian density, we may express the precision matrix $\boldsymbol{\Lambda}$ in terms of the precisions λ_i of random variables Y_i and the correlation $\rho_{1,2}$ between random variables Y_1 and Y_2

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & -\rho_{1,2}\sqrt{\lambda_1\lambda_2} \\ -\rho_{1,2}\sqrt{\lambda_1\lambda_2} & \lambda_2 \end{pmatrix}. \tag{3.58}$$

The dependencies $\lambda_{i,j} = -\rho_{i,j}\sqrt{\lambda_i\lambda_j}$ do not hold for general precision matrices of higher-variate Gaussian densities. Since, we only need bivariate Gaussian densities in the following, we limit the further derivations to the bivariate case.

Again, we seek a parameterization of the bivariate Gaussian density that is defined on unconstrained parameters. To this end, we use the same transformation $\lambda_i = \exp(\kappa_i), \kappa_i \in \mathbb{R}$ as in the univariate case for the precisions. Additionally, we define the co-precision as $\lambda_{1,2} = -\tanh(r_{1,2})\sqrt{\exp(\kappa_1)\exp(\kappa_2)}, r_{1,2} \in \mathbb{R}$, where the hyperbolic tangent maps the real-valued parameter $r_{1,2}$ to the interval $(-1, 1)$ as required for correlation. This leads to a definition of the precision matrix in terms of the vector of precisions $\boldsymbol{\kappa} = (\kappa_1, \kappa_2)$ and the correlation parameter $r_{1,2}$

$$\boldsymbol{\Lambda}(\boldsymbol{\kappa}, r_{1,2}) = \begin{pmatrix} \exp(\kappa_1) & -\tanh(r_{1,2})\sqrt{\exp(\kappa_1)\exp(\kappa_2)} \\ -\tanh(r_{1,2})\sqrt{\exp(\kappa_1)\exp(\kappa_2)} & \exp(\kappa_2) \end{pmatrix}, \tag{3.59}$$

which can then be used to define the bivariate Gaussian density in terms of $\boldsymbol{\mu}$, $\boldsymbol{\kappa}$, and $r_{1,2}$:

$$\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu},\boldsymbol{\kappa},r_{1,2}) = \frac{\det(\boldsymbol{\Lambda}(\boldsymbol{\kappa},r_{1,2}))^{1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\boldsymbol{\Lambda}(\boldsymbol{\kappa},r_{1,2}))(\boldsymbol{y}-\boldsymbol{\mu})\right). \tag{3.60}$$

We use the bivariate Gaussian density to model the distribution of the occurrence of multiple motifs in a sequence of length L. In this case, we decide to restrict the means μ_i to the interval

$[1, L]$. We achieve this by an additional transformation $\mu_i(\nu_i) = L\frac{\exp(\nu_i)}{1+\exp(\nu_i)}, \nu_i \in \mathbb{R}$, and yield

$$\mathcal{N}(\boldsymbol{y}|\boldsymbol{\nu}, \boldsymbol{\kappa}, r_{1,2}) = \frac{\det(\boldsymbol{\Lambda}(\boldsymbol{\kappa}, r_{1,2}))^{1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{\nu}))^T \boldsymbol{\Lambda}(\boldsymbol{\kappa}, r_{1,2}))(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{\nu}))\right), \quad (3.61)$$

where $\boldsymbol{\nu} = (\nu_1, \nu_2)$ and $\boldsymbol{\mu}(\boldsymbol{\nu}) = (\mu_1(\nu_1), \mu_2(\nu_2))$.

The bivariate Gaussian density is defined for continuous values of $\boldsymbol{y}$, whereas we consider only discrete positions ℓ for the position distribution in section 4.2. We deal with this problem by normalizing the Gaussian density by the sum over all admissible positions in section 4.2.

3.3.2.2. Gamma density

Let Y be a gamma distributed random variable with values $y \in \mathbb{R}^+$. The gamma density with shape $a \in \mathbb{R}^+$ and rate $b \in \mathbb{R}^+$ of values y is defined as

$$\mathcal{G}(y|a, b) = \frac{b^a}{\Gamma(a)} y^{a-1} e^{(-by)} \quad (3.62)$$

Alternatively, the gamma density can be parameterized by a scale parameter $s = \frac{1}{b}$ and the decision for either of the two parameterizations is somewhat arbitrary.

Again, we need to define transformations for the two limited parameters a and b. We choose $a = \exp(\gamma)$ and $b = \exp(\beta)$, yielding the density

$$\mathcal{G}(y|\gamma, \beta) = \frac{e^{\beta \exp(\gamma)}}{\Gamma(e^\gamma)} e^{(\exp(\gamma)-1)\log(y) - \exp(\beta)y} \quad (3.63)$$

The gamma density is depicted in figure 3.5 for different values of γ and β.

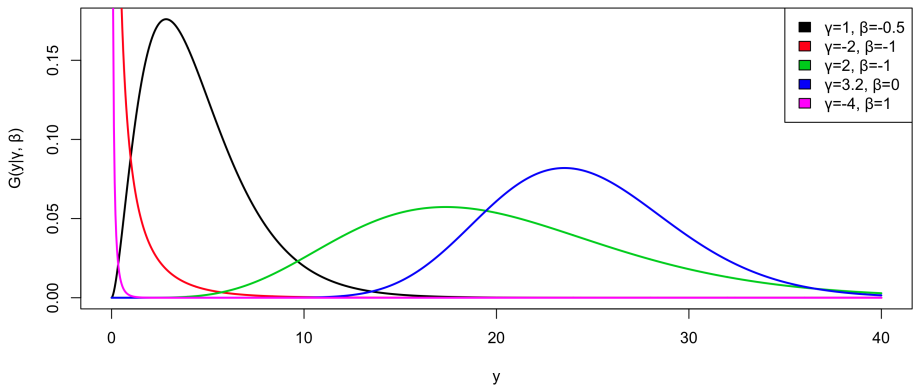


Figure 3.5.: Gamma density for different values of γ and β.

3.4. Priors

In section 3.2.2, we defined the supervised posterior as the product of conditional likelihood $P(\boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\beta})$ and a prior $q(\boldsymbol{\beta} \mid \boldsymbol{\alpha})$ on the parameters $\boldsymbol{\beta}$. In this section, we present priors for the parameters of Markov models, as well as for the parameters of Gaussian and gamma densities. For Markov models, we define Gaussian, Laplace, and Dirichlet priors, and we discuss advantages and drawbacks of choosing any of these priors. For the Gaussian density, we define the conjugate normal-gamma prior, whereas for the gamma density, we derive a conjugate density via the general definition of a conjugate prior for the exponential family.

3.4.1. Markov models: Gaussian and Laplace priors

Gaussian and Laplace priors are a common choice for the parameters of Markov random fields (Chen and Rosenfeld, 1999) and logistic regression (Madigan et al., 2005; Genkin et al., 2005; Cawley et al., 2007). The natural parameterization of Markov random fields is closely related to the parameterization of Markov models in terms of $\boldsymbol{\xi}$-parameters chosen here. Furthermore, Markov models are included in the class of Markov random fields as special cases. Hence, it may be worthwhile to investigate Gaussian and Laplace priors for Markov models as well.

We define the Gaussian prior for a parameter ξ_i, i.e, $\xi_c, \xi_{1,a|c}$, or $\xi_{\ell,a|\boldsymbol{b},c}$, as

$$p(\xi_i|\mu_i, \sigma_i^2) \quad = \quad \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{1}{2}\frac{(\xi_i - \mu_i)^2}{\sigma_i^2}\right), \tag{3.64}$$

where μ_i denotes the prior mean and σ_i^2 denotes the prior variance of parameter ξ_i. In contrast to the Gaussian density presented in section 3.3.2.1, we choose the more common parameterization in terms of the variance, because mean and variance are hyper-parameters in this case and therefore do not need to be optimized. We assume that all parameters in $\boldsymbol{\xi}$ are independent, yielding the complete prior

$$\begin{aligned} p(\boldsymbol{\xi}|\boldsymbol{\mu}, \boldsymbol{\sigma^2}) = & \left[\prod_{c=1}^{K-1} p(\xi_c|\mu_c, \sigma_c^2)\right] \cdot \\ & \prod_{c=1}^{K} \prod_{a=1}^{|\Sigma|-1} p(\xi_{1,a|c}|\mu_{1,a|c}, \sigma_{1,a|c}^2) \prod_{\ell=2}^{L} \prod_{\boldsymbol{b}\in\Sigma^{\min\{\ell-1,d_c\}}} p(\xi_{\ell,a|\boldsymbol{b},c}|\mu_{\ell,a|\boldsymbol{b},c}, \sigma_{\ell,a|\boldsymbol{b},c}^2), \end{aligned} \tag{3.65}$$

where $\boldsymbol{\mu}$ denotes the vector of all prior means

$$\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{K-1}, \mu_{1,a_1|1}, \ldots, \mu_{1,a_{|\Sigma|}-1|K}, \mu_{2,a_1|a_1,1}, \ldots, \mu_{L,a_{|\Sigma|}-1|a_{|\Sigma|}\ldots a_{|\Sigma|},K}) \tag{3.66}$$

and $\boldsymbol{\sigma^2}$ denotes the vector of all prior variances

$$\boldsymbol{\sigma^2} = (\sigma_1^2, \ldots, \sigma_{K-1}^2, \sigma_{1,a_1|1}^2, \ldots, \sigma_{1,a_{|\Sigma|}-1|K}^2, \sigma_{2,a_1|a_1,1}^2, \ldots, \sigma_{L,a_{|\Sigma|}-1|a_{|\Sigma|}\ldots a_{|\Sigma|},K}^2). \tag{3.67}$$

The assumption of the independence of all parameters is clearly violated for those parameters living on a common simplex, which is a conceptual drawback of the Gaussian as well as the

Laplace prior.

The prior means μ_c and prior variances σ_c^2 for the classes c are chosen problem specific and reflect the a-priori probabilities of the classes. In the following, we a-priorily assume that all nucleotides occur with the same probability. This corresponds to assuming that all parameters $\xi_{1,a|c}$ and $\xi_{\ell,a|\boldsymbol{b},c}$ are equal to zero. Therefore, we choose $\forall c \in \mathcal{C}, \forall a \in \Sigma : \mu_{1,a|c} = 0$ and $\forall c \in \mathcal{C}, \forall \ell \in [1, L], \forall a \in \Sigma, \forall \boldsymbol{b} \in \Sigma^{\min(\ell-1,d_c)} : \mu_{\ell,a|\boldsymbol{b},c} = 0$.

We define the variances as $\sigma_{1,a|c}^2 = \kappa_c |\Sigma|$ and $\sigma_{\ell,a|\boldsymbol{b},c}^2 = \kappa_c |\Sigma|^{\min(\ell-1,d_c)}$, where κ_c is a class-specific constant. The motivation for this choice is that we assume that the allowed variability of the parameters $\xi_{\ell,a|\boldsymbol{b},c}$ should increase with increasing order. This accounts for the increasing relative influence of the prior compared to the decreasing number of samples with given context $\boldsymbol{b}$, which may be balanced by a higher variance (Grau et al., 2007b).

Additionally, this choice of variances has the following property: Assume that we model position ℓ either with order d or with order $d+1$. The a-priori means at this position are zero in both cases. In the first case the variance is $\sigma_{\ell,a|\boldsymbol{b},c}^2 = \kappa_c |\Sigma|^d$, whereas in the second case it is $\sigma_{\ell,a|\boldsymbol{b}b_{d+1},c}^2 = \kappa_c |\Sigma|^{d+1}$. For each parameter $\xi_{\ell,a|\boldsymbol{b},c}$ in the first model we have $|\Sigma|$ different parameters $\xi_{\ell,a|\boldsymbol{b}b_{d+1},c}$ with augmented context $\boldsymbol{b}b_{d+1}, b_{d+1} \in \Sigma$ in the second model. Further assume that the additional context b_{d+1} is irrelevant and, hence, $\forall b_{d+1} \in \Sigma : \xi_{\ell,a|\boldsymbol{b}b_{d+1},c} = \xi_{\ell,a|\boldsymbol{b},c}$. Under these assumptions, we obtain the following Gaussian prior for the order $d+1$:

$$\prod_{b_{d+1} \in \Sigma} \frac{1}{\sqrt{2\pi\kappa_c |\Sigma|^{d+1}}} \exp\left(-\frac{1}{2}\frac{\left(\xi_{\ell,a|\boldsymbol{b},c} - 0\right)^2}{\kappa_c |\Sigma|^{d+1}}\right) \tag{3.68}$$

$$= \left(\frac{1}{\sqrt{2\pi\kappa_c |\Sigma|^{d+1}}}\right)^{|\Sigma|} \exp\left(-\frac{1}{2}\frac{\left(\xi_{\ell,a|\boldsymbol{b},c}\right)^2}{\kappa_c |\Sigma|^{d}}\right), \tag{3.69}$$

which is proportional to the prior for order d

$$\propto \frac{1}{\sqrt{2\pi\kappa_c |\Sigma|^{d}}} \exp\left(-\frac{1}{2}\frac{\left(\xi_{\ell,a|\boldsymbol{b},c}\right)^2}{\kappa_c |\Sigma|^{d}}\right) \tag{3.70}$$

Hence, we may state that, with this choice of the variances and means, the total influence of the prior on the parameters of a model of order $d+1$ is equal to the influence on the parameters of a model of order d.

In analogy to the Gaussian prior, we define the Laplace prior with mean μ_i and scale s_i for parameter ξ_i as

$$p(\xi_i|\mu_i, s_i) \quad = \quad \frac{1}{2s_i} \exp\left(-\frac{|\xi_i - \mu_i|}{s_i}\right), \tag{3.71}$$

and the complete prior as

$$p(\boldsymbol{\xi}|\mu,\boldsymbol{s}) = \left[\prod_{c=1}^{K-1} p(\xi_c|\mu_c,s_c)\right] \cdot \prod_{c=1}^{K}\prod_{a=1}^{|\Sigma|-1} p(\xi_{1,a|c}|\mu_{1,a|c},s_{1,a|c}) \prod_{\ell=2}^{L}\prod_{\boldsymbol{b}\in\Sigma^{\min\{\ell-1,d\}}} p(\xi_{\ell,a|\boldsymbol{b},c}|\mu_{\ell,a|\boldsymbol{b},c},s_{\ell,a|\boldsymbol{b},c}). \tag{3.72}$$

We choose the a-priori means in the same manner as for the Gaussian prior, and we choose the scale parameters such that the variance of the Laplace density is equal to the corresponding variance of the Gaussian density. This leads to the definition of the scale parameters as $s_i = \sqrt{\sigma_i^2/2}$. The motivation for this choice is to make the Gaussian and Laplace prior comparable to some extent.

The two drawbacks that are common to both the Gaussian and the Laplace prior are that both make incorrect assumptions about the independence of parameters and that their hyper-parameters cannot be interpreted intuitively in terms of a-priori assumptions about the data, which is a consequence of both priors not being conjugate to the likelihood of Markov models. In the next section we introduce the Dirichlet prior, which does not exhibit these disadvantages.

3.4.2. Markov models: Dirichlet prior

Commonly, the Dirichlet prior on the parameters of Markov models is defined in the $\boldsymbol{\phi}$-parameterization. As the parameters in different classes, at different positions, and for different contexts are assumed to be independent (Heckerman et al., 1995), the product-Dirichlet prior is the product of independent Dirichlet densities, each defined on a subset of parameters which live on a common simplex:

$$p\left(\boldsymbol{\phi}\,|\,\boldsymbol{\alpha}\right) \;=\; \Gamma(\alpha_\cdot)\prod_{k=1}^{K}\frac{\phi_k^{\alpha_k-1}}{\Gamma(\alpha_k)}\prod_{\ell=1}^{L}\prod_{\boldsymbol{b}\in\Sigma^{\min(d_k,\ell-1)}}\Gamma(\alpha_{\ell,\cdot|\boldsymbol{b},k})\prod_{a\in\Sigma}\frac{\phi_{\ell,a|\boldsymbol{b},k}^{\alpha_{\ell,a|\boldsymbol{b},k}-1}}{\Gamma(\alpha_{\ell,a|\boldsymbol{b},k})}, \tag{3.73}$$

where $\alpha_\cdot = \sum_k \alpha_k$ and $\alpha_{\ell,\cdot|\boldsymbol{b},k} = \sum_a \alpha_{\ell,a|\boldsymbol{b},k}$, and $\boldsymbol{b}$ is the empty string for $\ell = 1$.

We transform this definition of the Dirichlet prior to the space of $\boldsymbol{\xi}$-parameters using the transformation defined in equations (3.31) through (3.33) (p. 20). A detailed derivation of the transformed prior is given in (Keilwagen et al., 2010b). The Jacobian of the transformation amounts to (Keilwagen et al., 2010b)

$$\det\left(\frac{\partial \boldsymbol{t}(\boldsymbol{\xi})}{\partial\boldsymbol{\xi}}\right) \;=\; \prod_{k=1}^{K}\frac{\exp\left(\xi_k\right)Z_k(\boldsymbol{\xi})}{\sum_{\tilde{c}}\exp\left(\xi_{\tilde{c}}\right)Z_{\tilde{c}}(\boldsymbol{\xi})}\prod_{\ell=1}^{L}\prod_{\boldsymbol{b}\in\Sigma^{\min(d_k,\ell-1)}}\prod_{a\in\Sigma}\frac{\exp\left(\xi_{\ell,a|\boldsymbol{b},k}\right)Z_{\ell,a|\boldsymbol{b},k}(\boldsymbol{\xi})}{\sum_{\tilde{a}}\exp\left(\xi_{\ell,\tilde{a}|\boldsymbol{b},k}\right)Z_{\ell,\tilde{a}|\boldsymbol{b},k}(\boldsymbol{\xi})}.$$

Consequently, the transformed Dirichlet prior is defined as

$$q(\boldsymbol{\xi} \mid \boldsymbol{\alpha}) = \Gamma(\alpha_\cdot) \prod_{k=1}^{K} \frac{1}{\Gamma(\alpha_k)} \left(\frac{\exp(\xi_k) Z_k(\boldsymbol{\xi})}{\sum_{\tilde{c}} \exp(\xi_{\tilde{c}}) Z_{\tilde{c}}(\boldsymbol{\xi})} \right)^{\alpha_k} \tag{3.74}$$
$$\prod_{\ell=1}^{L} \prod_{\boldsymbol{b} \in \Sigma^{\min(d_k, \ell-1)}} \Gamma(\alpha_{\ell,\cdot|\boldsymbol{b},k}) \prod_{a \in \Sigma} \frac{1}{\Gamma(\alpha_{\ell,a|\boldsymbol{b},k})} \left(\frac{\exp\left(\xi_{\ell,a|\boldsymbol{b},k}\right) Z_{\ell,a|\boldsymbol{b},k}(\boldsymbol{\xi})}{\sum_{\tilde{a}} \exp\left(\xi_{\ell,\tilde{a}|\boldsymbol{b},k}\right) Z_{\ell,\tilde{a}|\boldsymbol{b},k}(\boldsymbol{\xi})} \right)^{\alpha_{\ell,a|\boldsymbol{b},k}}$$

We use BDeu (Bayesian Dirichlet likelihood-equivalent, uniform) hyper-parameters (Heckerman et al., 1995; Buntine, 1991) in the following. Such hyper-parameters have the properties that they can be interpreted as pseudo-counts stemming from a common set of pseudo-data, and assume that all possible sequences of length L occur with equal probability in the pseudo-data. These two properties allow for the application of the Dirichlet prior to the parameters of Markov models of differing orders while modelling the same a-priori information.

We define the hyper-parameters based on a set of *joint* hyper-parameters $\alpha_{\boldsymbol{x}|c}$ for each sequence $\boldsymbol{x} \in \Sigma^L$:

$$\alpha_{\ell,a|\boldsymbol{b},c} := \sum_{\boldsymbol{x} \in \Sigma^L} \alpha_{\boldsymbol{x}|c} \cdot \delta_{x_\ell,a} \cdot \delta_{x_{\ell-d_c} \ldots x_{\ell-1},\boldsymbol{b}} \tag{3.75}$$

$$\alpha_c := \sum_{\boldsymbol{x} \in \Sigma^L} \alpha_{\boldsymbol{x}|c} \tag{3.76}$$

The hyper-parameter α_c is often referred to as *equivalent sample size* (Heckerman et al., 1995; Buntine, 1991; Grau et al., 2007b), as it is equal to the size of the a-priorily observed set of pseudo-data in class c. Without further assumptions about the joint hyper-parameters, this choice of hyper-parameters corresponds to the BDe (Bayesian Dirichlet likelihood-equivalent) prior (Heckerman et al., 1995).

Under the assumption of uniform pseudo-data, the joint hyper-parameters $\alpha_{\boldsymbol{x}|c}$ are equal for each sequence $\boldsymbol{x} \in \Sigma^L$. This results in a definition of the hyper-parameters $\alpha_{\ell,a|\boldsymbol{b},c}$ based on the equivalent sample size α_c for class c

$$\alpha_{\ell,a|\boldsymbol{b},c} := \frac{\alpha_c}{|\Sigma|^{|\boldsymbol{b}|+1}}. \tag{3.77}$$

In case of BDeu hyper-parameters, many of the normalization terms of equation (3.74) cancel. Since $Z(\boldsymbol{\xi}) = \sum_{\tilde{c}} \exp(\xi_{\tilde{c}}) Z_{\tilde{c}}(\boldsymbol{\xi})$, we obtain a simplified version of the transformed product-Dirichlet prior for Markov models

$$q(\boldsymbol{\xi} \mid \boldsymbol{\alpha}) = \frac{1}{Z(\boldsymbol{\xi})^{\alpha_\cdot}} \Gamma(\alpha_\cdot) \prod_{k=1}^{K} \frac{e^{\alpha_k \xi_k}}{\Gamma(\alpha_k)} \prod_{\ell=1}^{L} \prod_{\boldsymbol{b} \in \Sigma^{\min(d_k, \ell-1)}} \Gamma(\alpha_{\ell,\cdot|\boldsymbol{b},k}) \prod_{a \in \Sigma} \frac{e^{\alpha_{\ell,a|\boldsymbol{b},k} \xi_{\ell,a|\boldsymbol{b},k}}}{\Gamma(\alpha_{\ell,a|\boldsymbol{b},k})} \tag{3.78}$$

$$\propto \frac{1}{Z(\boldsymbol{\xi})^{\alpha_\cdot}} \exp\left(\sum_{k=1}^{K} \alpha_k \xi_k + \sum_{\ell=1}^{L} \sum_{\boldsymbol{b} \in \Sigma^{\min(d_k, \ell-1)}} \sum_{a \in \Sigma} \alpha_{\ell,a|\boldsymbol{b},k} \xi_{\ell,a|\boldsymbol{b},k} \right). \tag{3.79}$$

A more general definition of the transformed product-Dirichlet prior, which applies to moral Bayesian networks as well as Markov random fields is given in (Keilwagen et al., 2010b). A

proof of the concavity of the transformed product-Dirichlet of equation (3.79), which is relevant for numerical optimization, is given in appendix A.1.

The product-Dirichlet prior for homogeneous Markov models in $\boldsymbol{\phi}$-parameterization is defined as

$$\begin{aligned} p(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) &= \Gamma(\alpha_\cdot) \prod_{k=1}^{K} \frac{\phi_k^{\alpha_k - 1}}{\Gamma(\alpha_k)} \left[\prod_{\ell=1}^{d_c} \prod_{\boldsymbol{g} \in \Sigma^{\ell-1}} \Gamma(\alpha_{\ell,\cdot|\boldsymbol{g},k}) \prod_{a \in \Sigma} \frac{\phi_{\ell,a|\boldsymbol{g},k}^{\alpha_{\ell,a|\boldsymbol{g},k}-1}}{\Gamma(\alpha_{\ell,a|\boldsymbol{g},k})} \right] \cdot \\ &\quad \left[\prod_{\boldsymbol{b} \in \Sigma^{d_c}} \Gamma(\alpha_{\cdot|\boldsymbol{b},k} \cdot (L_E - d_c)) \prod_{a \in \Sigma} \frac{\phi_{a|\boldsymbol{b},k}^{\alpha_{a|\boldsymbol{b},k} \cdot (L_E - d_c) - 1}}{\Gamma(\alpha_{a|\boldsymbol{b},k} \cdot (L_E - d_c))} \right], \end{aligned} \tag{3.80}$$

where L_E denotes the expected length of sequences to be scored. The term $(L_E - d_c)$ accounts for the employment of the same homogeneous parameters for all positions $\ell > d_c$.

In case of homogeneous Markov models, we transform the product-Dirichlet prior according to the transformation $\boldsymbol{t}_{\text{hom}}$. The Jacobian can be computed independently for each transformation of those parameters living on a common simplex in this case, resulting in

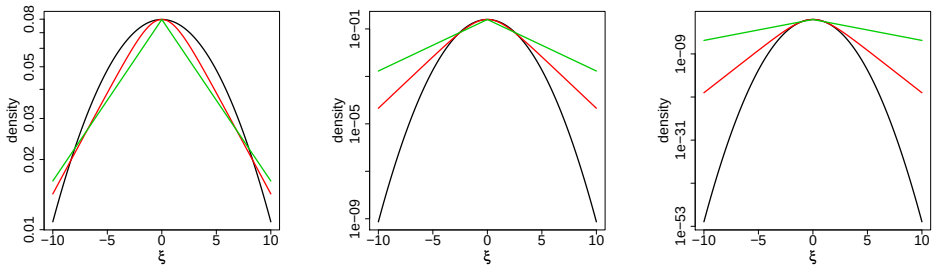
$$\begin{aligned} \det\left(\frac{\partial \boldsymbol{t}_{\text{hom}}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right) &= \prod_{k=1}^{K} \frac{\exp(\xi_k)}{\sum_{\tilde{k}} \exp(\xi_{\tilde{k}})} \left[\prod_{\ell=1}^{d_c} \prod_{\boldsymbol{g} \in \Sigma^{\ell-1}} \prod_{a \in \Sigma} \frac{\exp(\xi_{\ell,a|\boldsymbol{g},k})}{\sum_{\tilde{a}} \exp(\xi_{\ell,\tilde{a}|\boldsymbol{g},k})} \right] \cdot \\ &\quad \left[\prod_{\boldsymbol{b} \in \Sigma^{d_c}} \prod_{a \in \Sigma} \frac{\exp(\xi_{a|\boldsymbol{b},k})}{\sum_{\tilde{a}} \exp(\xi_{\tilde{a}|\boldsymbol{b},k})} \right] \end{aligned} \tag{3.81}$$

yielding the transformed product-Dirichlet prior for homogeneous Markov models

$$\begin{aligned} q(\boldsymbol{\xi} \mid \boldsymbol{\alpha}) &= \frac{\Gamma(\alpha_\cdot)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \left(\frac{\exp(\xi_k)}{\sum_{\tilde{k}} \exp(\xi_{\tilde{k}})} \right)^{\alpha_k} \cdot \\ &\quad \left[\prod_{\ell=1}^{d_c} \prod_{\boldsymbol{g} \in \Sigma^{\ell-1}} \frac{\Gamma(\alpha_{\ell,\cdot|\boldsymbol{g},k})}{\prod_{a \in \Sigma} \Gamma(\alpha_{\ell,a|\boldsymbol{g},k})} \prod_{a \in \Sigma} \left(\frac{\exp(\xi_{\ell,a|\boldsymbol{g},k})}{\sum_{\tilde{a}} \exp(\xi_{\ell,\tilde{a}|\boldsymbol{g},k})} \right)^{\alpha_{\ell,a|\boldsymbol{g},k}} \right] \cdot \\ &\quad \left[\prod_{\boldsymbol{b} \in \Sigma^{d_c}} \frac{\Gamma(\alpha_{\cdot|\boldsymbol{b},k} \cdot (L_E - d_c))}{\prod_{a \in \Sigma} \Gamma(\alpha_{a|\boldsymbol{b},k} \cdot (L_E - d_c))} \prod_{a \in \Sigma} \left(\frac{\exp(\xi_{a|\boldsymbol{b},k})}{\sum_{\tilde{a}} \exp(\xi_{\tilde{a}|\boldsymbol{b},k})} \right)^{\alpha_{a|\boldsymbol{b},k} \cdot (L_E - d_c)} \right]. \end{aligned} \tag{3.82}$$

The hyper-parameters of this prior are chosen according to the assumption of uniform pseudo-data in the same manner as for inhomogeneous Markov models.

Since the periodic Markov model is parameterized in close analogy to homogeneous Markov models, we can adapt this product-Dirichlet prior with a slight modification of hyper-parameters for the periodic Markov model as well. For the mixture parameters over the three frames, $\beta_{f|c}$, we define hyper-parameters $\alpha_{f|c} = \frac{\alpha_k}{3}$ assuming that all frames occur with equal probabilities. The parameters of the initial probability distribution depend on the chosen frame f. Hence, we use the transformed product-Dirichlet prior in analogy to the initial distribution of homogeneous Markov models, but set the ESS to $\frac{\alpha_k}{3}$. Finally, the parameters of the periodic part are used for a single frames, but each of the sets parameters is used only every three basepairs. Hence, we use the original ESS α_k, but adapt the expected length



(a) $\alpha = 0.2$ (b) $\alpha = 1$ (c) $\alpha = 5$

Figure 3.6.: The transformed Dirichlet prior (red line) for one free parameter ξ in comparison to the Laplace prior (green line) and Gaussian prior (black line) for different values of hyper-parameters. All densities are plotted on a logarithmic scale, and the hyper-parameters of the Gaussian and Laplace prior are chosen such that all three priors achieve the same maximum value.

L_E, i.e. the expected number of symbols in the sequence aside the initial three positions, to $L'_E = \frac{L_E}{3}$.

Compared to the Gaussian or Laplace priors, the Dirichlet prior has the advantage that its hyper-parameters are easily interpretable in terms of pseudo-data, and it is the commonly used prior when learning Markov models by the generative MAP principle. Hence, it allows for an unbiased comparison of the MSP and MAP principle for the same model, i.e. Markov models of the same order. When using BDeu hyper-parameters, it also allows for a comparison of Markov models of different orders learned by MAP or MSP using the same a-priori information (Keilwagen et al., 2010b).

Figure 3.6 illustrates the densities of Gaussian, Laplace and the transformed Dirichlet for one free parameter ξ using different values of the hyper-parameters α, while the hyper-parameters of the Gaussian and Laplace priors are chosen such that all densities achieve the same maximum value. The densities are plotted on a logarithmic scale to illustrate the linear characteristic of the Laplace prior and the quadratic characteristic of the Gaussian prior. Interestingly, the transformed Dirichlet prior lies in between these two extremes. It shows an almost quadratic characteristic near the maximum, whereas it is linear in the far tails.

3.4.3. Gaussian density: Normal-Gamma and Normal-Wishart priors

A conjugate prior for the Gaussian density with mean μ and precision λ presented in section 3.3.2.1 (p. 25) is the normal-gamma density with a-priori mean μ_0, equivalent sample size γ, shape parameter τ_1 and rate parameters τ_2:

$$p(\mu, \lambda|\mu_0, \gamma, \tau_1, \tau_2) = \frac{\tau_2^{\tau_1}\sqrt{\gamma}}{\Gamma(\tau_1)\sqrt{2\pi}}\lambda^{\tau_1-\frac{1}{2}}e^{-\lambda\left[\tau_2+\frac{1}{2}\gamma(\mu-\mu_0)^2\right]}. \tag{3.83}$$

The a-priori mean defines the a-priorily expected value of the mean parameter μ, while the shape and scale parameter model a-priori assumptions about the precision λ. If we a-priorily

expect a mean precision λ_0 with variance λ_1, we can derive the corresponding values of τ_1 and τ_2 from the expectation $\frac{\tau_1}{\tau_2}$ and variance $\frac{\tau_1}{\tau_2^2}$ of the Gamma density, leading to $\tau_1 = \frac{\lambda_0^2}{\lambda_1}$ and $\tau_2 = \frac{\lambda_0}{\lambda_1}$. The confidence in the a-priori information about the mean is represented by the equivalent sample size γ, which again can be interpreted as the size of a set of pseudo-data, which has been observed before seeing the actual training data.

As we parameterize the Gaussian density in terms of μ and κ in section 3.3.2.1, we need to transform the normal-gamma density accordingly. The Jacobian amounts to $\exp(\kappa)$ in this case, leading to the transformed normal-gamma density

$$p(\mu, \kappa|\mu_0, \gamma, \tau_1, \tau_2) = \frac{\tau_2^{\tau_1}\sqrt{\gamma}}{\Gamma(\tau_1)\sqrt{2\pi}} e^{\kappa\left(\tau_1+\frac{1}{2}\right)} e^{-\exp(\kappa)\left[\tau_2+\frac{1}{2}\gamma(\mu-\mu_0)^2\right]}. \tag{3.84}$$

We also use bivariate Gaussian densities in this work (see section 3.3.2.1) to model the joint distribution of motif occurrences in a set of sequences in section 4.2. A conjugate prior for the multivariate Gaussian density $\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Lambda})$ is the normal-Wishart density (DeGroot, 2004), which amounts to the product of a multivariate Gaussian prior for the mean vector $\boldsymbol{\mu}$ and a Wishart density for the precision matrix $\boldsymbol{\Lambda}$

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0, \gamma, \alpha) = \mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \gamma\boldsymbol{\Lambda}) \cdot \mathcal{W}(\boldsymbol{\Lambda}|\boldsymbol{\Lambda}_0, \alpha), \tag{3.85}$$

where $\boldsymbol{\mu}_0$ denotes the vector of prior means, $\boldsymbol{\Lambda}_0$ denotes the a-priori precision matrix, and γ and α denote the equivalent sample sizes of the Gaussian and the Wishart component of the prior. In many cases it might be reasonable to set $\gamma = \alpha$, because other assignments would contradict the concept of a-priorily observed pseudo-data. However, there are cases where we might feel more confident in our a-priori assumptions about the mean vector than the precision matrix, or vice versa.

The Gaussian component of the prior is defined as (DeGroot, 2004)

$$\mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \gamma\boldsymbol{\Lambda}) = \frac{\det(\gamma\boldsymbol{\Lambda})^{1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{\mu}_0)^T \gamma\boldsymbol{\Lambda}(\boldsymbol{\mu}-\boldsymbol{\mu}_0)\right), \tag{3.86}$$

and the Wishart component is defined as

$$\mathcal{W}(\boldsymbol{\Lambda}|\boldsymbol{\Lambda}_0, \alpha) \propto \det(\boldsymbol{\Lambda}_0)^{\alpha/2} \det(\boldsymbol{\Lambda})^{\frac{\alpha-D-1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\boldsymbol{\Lambda}_0\boldsymbol{\Lambda})\right). \tag{3.87}$$

A-priorily, we assume that the random variables at the different dimensions of the multivariate Gaussian density are statistically independent. We can model this assumption by setting all off-diagonal elements of the a-priori precision matrix $\boldsymbol{\Lambda}_0$ to zero (DeGroot, 2004). Hence, the trace of the matrix-product $\mathrm{tr}(\boldsymbol{\Lambda}_0\boldsymbol{\Lambda})$ is equal to the sum of the products of the on-diagonal elements of $\boldsymbol{\Lambda}_0$ and $\boldsymbol{\Lambda}$, and the determinant $\det(\boldsymbol{\Lambda}_0)$ is equal to the product of the on-diagonal elements of $\boldsymbol{\Lambda}_0$

$$\mathcal{W}(\boldsymbol{\Lambda}|\boldsymbol{\Lambda}_0, \alpha) \propto \left(\prod_{d=1}^{D} \lambda_{0,d}\right)^{\alpha/2} \det(\boldsymbol{\Lambda})^{\frac{\alpha-D-1}{2}} \exp\left(-\frac{1}{2}\sum_{d=1}^{D} \lambda_{0,d}\lambda_d\right). \tag{3.88}$$

In section 3.3.2.1, we define a transformation $\boldsymbol{\Lambda}(\boldsymbol{\kappa}, r_{1,2})$ for the precision matrix of a bivariate Gaussian density. In the following we want to transform the normal-Wishart density according to this transformation for the bivariate case.

The Jacobian of the transformation is the product of the on-diagonal elements of the Jacobi matrix in this case, because the derivatives of $\lambda_i = \exp(\kappa_i)$ are zero for the other parameters $\kappa_j, j \neq i$ and $r_{1,2}$:

$$\begin{aligned} \frac{\partial \boldsymbol{\Lambda}(\boldsymbol{\kappa}, r_{1,2})}{\partial (\boldsymbol{\kappa}, r_{1,2})} &= \left| \prod_{d=1}^{2} \exp\left(\frac{3}{2}\kappa_d\right) \left(-\operatorname{sech}(r_{1,2})^2\right) \right| \\ &= \prod_{d=1}^{2} \exp\left(\frac{3}{2}\kappa_d\right) \operatorname{sech}(r_{1,2})^2, \end{aligned} \tag{3.89}$$

where $\operatorname{sech}(.)^2$ denotes the square of the hyperbolic secant, which is the first derivative of the hyperbolic tangent.

Considering the definition of $\boldsymbol{\Lambda}(\boldsymbol{\kappa}, r_{1,2})$ in equation (3.59) (p. 26), we find that we can factor $\sqrt{\exp(\kappa_i)}$ out of row i and we can factor $\sqrt{\exp(\kappa_j)}$ out of column j of the determinant $\det(\boldsymbol{\Lambda}(\boldsymbol{\kappa}, r_{1,2}))$, resulting in

$$\det(\boldsymbol{\Lambda}(\boldsymbol{\kappa}, r_{1,2})) = \det\begin{pmatrix} 1 & -\tanh(r_{1,2}) \\ -\tanh(r_{1,2}) & 1 \end{pmatrix} \prod_{d=1}^{2} \exp(\kappa_d). \tag{3.90}$$

In the following, we denote

$$\boldsymbol{T}(r_{1,2}) = \begin{pmatrix} 1 & -\tanh(r_{1,2}) \\ -\tanh(r_{1,2}) & 1 \end{pmatrix}. \tag{3.91}$$

If we insert all of these results into equation (3.88) for the bivariate case, we obtain

$$\begin{aligned} \mathcal{W}(\boldsymbol{\kappa}, r_{1,2}|\boldsymbol{\Lambda}_0, \alpha) \propto & \left(\prod_{d=1}^{2} \lambda_{0,d}\right)^{\alpha/2} \left[\det(\boldsymbol{T}(r_{1,2})) \prod_{d=1}^{2} \exp(\kappa_d)\right]^{\frac{\alpha-3}{2}} \cdot \\ & \exp\left(-\frac{1}{2}\sum_{d=1}^{2} \lambda_{0,d} \exp(\kappa_d)\right) \cdot \prod_{d=1}^{2} \exp\left(\frac{3}{2}\kappa_d\right) \operatorname{sech}(r_{1,2})^2, \end{aligned} \tag{3.92}$$

which we can rewrite as follows:

$$\begin{aligned} \mathcal{W}(\boldsymbol{\kappa}, r_{1,2}|\boldsymbol{\Lambda}_0, \alpha) \propto & \left[\prod_{d=1}^{2} \left(\frac{\lambda_{0,d}}{2}\right)^{\alpha/2} \exp\left(\kappa_d \frac{\alpha}{2}\right) \exp\left(-\frac{\lambda_{0,d}}{2}\exp(\kappa_d)\right)\right] \cdot \\ & \det(\boldsymbol{T}(r_{1,2}))^{\frac{\alpha-3}{2}} \operatorname{sech}(r_{1,2})^2 \end{aligned} \tag{3.93}$$

$$\propto \left[\prod_{d=1}^{2} \mathcal{G}\left(\kappa_d \middle| \frac{\alpha}{2}, \frac{\lambda_{0,d}}{2}\right)\right] \left(1 - \tanh(r_{1,2})^2\right)^{\frac{\alpha-3}{2}} \operatorname{sech}(r_{1,2})^2 \tag{3.94}$$

Additionally, we transform the Gaussian component according to $\mu_d = L\frac{\exp(\nu_d)}{1+\exp(\nu_d)}, \nu_d \in \mathbb{R}$ of

equation (3.61) (p. 27). The Jacobian of this transformation amounts to

$$\frac{\partial \boldsymbol{\mu}(\boldsymbol{\nu})}{\partial \boldsymbol{\nu}} = \prod_{d=1}^{2} L \frac{\exp(\nu_d)}{(1+\exp(\nu_d))^2}, \tag{3.95}$$

resulting in the transformed Gaussian density

$$\mathcal{N}(\boldsymbol{\nu}|\boldsymbol{\mu}_0, \gamma\boldsymbol{\Lambda}) = \frac{\det(\gamma\boldsymbol{\Lambda})^{1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{\mu}(\boldsymbol{\nu})-\boldsymbol{\mu}_0\right)^T \gamma\boldsymbol{\Lambda}\left(\boldsymbol{\mu}(\boldsymbol{\nu})-\boldsymbol{\mu}_0\right)\right) \prod_{d=1}^{2} L \frac{\exp(\nu_d)}{(1+\exp(\nu_d))^2}.$$

The transformed normal-Wishart density of $\boldsymbol{\nu}$, $\boldsymbol{\kappa}$, and $r_{1,2}$ with hyper-parameters γ, α, $\boldsymbol{\Lambda}_0$, and $\boldsymbol{\mu}_0$ is then defined as the product of the transformed Wishart density and the transformed Gaussian density, yielding

$$p(\boldsymbol{\nu}, \boldsymbol{\kappa}, r_{1,2}|\boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0, \gamma, \alpha) = \mathcal{N}(\boldsymbol{\nu}|\boldsymbol{\mu}_0, \gamma\boldsymbol{\Lambda}(\boldsymbol{\kappa}, r_{1,2})) \cdot \mathcal{W}(\boldsymbol{\kappa}, r_{1,2}|\boldsymbol{\Lambda}_0, \alpha). \tag{3.96}$$

3.4.4. Gamma density: Conjugate priors for the exponential family

In case of the gamma density defined in section 3.3.2.2, no conjugate prior is readily available. To derive such a conjugate prior, we use the general definition of a conjugate prior for the exponential family (Bishop, 2006).

All densities stemming from the exponential family for values $\boldsymbol{y} \in \mathbb{R}^{D'}$ can be expressed in terms of abstract parameters $\boldsymbol{\eta} \in \mathbb{R}^D$

$$P(\boldsymbol{y}|\boldsymbol{\eta}) = h(\boldsymbol{y}) g(\boldsymbol{\eta}) e^{\boldsymbol{\eta}^T \boldsymbol{u}(\boldsymbol{y})}, \tag{3.97}$$

where $h(\boldsymbol{y})$ is a normalization constant depending on the data $\boldsymbol{y}$, $g(\boldsymbol{\eta})$ is a normalization constant depending on the parameters $\boldsymbol{\eta}$, and $\boldsymbol{u}(\boldsymbol{y})$ is a function $\boldsymbol{u} : \mathbb{P}^{D'} \rightarrow \mathbb{Q}^D, \mathbb{P}, \mathbb{Q} \subseteq \mathbb{R}$.

The conjugate prior for the exponential family is then defined as

$$p(\boldsymbol{\eta}|\boldsymbol{\chi}, \nu) = f(\boldsymbol{\chi}, \nu) g(\boldsymbol{\eta})^{\nu} e^{\nu \boldsymbol{\eta}^T \boldsymbol{\chi}}, \tag{3.98}$$

where ν is the equivalent sample size, $\boldsymbol{\chi}$ is the vector of specific hyper-parameters, and $f(\boldsymbol{\chi}, \nu)$ is a normalization constant depending on the hyper-parameters. Comparing equation (3.62) (p. 27) to the definition of the exponential family of equation (3.97), we define

$$\begin{aligned} \boldsymbol{\eta} &:= (a, b) \\ h(y) &:= y^{-1} \\ g(a, b) &:= \frac{b^a}{\Gamma(a)} \\ \boldsymbol{u}(y) &:= (\log(y), -y) \end{aligned}$$

With these prerequisites, we can now define a conjugate prior with equivalent sample size α

and specific hyper-parameters χ_1 and χ_2 for the gamma density as

$$p(a, b|\chi_1, \chi_2, \alpha) \quad = \quad f(\chi_1, \chi_2, \alpha) \left(\frac{b^a}{\Gamma(a)} \right)^{\alpha} e^{\alpha(a\chi_1 + b\chi_2)}. \tag{3.99}$$

Despite this prior being conjugate to the gamma density, we are yet missing an intuitive interpretation of the hyper-parameters χ_1 and χ_2. With the goal of obtaining such an interpretation, we compare the definition of the prior to the likelihood of a set of independent gamma distributed values $\boldsymbol{y}$:

$$\mathcal{G}(\boldsymbol{y}|a, b) = e^{-\sum_{n=1}^{N} \log(y_n)} \left(\frac{b^a}{\Gamma(a)} \right)^{N} e^{a \sum_{n=1}^{N} \log(y_n) - b \sum_{n=1}^{N} y_n} \tag{3.100}$$

As expected, the equivalent sample size α corresponds to the size of the data set N. By factoring N out of the exponent, we obtain

$$\mathcal{G}(\boldsymbol{y}|a, b) = e^{-N \log(\bar{\boldsymbol{y}}_{\text{geo}})} \left(\frac{b^a}{\Gamma(a)} \right)^{N} e^{N[a \log(\bar{\boldsymbol{y}}_{\text{geo}}) - b\bar{\boldsymbol{y}}_{\text{ar}}]}, \tag{3.101}$$

where $\bar{\boldsymbol{y}}_{\text{geo}}$ denotes the geometric mean and $\bar{\boldsymbol{y}}_{\text{ar}}$ denotes the arithmetic mean of the values $\boldsymbol{y}$. We can, hence, interpret χ_1 as the logarithm of the expected geometric mean and χ_2 as the expected arithmetic mean, and set the values of both hyper-parameters accordingly.

In section 3.3.2.2, we defined a transformation for the shape and rate parameter of the gamma density as $a = \exp(\gamma)$ and $b = \exp(\beta)$ to obtain positive values a and b for parameters $\gamma, \beta \in \mathbb{R}$. The Jacobian is $\exp(\gamma)\exp(\beta)$ in this case, resulting in the transformed conjugate prior for the gamma density with parameters γ and β

$$p(\gamma, \beta|\chi_1, \chi_2, \alpha) \quad = \quad f(\chi_1, \chi_2, \alpha) \left(\frac{e^{\beta \exp(\gamma)}}{\Gamma(e^a)} \right)^{\alpha} e^{\alpha[\exp(\gamma)\chi_1 + \exp(\beta)\chi_2]} \cdot e^{\gamma} \cdot e^{\beta}. \tag{3.102}$$

The normalization constant $f(\chi_1, \chi_2, \alpha)$ must be chosen such that $p(\gamma, \beta|\chi_1, \chi_2, \alpha)$ defines a proper density, i.e. $\forall\gamma, \forall\beta : p(\gamma, \beta|\chi_1, \chi_2, \alpha) > 0$ and

$$\int_{\mathbb{R}} \int_{\mathbb{R}} p(\gamma, \beta|\chi_1, \chi_2, \alpha) \mathrm{d}\gamma \mathrm{d}\beta = 1. \tag{3.103}$$

We consequently define $f(\chi_1, \chi_2, \alpha)$ as

$$f(\chi_1, \chi_2, \alpha) \quad = \quad \left(\int_{\mathbb{R}} \int_{\mathbb{R}} \left(\frac{e^{\beta \exp(\gamma)}}{\Gamma(e^a)} \right)^{\alpha} e^{\alpha[\exp(\gamma)\chi_1 + \exp(\beta)\chi_2]} \cdot e^{\gamma} \cdot e^{\beta} \mathrm{d}\gamma \mathrm{d}\beta \right)^{-1}. \tag{3.104}$$

This integral cannot be solved analytically. Hence, we must use numerical integration techniques if we need a normalized prior for the gamma density, which would be the case for instance when using the (supervised) posterior as a criterion for model selection.

3.5. Assessment of classifiers

In many situations, we want to compare the accuracy of a number of classifiers and decide which of these is suited best for a given application. For instance, the classifiers considered may differ in the employed models or in the learning principles used to learn the parameters. To this end, we learn each classifier on a training data set and test its accuracy on an independent test data set. Classification performance may be measured by a wealth of measures. The performance measures used in this work are presented in section 3.5.1. In some applications, we do not have enough data available to partition the data into dedicated training and test data sets while preserving statistically meaningful results. In section 3.5.2, we present two approaches to overcome this problem, namely cross validation and holdout sampling.

3.5.1. Performance measures

For the definition of performance measures, we assume that we obtain an already trained classifier with parameters $\boldsymbol{\beta}$. This classifier together with its parameters constitutes the class posterior $P(c \,|\, \boldsymbol{x}, \boldsymbol{\beta})$ for a sequence $\boldsymbol{x}$ and class c. We further assume that we obtain a test data set comprising a set of sequences $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and associated class labels $\boldsymbol{c} = (c_1, \ldots, c_N)$. We can now use the classification criterion of equation (3.1) (p. 9) to assign to each sequence $\boldsymbol{x}_n$ the most probable class, which we denote by $c^*(\boldsymbol{x}_n, \boldsymbol{\beta})$.

The *classification rate* of a classifier on the test data set $\boldsymbol{X}, \boldsymbol{c}$ is defined as

$$CR(\boldsymbol{X}, \boldsymbol{c}, \boldsymbol{\beta}) \;=\; \frac{1}{N} \sum_{n=1}^{N} \delta_{c_n, c^*(\boldsymbol{x}_n, \boldsymbol{\beta})}, \tag{3.105}$$

where the Kronecker delta is equal to 1 if both indices are equal and 0 otherwise.

We can define a number of additional performance measures for the case of two-class problems, where we can slightly reformulate the classification criterion. We assign a sequence $\boldsymbol{x}$ to the first class, i.e. $c^*(\boldsymbol{x}_n, \boldsymbol{\beta}) = 1$, if the following inequation holds

$$\frac{P(c = 1 \,|\, \boldsymbol{x}, \boldsymbol{\beta})}{P(c = 2 \,|\, \boldsymbol{x}, \boldsymbol{\beta})} > T, \tag{3.106}$$

or, equivalently

$$\log P(\boldsymbol{x}, c = 1 \,|\, \boldsymbol{\beta}) - \log P(\boldsymbol{x}, c = 2 \,|\, \boldsymbol{\beta}) > \log T, \tag{3.107}$$

and to the second class otherwise, where T denotes a threshold. The difference on the left side of inequation (3.107) is often referred to as *log likelihood ratio*. If the threshold T is equal to 1, we obtain an alternative formulation of equation (3.1). Other values of T may be interpreted as adjustments of the a-priori class probabilities of the two classes: If $T > 1$, we shift the a-priori class probabilities in favor of class 2. If on the other hand $T < 1$, we prefer class 1. Hence, we can control the number of predictions for class 1 and class 2 by means of the threshold T. We denote this dependence of the classification result on the threshold by $c_T^*(\boldsymbol{x}_n, \boldsymbol{\beta})$.

We define the entries of the *confusion matrix*, namely *true positives*, *true negatives*, *false positives*, and *false negatives* for a given threshold T. The true positives (TP) are the number of sequences $\boldsymbol{x}_n$ belonging to the positive class, i.e. $c_n = 1$, which are also assigned to class 1, i.e. $c_T^*(\boldsymbol{x}_n, \boldsymbol{\beta}) = 1$, and the true negatives (TN) are the number of sequences $\boldsymbol{x}_n$ belonging to the negative class, i.e. $c_n = 2$, which are also assigned to class 2, i.e. $c_T^*(\boldsymbol{x}_n, \boldsymbol{\beta}) = 2$:

$$TP(T) = \sum_{n=1}^{N} \delta_{c_n,1} \cdot \delta_{c_T^*(\boldsymbol{x}_n,\boldsymbol{\beta}),1} \qquad TN(T) = \sum_{n=1}^{N} \delta_{c_n,2} \cdot \delta_{c_T^*(\boldsymbol{x}_n,\boldsymbol{\beta}),2} \tag{3.108}$$

Considering false predictions, the false positives are the number of sequences $\boldsymbol{x}_n$ belonging to the negative class, i.e. $c_n = 2$, which are erroneously assigned to class 1, i.e. $c_T^*(\boldsymbol{x}_n, \boldsymbol{\beta}) = 1$ and the false negatives are the number of sequences $\boldsymbol{x}_n$ belonging to the positive class, i.e. $c_n = 1$, which are erroneously assigned to class 2, i.e. $c_T^*(\boldsymbol{x}_n, \boldsymbol{\beta}) = 2$:

$$FP(T) = \sum_{n=1}^{N} \delta_{c_n,2} \cdot \delta_{c_T^*(\boldsymbol{x}_n,\boldsymbol{\beta}),1} \qquad FN(T) = \sum_{n=1}^{N} \delta_{c_n,1} \cdot \delta_{c_T^*(\boldsymbol{x}_n,\boldsymbol{\beta}),2} \tag{3.109}$$

The *sensitivity* (Sn) of a classifier for threshold T is then defined as the percentage of sequences belonging to the positive class that are classified correctly:

$$Sn(T) \quad = \quad \frac{TP(T)}{TP(T) + FN(T)} \tag{3.110}$$

The sensitivity is also called *recall* in some contexts. The *specificity* (Sp) for threshold T is defined as the number of sequences belonging to the negative class and are classified correctly. It is also common to measure the classification accuracy for the negative class by means of the *false positive rate* (FPR) which is the percentage of falsely classified sequences stemming from the negative class:

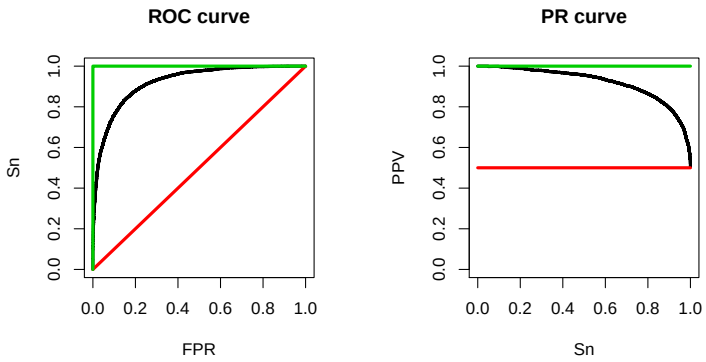
$$Sp(T) = \frac{TN(T)}{TN(T) + FP(T)} \qquad FPR(T) = \frac{FP(T)}{TN(T) + FP(T)} = 1 - Sp(T) \tag{3.111}$$

Finally, we measure the percentage of correctly classified sequences in all sequences assigned to the positive class by the *positive predictive value* and the percentage of correctly classified sequences in all sequences assigned to the negative class by the *negative predictive values*:

$$PPV(T) = \frac{TP(T)}{TP(T) + FP(T)} \qquad NPV(T) = \frac{TN(T)}{TN(T) + FN(T)} \tag{3.112}$$

The positive predictive value is also referred to as *precision* of the classifier.

All measures considered so far depend on the chosen threshold T and are called *point measures*. We define additional measures of classification accuracy which assess the overall performance of classifiers. The *receiver operating characteristic* (ROC) curve is the plot of $Sn(T)$ against $1 - Sp(T) = FPR(T)$ for all possible thresholds T. An example of a ROC curve is given in figure 3.7(a). A classifier which is randomly guessing achieves equal $Sn(T)$ and $FPR(T)$ for each threshold T resulting in a straight line from $(0, 0)$ to $(1, 1)$, which is plotted in red in figure 3.7(a). On the other extreme, a perfect classifier (green line) yields $Sn(T) = 1$ for any threshold T and, hence, regardless of $FPR(T)$. In most applications, the ROC curve



(a) ROC curve.

(b) PR curve on a balanced data set.

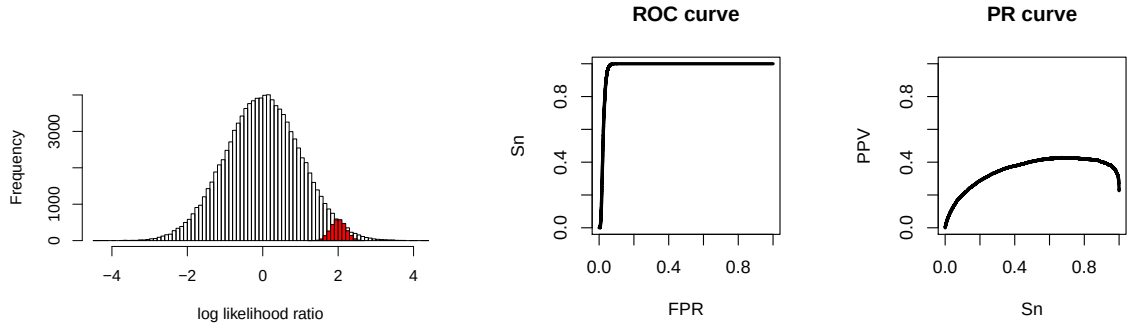
Figure 3.7.: Examples of an ROC and a PR curve. The red line shows the curves of random guessing. The black line illustrates the curves of a classifier which performs better than random guessing but is not perfect. The green line corresponds to a perfect classifier.

of the considered classifier lies between these two extremes as illustrated by the black line in figure 3.7(a). If the ROC curve of a classifier is located below the diagonal of random guessing, classification may be improved by switching class labels.

The comparison of curves is not always a convenient way to compare classifiers, for instance if we want to average classification performance over a number of test data sets. We measure the overall accuracy of a classifier by the *area under the ROC curve* (AUC-ROC), which integrates over all $FPR(T) \in [0, 1]$. The AUC-ROC of random guessing is 0.5, whereas that of a perfect classifier is 1.0. As the test data set $\boldsymbol{X}$ comprises only a finite number of sequences, reasonable values of T correspond to log likelihood ratios that can be observed for elements of $\boldsymbol{X}$, and we obtain at most N discrete points $(FPR(T), Sn(T))$. We compute AUC-ROC by a linear interpolation between these points (Davis and Goadrich, 2006).

The *precision-recall curve* (PR curve) is the plot of $PPV(T)$ against $Sn(T)$ for all possible thresholds T. Figure 3.7(b) illustrates the PR curve of a perfect classifier, a classifier which is randomly guessing, and a realistic classifier. In contrast to the ROC curve, the PR curve is not necessarily monotonic or concave. As Davis and Goadrich (2006) point out, a non-linear interpolation between the points $(Sn(T), PPV(T))$ yields a more accurate approximation of the PR curve than a linear one. If we increase the threshold such that all sequences are classified as negative, we obtain $Sn(T) = 0$ and $PPV(T) = \frac{0}{0}$. According to the interpolation proposed by (Davis and Goadrich, 2006), $PPV(T)$ must be set to the last defined value in this case. Using this interpolation, we may integrate $PPV(T)$ over all $Sn(T)$ and obtain the *area under the precision-recall curve* (AUC-PR).

If the test data set contains an unbalanced number of sequences for the two classes, the ROC curve or AUC-ROC may be less suited for comparing classifiers: Assume that the test data set contains 10 sequences from the positive class and 1000 sequences from the negative class. Further assume a classifier, which assigns a log likelihood ratio of -1 to 990 out of the 1000 negative sequences and a log likelihood ratio of 1 to the remaining 10 negative and all positive sequences. With these log likelihood ratios, we still obtain $Sn(0) = 1$ and $FPR(0) = 0.01$ for



(a) Histogram of scores (b) ROC curve (c) PR curve

Figure 3.8.: ROC and PR curve on an unbalanced data set. Figure (a) shows a histogram of the scores of the negative class (black) and the positive class (red). Both classes are clearly overlapping. Figure (b) and (c) illustrate the ROC and PR curves computed on these scores.

a threshold of 0. By increasing the threshold to 1 we obtain $Sn(1) = 0$ and $FPR(1) = 0$. We interpolate linearly between these points and the resulting AUC-ROC is 0.995, although the positive class completely overlaps with the negative class. Hence, the AUC-ROC of classifiers separating the classes more clearly can be expected between 0.995 and 1.0. A more realistic example of an ROC curve on an unbalanced data set is depicted in figure 3.8(b). In such cases, the area under the *precision-recall curve* (PR curve) may be more meaningful for the comparison of classifiers. Considering the example of unbalanced classes, we obtain $Sn(T) = 1$ and $PPV(T) = 0.5$ for all thresholds $-1 < T < 1$, whereas for $T = 1$ we obtain $Sn(1) = 0$ and $PPV(1) = \frac{0}{0} := 0.5$, as 0.5 is the last defined PPV. The resulting AUC-PR amounts to 0.5, which reflects that the positive class is completely overlapping the negative class. A similar behavior can be observed in figure 3.8(c).

However, this behavior of the ROC curve can also be perceived as an advantage over the PR curve, since we obtain an assessment of classification accuracy that does not depend on the a-priori probability of the two classes (Fawcett, 2006).

If the number of sequences in the positive class considerably exceeds the number of those in the negative class, we use an inverted variant of the PR curve. In this case, we plot NPV against Sp, which is equivalent to the original definition of the PR curve when switching class labels and inverting the sign of all log likelihood ratios. We refer to this curve as PRI curve and to the area under the PRI curve as AUC-PRI.

3.5.2. Cross validation and sampling

The data available in bioinformatics applications are often limited. For instance, typical data sets of transcription factor binding sites comprise between 30 and 300 binding sites. Hence, partitioning the data into training and test data sets entails disadvantages for the comparison of different classifiers. First, the resolution of performance measures is limited to a low number of discrete values. Second, the values of these measures may vary highly depending on the

chosen partitioning and, hence, the generalizability of such results is questionable. Several approaches for a robust measurement of classification accuracy on limited data exist. In the following, we use two of these, namely cross validation and stratified holdout sampling.

A *K-fold cross validation* requires K non-overlapping data sets $(\boldsymbol{X}_1, \boldsymbol{c}_1), \ldots, (\boldsymbol{X}_K, \boldsymbol{c}_K)$. These data sets may either be obtained by partitioning some original data set or originate from biological background, e.g. different chromosomes of an organism. In each cross validation run $k = 1, \ldots, K$, the classifiers of interest are trained on $K - 1$ of the data sets $\cup_{i \neq k}(\boldsymbol{X}_i, \boldsymbol{c}_i)$ and classification performance is tested on the remaining data set $(\boldsymbol{X}_k, \boldsymbol{c}_k)$ using performance measures as presented in section 3.5.1. These measures are then averaged over all K cross validation runs. As a measure of deviation from the average performance we can also compute the standard error over the K runs, which estimates the deviation that can be expected if the cross validation is repeated using another partitioning of the data.

A *K-fold stratified holdout sampling* starts from one original data set $(\boldsymbol{X}, \boldsymbol{c})$. In each of the K iterations, we randomly sample without replacement $p\%$ of the data set as training data set and the remaining $(100 - p)\%$ are used as test data set. In each iteration k the classifiers are trained on the sampled training data set and the performance measures are computed on the non-overlapping test data set. Like for cross validation, the performance measures are averaged over the K iterations and the standard error of these measures is computed. Compared to cross-validation, stratified holdout sampling permits the assessment of classifiers on smaller data sets due to is sampling procedure, which allows for a larger number K of iterations.

Cross validation and holdout sampling experiments are called *stratified*, if we ensure that the proportion of sequences stemming from the different classes is equal in each training and test data set. In order to prevent overlaps between training and test data, we do not sample sub-sequences of the same original sequences into different partitions of the data, although this could lead to more balanced sizes of the sampled training and test data over different iterations.

4. Applications

This chapter comprises applications of the Bayesian discriminative MSP principles to a variety of problems in bioinformatics that are related to gene regulation. In the first section, we employ the discriminative MSP principle and the generative MAP principle for learning the parameters of Markov models, and we compare the classification accuracy achieved by both learning principles for the recognition of known transcription factor binding sites (TFBSs) stemming from prokaryotes as well as eukaryotes. In real-world applications, we often neither know the exact binding sites of the factors of interest, nor do we know which transcription factor is responsible for the regulation of a set of genes. Rather we obtain a number of genes which are for instance co-expressed and, hence, potentially co-regulated. In the second section, we present an approach for finding motifs in a set of promoters of such genes. This task is often referred to as *de-novo motif discovery* and is still one of the most challenging problems of bioinformatics. The prediction of TFBSs typically suffers from a large number of false-positives. Although a given stretch of DNA is highly similar to the binding sites of a specific transcription factor, this very stretch may not be accessible to a transcription factor because of chromatin structure. One major building block of chromatin organization are nucleosomes. DNA bound in nucleosomes can not be bound by transcription factors and consequently may be no functional regulatory element. With the goal of excluding these parts of DNA from the prediction of TFBSs, a new method for the prediction of nucleosome positions from DNA sequence is proposed in the third section. Although not directly involved in gene regulation, splicing of pre-mRNA yielding mature mRNA does affect the final gene product, i.e. the protein or enzyme, of most genes in eukaryotic organisms. In the fourth section, we present a novel approach for the prediction of splice donor sites and we demonstrate its utility on splice donor sites stemming from different organisms. During the last years another mode of gene regulation besides binding of transcription factors has come into focus, namely miRNAs. MiRNAs are short RNA sequences which bind to mRNA and either inhibit translation or facilitate the degradation of the bound mRNA. We present a new approach for predicting the targets of a given miRNA in the fifth section.

4.1. Recognition of transcription factor binding sites

Transcription factors bind to short stretches of DNA in the promoter regions of genes and, as a consequence, activate or repress the transcription of that gene (see also section 2.1). The prediction of these TFBSs has been a major topic of bioinformatics almost since its beginnings and is still a challenging problem today.

4.1.1. Background

As one of the first, Staden (1984) uses position weight matrix (PWM) models – also referred to as position specific scoring matrices (PSSMs) – to model promoters of *Escherichia coli* including the -35 and -10 boxes. These boxes correspond to the binding sites of the prokaryotic transcription factor σ^{70}, which are also considered in this work. Staden adapts the concept of PWM models from (Stormo et al., 1982), who employ PWM models to model translation initiation sites of *E. coli*. While Stormo et al. learn the parameters of the PWM model by the *perceptron algorithm*, which originally emerged in the field of artificial neural networks, Staden defines these parameters as the logarithms of the relative frequencies of each nucleotide at each position. Besides some specific treatment of zero-occurrences, Staden's approach corresponds to learning the parameters of the PWM by the maximum likelihood (ML) principle as presented in section 3.1, where the score defined by (Staden, 1984) corresponds to the log-likelihood.

To date, PWM models are at the heart of popular tools like Match$^{\text{TM}}$ (Kel et al., 2003) or P-Match (Chekmenev et al., 2005), which use weight matrices from the Transfac® database (Wingender et al., 1996; Matys et al., 2006). PWM models are also used by Berman et al. (2002) and Pape et al. (2009) to represent binding sites of individual transcription factors in cis-regulatory modules. Weindl et al. (2007) use non-standard weight matrices that directly represent binding energies to predict binding sites of the σ^{70} transcription factor. Many approaches for the de-novo discovery of motifs also rely on PWM models (see also section 4.2).

Zhang and Marr (1993) generalize the PWM model to dinucleotide frequencies in an approach termed *weight array method* or *weight array model* (WAM) and apply it to the recognition of splice donor sites of *Schizosaccharomyces pombe*. WAM models are equivalent to Markov models of order 1 as presented in section 3.3.1, and accordingly drop the assumption of statistical independence between positions. Zhang and Marr (1993) show that modelling dinucleotide frequencies and, hence, dependencies between directly adjacent positions can improve the prediction accuracy of splice donor sites. Markov models of order 1 are also employed by (Salzberg, 1997) for the detection of eukaryotic translation initiation sites, splice donor sites, and splice acceptor sites. Salzberg (1997) also presents a dynamic programming approach for computing consensus sequences from the conditional probabilities of a first order Markov model.

Current research focusses on two main directions: improving the statistical models for representing TFBSs and utilizing enhanced principles for learning the parameters of these models. Following the former direction, Ellrott et al. (2002) propose permuted Markov models (PMMs) for the prediction of binding sites of the human transcription factor HNF4α. Permuted Markov

models allow for permuting the positions of the binding sites before defining dependencies according to a standard Markov model and, as a consequence, capture non-adjacent dependencies between positions. In (Ellrott et al., 2002) this permutation is chosen such that the dependency between adjacent positions after the permutation is maximized as measured by χ^2.

The concept of permuted Markov models is extended to permuted variable length Markov models (PVLMMs) by Zhao et al. (2005), who consider the prediction of splice donor sites from SpliceDB (Burset et al., 2001) and transcription factor binding sites from Transfac®. PVLMMs combine PMMs with the concept of variable length Markov models (Rissanen, 1983; Bühlmann and Wyner, 1999), which are also termed variable length Markov chains or variable order Markov (VOM) models. In VOM models, the length of the context at each position, i.e. the number of preceding symbols taken into account (see section 3.3.1), may depend on the specific symbols observed at the context positions. This approach allows for modelling only those dependencies that are supported by the data, and can greatly reduce the number of parameters to be estimated. The variable order approach is also used by (Ben-Gal et al., 2005) in conjunction with modelling non-adjacent dependencies by Bayesian networks (Heckerman et al., 1995) for predicting the binding sites of the σ^{70} transcription factor of *E. coli*. These variable order Bayesian networks are also applied to eukaryotic TFBSs from Transfac® in (Posch et al., 2007).

Bayesian networks and Bayesian trees are also proposed by Barash et al. (2003), who compare the prediction performance of PWM models to Bayesian trees, mixtures of PWMs, and mixtures of Bayesian trees on data sets of TFBSs from Transfac®. Barash et al. (2003) also investigate the utility of these models for de-novo motif discovery. Mixture models are also employed by King and Roth (2003), who propose a model that is a mixture of singleton PWM models, i.e. PWM models with parameters estimated from a single sequence, for the prediction of TFBSs from Transfac®.

Other approaches are single layer perceptrons using dinucleotide features employed by (Rani et al., 2007) for the prediction if prokaryotic σ-factors, and the ab-initio prediction of TFBSs for transcription factors without known binding sites (Kaplan et al., 2005). Kaplan et al. (2005) demonstrate the utility of their approach by learning general binding preferences of *zinc-finger* transcription factors from known binding sites and transferring these findings to other factors from the same family.

Regarding enhanced learning principles, Yakhnenko et al. (2005) use Markov models of orders 1 to 3 learned by the discriminative maximum conditional likelihood (MCL) principle (see also section 3.2.1) to predict subcellular localization signals of prokaryotic and eukaryotic proteins from the SwissProt database (Boeckmann et al., 2003). Discriminatively learned Markov models are also used by Grau et al. (2007b) for predicting eukaryotic TFBSs from Transfac®. In contrast to (Yakhnenko et al., 2005), the parameters are learned by the Bayesian MSP principle using Gaussian and Laplace priors.

The methods used by Grau et al. (2007b) are a subset of those employed in the following, where we compare the classification performance of Markov models learned by the discriminative MSP principle to that of Markov models learned by the generative MAP principle. The main goal of this study is to investigate the potential of Bayesian discriminative learning

principles compared to generative ones. For this reason, we restrict the analyses to models of low complexity, namely Markov models. Another aspect of this study is the comparison of Gaussian, Laplace, and Dirichlet priors employed for the MSP principle.

4.1.2. Data

For the comparison of learning principles and priors, we consider binding sites of the prokaryotic σ^{70}-factor, binding sites of 7 mammalian TFs obtained from the Transfac® database (Wingender et al., 1996; Matys et al., 2006), and binding sites of two TFs from the Jaspar database (Sandelin et al., 2004) stemming from *Arabidopsis thaliana*.

The binding sites of the σ^{70}-factor and the corresponding background data set are those used in (Ben-Gal et al., 2005) stemming from *E. coli*. The foreground data set contains 238 TFBSs of length 12 that are present in the PromEC database[1] (Lisser and Margalit, 1993) as well as RegulonDB 3.0 (Salgado et al., 2000) and that could be mapped uniquely to non-coding regions of the *E. coli* genome. The background data set is selected such that it should not contain any σ^{70} TFBSs. To this end, Ben-Gal et al. (2005) extract intergenic regions from the complete *E. coli* genome that are located between two tail-to-tail genes situated on opposite strands. This data set contains 472 sequences of different lengths comprising $\sim$ 77.6 kbp in total.

The sets of mammalian TFBSs comprise binding sites of length 16 bp of the TF AP-1 (112 TFBSs), steroid hormone receptors (AR/GR/PR, 104 TFBSs), C/EBP (149 TFBSs), GATA (110 TFBSs), NF1 (96 TFBSs), Sp1 (257 TFBSs), and thyroid hormone receptor-like factors (Thyroid, 127 TFBSs) obtained from Transfac®, which are also considered in (Posch et al., 2007; Grau et al., 2007b). AP-1 stands for "activator protein 1", which is a dimeric transcription factor playing a role in cell proliferation and survival (Karin et al., 1997). The data set of steroid hormone receptors contains binding sites of androgen receptors (AR), glucocorticoid receptors (GR), and progesterone receptors (PR), which are activated by binding of a steroid hormone, and in turn bind to DNA and regulate gene expression (Evans, 1988). C/EBP is an acronym for CCAAT-enhancer binding proteins, which is a family of at least six TFs binding to the CCAAT box and regulating several cellular processes like proliferation, differentiation, or inflammation (Ramji and Foka, 2002). The GATA family contains at least three distinguishable TFs, which bind to variations of the motif GATA and play a role in development (Ko and Engel, 1993). X-ray structures of C/EBP and GATA have been presented in section 2.1. NF1 stands for "nuclear factor 1", a family of basal TFs that play a role in chromatin remodelling due to their competition with the formation of nucleosomes (Blomquist et al., 1999; Chikhirzhina et al., 2008). Sp1 is a family of TFs that is essential in early stages of embryonic development (Marin et al., 1997). Thyroid hormone receptor-like factors bind to the hormone thyroid and act in a similar way as described for the steroid hormone receptors (Evans, 1988). As background data set we choose second exons of human genes following (Kel et al., 2003), which are also selected in (Posch et al., 2007; Grau et al., 2007b). This choice minimizes the chance of false negatives in the background data set. However, it also simplifies classification

[1] `http://margalit.huji.ac.il/promec/index.html`

compared to a prediction of TFBSs in (non-coding) promoter sequences. The background data set contains 267 exons with a total length of ~ 68.1 kbp.

In order to diversify the comparison, we additionally include TFBSs of two TFs of the plant *Arabidopsis thaliana* into the study. These are AGL3 and Agamous (AG), which both belong to the family of MADS-box genes and play a role in floral meristem identity (Mizukami et al., 1996). We obtain 90 AGL3 binding sites of length 11 and 97 AG binding sites of length 10 from the Jaspar database (Sandelin et al., 2004). We randomly choose a selection of 97 promoter sequences of *A. thaliana* available from TAIR[2] comprising $\sim 48,5$ kbp in total as a background data set.

4.1.3. Model

As noted in the introduction, Markov models including PWM models and WAM models are widely used for the prediction of TFBSs. In most of the cases these are learned by one of the generative learning principles, namely ML or MAP. Here, we propose to learn the parameters of Markov models by the discriminative MSP principle (see section 3.2.2). Since the supervised posterior amounts to the product of the conditional likelihood and a prior on the parameters of the models and the a-priori probabilities of the classes, the choice of the prior influences the parameters learned and, consequently, the prediction of TFBSs. Here we compare three different priors, namely product-Dirichlet priors (section 3.4.2), Gaussian, and Laplace priors (section 3.4.1). Product-Dirichlet priors are a common choice when learning the parameters of Markov models or more general Bayesian networks by the generative MAP principle (Heckerman et al., 1995; Ben-Gal et al., 2005; Keilwagen et al., 2010b), whereas Gaussian and Laplace priors are often applied to the parameters of Markov random fields (Chen and Rosenfeld, 1999) and logistic regression (Madigan et al., 2005; Genkin et al., 2005; Cawley et al., 2007). However, as the product-Dirichlet prior is conjugate to the likelihood of Markov models, we consider it a "natural" choice for this class of models. Utilizing the transformed product-Dirichlet prior presented in section 3.4.2, we can eliminate the influence of the prior from the analysis by using exactly the same hyper-parameters representing identical a-priori assumptions for the generative MAP principle and the discriminative MSP principle.

Markov models are already defined in section 3.3.1, while the employed priors are defined in section 3.4. What is left to specify are the considered orders of the Markov models and the hyper-parameters of the employed priors. We denote by *tfbs* the class of TFBSs and by *bg* the class of background sequences, i.e. $\mathcal{C} = \{tfbs, bg\}$ and $|\mathcal{C}| = 2$. Here, we test Markov models of order $d_{tfbs} \in \{0, 1\}$, i.e. PWM models and WAM models, for modelling TFBSs, and Markov models of order $d_{bg} \in \{0, 1, 2, 3, 4\}$ for modelling background sequences. Regarding the product-Dirichlet prior employed for the MAP and the MSP principle, we consistently specify the equivalent sample size (ESS) for the class of TFBSs as 4 and for the background class as 1024, reflecting that we expect TFBSs occur with relative low frequency on a genomic scale.

For Gaussian and Laplace priors, we must define the a-prior mean μ_c and variance σ_c^2 for the prior on the a-priori class probabilities and the constants κ_c that are used for determining the

[2] `http://www.arabidopsis.org/`

variances for the parameters of the Markov models, while the corresponding prior means are fixed to zero. We derive μ_c and σ_c^2 from the results of a study by Stepanova et al. (2005), who investigate the frequency of occurrence of the binding sites of 184 different TFs in mammalian genomes. We use equation (3.38) (p. 20) on the relative frequencies reported by Stepanova et al. (2005) to derive the prior mean and variance in the space of $\boldsymbol{\xi}$-parameters, resulting in $\mu_{tfbs} = -8.634$ and $\sigma_{tfbs}^2 = 5.082$ (Grau et al., 2007b).

To determine appropriate values of κ_c, we perform a pre-study using the binding sites of the mammalian TF Sp1. For this data set, we perform a grid search on κ_{tfbs} (0.001 to 5, 12 values) and κ_{bg} (0.0005 to 0.5, 10 values), where we fix the order of the TFBS (foreground) model to $d_{tfbs} = 0$ and vary the background order from $d_{bg} = 0$ to $d_{bg} = 3$. For each combination, we use a 100-fold stratified holdout sampling procedure (see section 3.5.2) and determine the resulting average AUC-ROC. For each pair $(\kappa_{tfbs}, \kappa_{bg})$, we then average AUC-ROC over all background orders and choose the $(\kappa_{tfbs}^*, \kappa_{bg}^*)$ that yields the maximum AUC-ROC. The values determined by this procedure are $\kappa_{tfbs}^* = 2$ and $\kappa_{bg}^* = 0.005$ for the Gaussian prior and $\kappa_{tfbs}^* = 0.005$ and $\kappa_{bg}^* = 0.002$ for the Laplace prior. We fix these values of the κ_{tfbs} and κ_{bg} in all further analyses, which implies that, using Gaussian and Laplace priors, the results for Sp1 and AUC-ROC are biased by the pre-study. However, we focus on other performance measures, namely Sn, PPV, and AUC-PR, in the following.

4.1.4. Results & Discussion

We compare the classification performance of the generative MAP principle using the product-Dirichlet prior (MAP) and the discriminative MSP principle using the transformed product-Dirichlet prior (MSP-D), the Gaussian prior (MSP-G), and the Laplace prior (MSP-L) on the ten data sets introduced in section 4.1.2. We use Sn for a fixed Sp of 99.9%, PPV for a fixed Sn of 95%, and AUC-PR (see section 3.5.1) as performance measures. Sn for a fixed Sp of 99.9% measures the rate of true TFBSs that are also predicted as TFBSs if we fix the threshold such that we correctly classify 99.9% of the background sequences, or stated differently, such that we erroneously classify 0.1% of the background sequences as TFBSs. PPV for a fixed Sn of 95% measures the rate of correct predictions in all sequences classified as TFBSs if for the same threshold we recover 95% of the true TFBS. AUC-PR is a measure of the overall performance of a classifier. We compute the values of these performance measures for all learning principles tested and all combinations of Markov models in each iteration of a 1000-fold stratified holdout sampling procedure (see section 3.5.2) and report the average performance together with the standard error. The standard error gives an estimate of the range of deviation from the reported average that can be expected if we repeat the experiment. With a probability of 0.95 this deviation is at most two-fold the standard error, or stated differently, we expect a difference exceeding two-fold the standard error by chance with probability 0.05. Hence, we consider a difference of performance exceeding two-fold the standard error significant.

4.1.4.1. Comparison of overall classification performance

Figure 4.1 presents the Sn (first column), PPV (second column), and AUC-PR (third column) achieved by the studied learning principles. For each learning principle, we test all combi-

nations of Markov models of order 0 and 1 in the foreground and Markov models of order 0 to 4 in background. For each learning principle, we choose from these the combination that gained the best classification performance with respect to the current measure. For instance, a Markov model of order 1 in the foreground class (TFBSs) combined with a Markov model of order 2 in the background yields the best performance of all Markov models learned by the MAP principle on the AP-1 data set.

Considering the AP-1 data set, MSP-D achieves the best Sn of 0.675, the best PPV of 0.241, and the best AUC-PR of 0.642. The improvement of classification performance gained by MSP-D over MAP and MSP-L is significant for all three performance measures, since it exceeds two-fold the standard error as indicated by the error bars. The differences between MSP-D and MSP-G, however, are not significant for any of the three performance measures. For all studied classifiers, the optimal combination of the orders of Markov models differs only slightly between the three performance measures. Interestingly, MAP prefers an order of 1 for the foreground class combined with a fairly low order of 2 in the background class, whereas for MSP-D a combination of order 0, i.e. a PWM model, in the foreground and a higher order (3 or 4) in the background yields the best classification performance with respect to all three measures. The latter can be observed for the remaining data sets as well and we scrutinize this observation in more detail in figure 4.3.

Turning to the AR/GR/PR data set, we find a similar pattern, where again MSP-D yields the best classification performance with regard to all three measures. On this data set, MSP-D achieves a Sn of 0.589, a PPV of 0.226, and an AUC-PR of 0.548. MSP-D significantly outperforms all other classifiers considering Sn and AUC-PR, whereas for PPV only the improvement over MSP-L is significant. In contrast to the AP-1 data set, MAP achieves the best classification performance by a combination of two PWM models on the AR/GR/PR data set, while for MSP-D a combination of a PWM model and a higher order Markov model performs best. The choice of optimal orders for MSP-G and MSP-L is more unsteady for this data set than it is for AP-1.

MSP-D also performs best on the data set of C/EBP TFBSs yielding a Sn of 0.323, a PPV of 0.074, and an AUC-PR of 0.321. Considering Sn and AUC-PR, the improvement gained by the discriminatively learned classifiers over MAP is remarkable and highly significant. One explanation for this improvement might be the heterogeneity of the binding sites of the members of the C/EBP family. For this reason the model assumption of a single Markov model is wrong and it is known (Greiner et al., 2005) that in such cases discriminative parameter learning particularly improves classification accuracy. This assumption is supported by the observation that MAP performs best with regard to Sn and AUC-PR for the most complex of the studied models on this data set, which in part can compensate for the heterogeneity of binding sites. Regarding PPV, MSP-L performs significantly worse than MAP, MSP-D, and MSP-G, which yield a comparable PPV. However, PPV reflects the general poor classification performance of all approaches on the C/EBP data set. It states that less than 10% of the predicted BSs are correct leaving more than 90% of false positives if we require 95% of the true BSs to be recovered.

As for the previous data sets, MSP-D performs best on the GATA data set with regard to all three performance measures with a Sn of 0.767, a PPV of 0.468, and an AUC-PR of 0.674.

Figure 4.1.: Classification performance of Markov models using the generative MAP principle (MAP), the discriminative MSP principle with product-Dirichlet prior (MSP-D), Gaussian prior (MSP-G), and Laplace prior (MSP-L). For each combination of learning principle and prior, the best combination of foreground and background order is presented. The corresponding values are given in parentheses, e.g. "MAP: $(1, 3)$" for MAP-trained Markov models of order 1 in the foreground and order 3 in the background. For each data set and each performance measure, the bar of the classifier yielding the best classification performance is marked in green. The error bars indicate a deviation of two-fold standard error in both directions.

However, in this case the improvement over the next best classifier is significant for none of the three performance measures. Comparing MAP and MSP-D, which use the same a-priori information represented by identical hyper-parameters of the product-Dirichlet prior, we find a significant improvement of MSP-D over MAP with respect to PPV and AUC-PR, whereas the values of Sn differ only insignificantly.

MSP-D significantly outperforms MAP on the NF1 data set as well. However, in contrast to Sn and PPV, where MSP-D yields the best values of 0.707 and 0.349, respectively, MSP-G achieves a slightly better AUC-PR of 0.616 using a background model of lower order than MSP-D. Again, the improvement of the best classifier over the next best one is significant for none of the performance measures.

On the Sp1 data set all classifiers perform comparably well as measured by AUC-PR. Although MSP-D yields the largest Sn of 0.761, PPV of 0.538, and AUC-PR of 0.799, its improvement over the best of the remaining classifiers is significant only for Sn. Focusing the comparison on MAP and MSP-D, which use the same prior, we find a significant improvement of MSP-D over MAP with regard to Sn and PPV.

As a last mammalian TF, we consider the BSs of the Thyroid data set. On this data set, MSP-D performs best for none of the three performance measures and is outperformed by MAP yielding a Sn of 0.510, and by MSP-G yielding a PPV of 0.222 and an AUC-PR of 0.508. Comparing MAP and MSP-D we find a significant improvement of MSP-D over MAP with regard to PPV and AUC-PR.

MSP-G also achieves the best Sn of 0.753, the best PPV of 0.494, and the best AUC-PR of 0.720 on the AGL3 data set stemming from *A. thaliana*. The differences between MSP-D and MSP-G are not significant for any of the three performance measures. However, both significantly outperform MAP with regard to Sn and PPV, and MSP-L for all three measures. For the second plant data set, we also find a generally inferior performance of MSP-L compared to the other classifiers. MAP yields the best Sn of 0.768 and the best PPV of 0.505, whereas MSP-D yields the best AUC-PR of 0.759 for the AG data set. MAP and MSP-D gain a significant improvement of Sn over MSP-G and MSP-L. Considering PPV, MAP significantly outperforms the three other approaches, whereas MAP, MSP-G, and MSP-L achieve a significantly lower AUC-PR than MSP-D on this data set.

Finally, we analyze classification performance on the σ^{70} data set stemming from *E. coli*. We observe a ranking of MAP scoring best, followed by MSP-D and MSP-G, and MSP-L scoring worst with regard to Sn, where all differences are significant. While MAP yields the best Sn of 0.446, MSP-D achieves the largest PPV of 0.157 and AUC-PR of 0.510. In both cases the improvement of MSP-D over MAP and MSL-L is significant, while between MSP-D and MSP-G this is the case regarding AUC-PR and Sn.

We summarize the results on the ten data sets in tables 4.1(a) through 4.1(e). Here, we additionally include the performance measures AUC-ROC and FPR for a fixed Sn of 0.95 into the analysis. In each cell of the tables, we count the data sets for which the classifier given in the header of the column performs significantly better than the classifier specified in the header of the row. A better classification performance corresponds to greater values of Sn, PPV, AUC-PR, and AUC-ROC, whereas for FPR lower values indicate improved performance.

Table 4.1.: Summary of the comparison using five performance measures. Each sub-table shows a statistic about the number of data sets for which the learning principle specified in the header of the column yields a significant improvement of classification performance over the learning principle specified in the header of the row. For instance, MSP-D yields a significantly larger Sn than MSP-L for 8 of the 10 data sets.

(a) Sn

		greater			
		MAP	MSP-D	MSP-G	MSP-L
smaller	MAP		6	3	2
	MSP-D	1		0	0
	MSP-G	3	5		1
	MSP-L	5	8	5	

(b) PPV

		greater			
		MAP	MSP-D	MSP-G	MSP-L
smaller	MAP		7	7	2
	MSP-D	1		0	0
	MSP-G	1	0		0
	MSP-L	6	9	9	

(c) AUC-PR

		greater			
		MAP	MSP-D	MSP-G	MSP-L
smaller	MAP		8	5	1
	MSP-D	0		0	0
	MSP-G	0	4		0
	MSP-L	6	8	8	

(d) AUC-ROC

		greater			
		MAP	MSP-D	MSP-G	MSP-L
smaller	MAP		6	7	1
	MSP-D	2		2	1
	MSP-G	2	2		0
	MSP-L	5	8	9	

(e) FPR

		smaller			
		MAP	MSP-D	MSP-G	MSP-L
greater	MAP		6	5	1
	MSP-D	0		2	1
	MSP-G	1	0		0
	MSP-L	8	8	8	

Considering the first column of table 4.1(a) we find that MAP significantly outperforms MSP-D regarding Sn on one data set, MSP-G on three data sets, and MSP-L on five data sets. In turn, MSP-D yields a significantly improved Sn over MAP for six data sets, and over MSP-G and MSP-L for five and eight data sets, respectively. The table shows that regarding Sn, MSP-D achieves the best overall performance of all tested classifiers, as it performs better than any of the other classifiers for at least half of the data sets. The other extreme is MSP-L which performs significantly worse than any of the other classifiers for at least half of the data sets.

This inferiority of MSP-L becomes even more articulate for PPV and AUC-PR, where MSP-L performs worse than the other classifiers for six to nine data sets. With regard to PPV, MSP-D outperforms MAP for seven of the ten data sets, whereas the opposite holds true for only one data set. The classification performance of MAP is inferior to that of MSP-G in seven cases as well. In contrast, the PPV reached by MSP-D and MSP-G differs significantly for none of the ten data sets. Considering AUC-PR, MAP is again significantly outperformed by MSP-D as well as MSP-G in the majority of cases. For none of the data sets, MSP-D performs significantly worse than any of the tools, while MSP-G is surpassed only by MSP-D for four of the ten data sets.

The general picture remains similar with regard to AUC-ROC and FPR. Again, MSP-L is clearly inferior to the other classifiers, and MAP is more often outperformed by MSP-D and MSP-G than vice versa. However, the dominance of MSP-D over MSP-G is less pronounced

for AUC-ROC than it was for Sn and AUC-PR, as MSP-G outperforms MSP-D in two cases and in just as many cases we find the opposite. MSP-G yields a significantly lower FPR than MSP-D in two cases, whereas MSP-D does not perform significantly better than MSP-G for any of the ten data sets.

4.1.4.2. Influence of Gaussian vs. product-Dirichlet prior

MSP-G and MSP-D show an improved classification performance over MAP and MSP-L with regard to Sn, PPV, AUC-PR, AUC-ROC, and FPR. Comparing the results of MSP-G and MSP-D in table 4.1, we find a slight advantage of MSP-D which is particularly noticeable for Sn and AUC-PR. As both use the discriminative MSP principle, these differences can be attributed to the employed priors and associated hyper-parameters.

In figure 4.2, we investigate in which aspects the two priors, namely the Gaussian prior and the transformed product-Dirichlet prior, are different. To this end, we plot the density of the Gaussian prior for the chosen $\kappa_{tfbs} = 2$ and the density of the transformed Dirichlet prior with $\alpha_{tfbs} = 4$ against the value of parameter ξ_A on a logarithmic scale. The Gaussian prior assumes all parameters to be independent (see section 3.4.1), whereas the transformed Dirichlet prior explicitly models the interdependencies of parameters living on the same simplex. Thus we also consider different values of the other two free parameters, ξ_C and ξ_G, whereas the parameter ξ_T is not free but fixed to a value of 0 (see section 3.4.2).

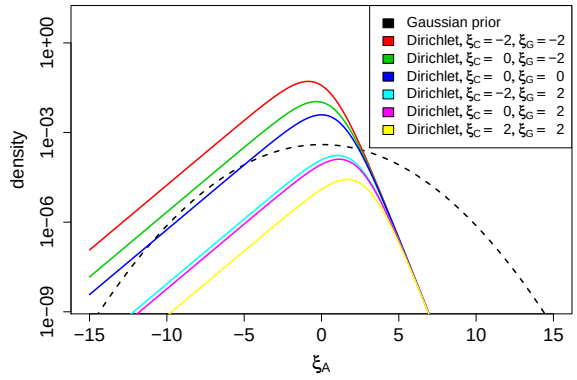


Figure 4.2.: Comparison of the density of the transformed Dirichlet prior with equivalent sample size $\alpha_{tfbs} = 4$ to the density of the Gaussian prior with $\kappa^*_{tfbs} = 2$ for parameter ξ_A. While all parameters are assumed to be independent for the Gaussian prior, the value of the density of the transformed Dirichlet prior depends on the values of the remaining free parameters ξ_C and ξ_G.

A first obvious difference, which has already been noted in section 3.4.2, is the almost linear decline of the transformed Dirichlet prior in the tails, where the Gaussian prior shows a quadratic characteristic. Although the transformed Dirichlet prior is almost quadratic in the vicinity of the maximum, this region is fairly narrow with regard to the values of ξ_A. From figure 4.2 we can also observe the effects of the interdependency between ξ_A, and ξ_C and ξ_G. If ξ_C and ξ_G are equal to 0, the maximum of the density with regard to ξ_A is located at a value of

0, which corresponds to a uniform probability distribution over all four nucleotides. If ξ_C or ξ_G deviate from 0, we find that the maximum with regard to ξ_A is shifted into the same direction, which also shifts the optimal parameters in the direction of a uniform distribution, reflecting the choice of hyper-parameters. Nevertheless, these deviations leave the characteristic of the density above $\xi_A = 5$ almost unchanged. Hence, reasons for the advantage of MSP-D over MSP-G might be the explicit modelling of interdependencies between parameters on the same simplex and the combination of a linear characteristic in the far tails, which penalizes justified deviations from the uniform distribution less than a quadratic decline, and a quadratic characteristic in the vicinity of the maximum, which is more convenient than the discontinuity of the Laplace prior.

4.1.4.3. Influence of the order for MAP and MSP-D

We might argue that the choice of hyper-parameter for the different priors greatly influences classification performance and, hence, the results of the comparison. This is definitely true for the Gaussian and the Laplace prior, i.e. MSP-G and MSP-L. Although the values of the κ_c, which determine the hyper-parameters, are carefully chosen in a pre-study, it is conceivable to find other values of κ_c that lead to an improved overall performance of MSP-G and MSP-L on these data sets. However, regarding the comparison of MSP-D to MAP we can at least state that both use equivalent priors with identical hyper-parameters and, hence, this comparison is unbiased by the influence of different priors, and the improvement of classification performance achieved by MSP-D can be attributed to the discriminative learning principle. Nonetheless, a recent study (Keilwagen et al., 2010c) gives indication that the discriminative MSP principle and the generative MAP principle react differently to an increase or decrease of ESS and each principle prefers different magnitudes of ESS. Hence, although the comparison is unbiased, it may not be "fair" from a more general perspective.

In the following, we scrutinize the different behavior of discriminative and generative learning principles with regard to the orders of the employed Markov models. Since the comparison of MAP and MSP-D is unbiased by the choice of priors, we limit the further analyses to these two approaches. Figure 4.3 presents AUC-PR of MAP and MSP-D for all tested combinations of orders on three exemplary data sets that represent well the spectrum across all data sets. Considering the results of MAP and MSP-D on the C/EBP data set, we find a general tendency to prefer higher order background models. The only exception is the combination of two PWM models for MAP, which performs notably well compared to the other combinations learned by MAP. While the order of the foreground model has only minor influence for MAP, we find a clear decrease of AUC-PR considering MSP-D for order 1 in the foreground as opposed to those combinations using a PWM model, which can possibly be attributed to over-fitting. We also find that the optimal combination of model orders, which is $(1, 4)$ for MAP and $(0, 4)$ for MSP-D, yields a significant improvement of AUC-PR compared to all other combinations of orders for both learning principles.

The latter can not be observed for the Sp1 data set. The behavior of MAP with regard to model orders appears fairly erratic on first sight, but with the exception of the combination of two PWM models, we find an increasing AUC-PR with increasing order of the background

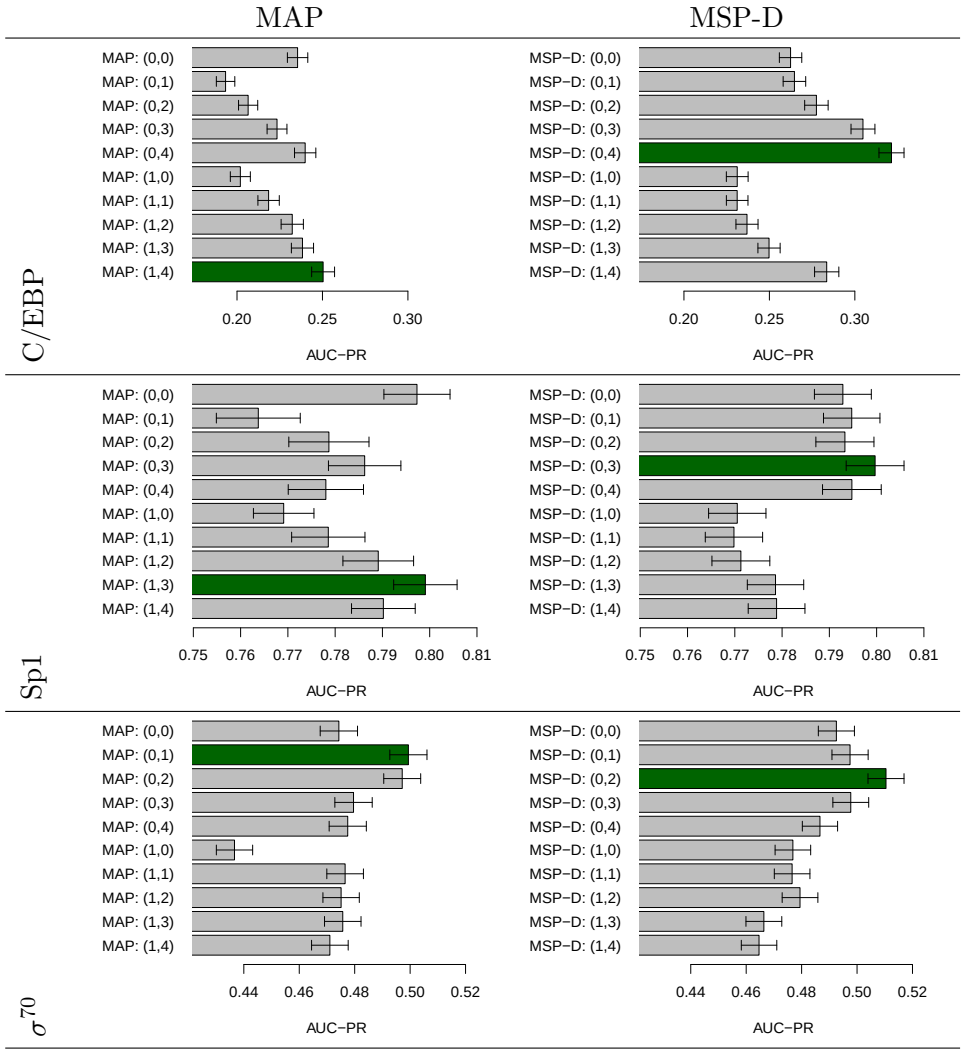


Figure 4.3.: Comparison of MAP and MSP-D for all tested combination of orders of the employed Markov models on three exemplary data sets. For each data set the bar of the combination of orders yielding the greatest AUC-PR is marked in green.

model up to order 3, whereas AUC-PR decreases for order 4. For MAP, we also find a tendency to prefer a WAM model in the foreground instead of a PWM model. However, the optimal combination of orders $(1, 3)$ does not improve classification performance significantly as compared to $(0, 0)$. Again, we observe a clear preference of MSP-D for a PWM model in the foreground. However, a clear distinction between different orders of the background model cannot be made and we observe only a slight improvement for higher order background models, if we use a first order model in the foreground.

On the σ^{70} data set, the clear preference of MSP-D with regard to the foreground order is superimposed by the predominance of lower order background models. The best combination of orders $(0, 2)$ using MSP-D is significantly superior to all other combinations on this data set, whereas for MAP the combinations of a PWM model in the foreground and a first or second order Markov model in the background achieve comparable values of AUC-PR.

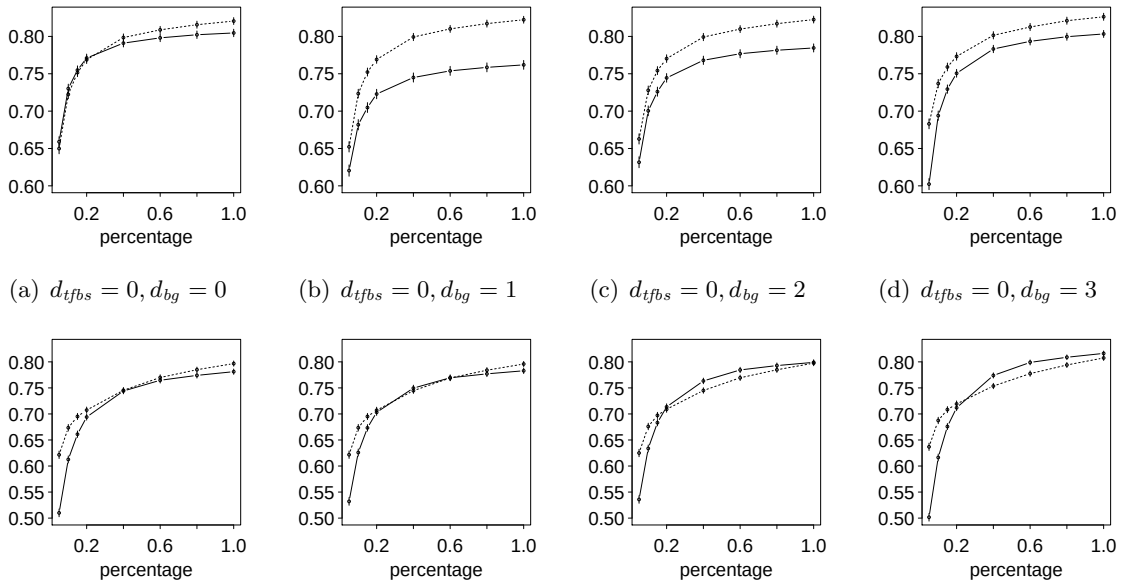
Summarizing the results, we do not find a congruent preference regarding model orders for MAP and MSP-D. For MSP-D, we observe the tendency that a PWM model in the foreground

combined with a higher order Markov model in the background achieves the best AUC-PR. In contrast, MAP shows no consistent preference of model orders. However, we do find that a combination of two PWM models often performs surprisingly well. One explanation for the low performance of WAM models in the foreground for MSP-D could be over-fitting. For the chosen hyper-parameters, we use the same hyper-parameters for the parameters of a fourth order Markov model in the background ($\frac{1024}{4^5} = 1$) as for a PWM model in the foreground ($\frac{4}{4^1} = 1$), whereas the hyper-parameters for the WAM model are $\frac{4}{4^2} = 0.25$. However, it is not obvious, why this does affect MSP-D but not MAP, which uses the very same hyper-parameters.

4.1.4.4. Classification performance on small data sets

Finally, we investigate on the Sp1 data set, how the classification performance of MAP and MSP-D depends on the amount of training data. To this end, we extend the stratified holdout sampling by an additional sub-sampling on the partitions used for training to artificially reduce the size of the training data set. Since the background data set contains a number of long sequences and we only sample complete sequences (see section 3.5.2, p. 41), we cut the background data set into chunks of 100 bp beforehand. Conducting an independent sub-sampling on the foreground and background data set, we adhere to a stratified schema and keep the proportions of the two classes constant. We present the results with regard to AUC-PR for sub-sampled training data sets of 5% to 100% of the original size in figure 4.4. The evaluation for the other considered performance measures gives similar results (data not shown). Besides the different treatment of the background data, the results for 100% of the training data correspond to those presented in figure 4.3. Considering the first row of figure 4.4, we find that MSP-D (dashed line) performs equally or better than MAP (solid line) for all relative sizes of the training data. The improvement of MSP-D over MAP is especially noticeable for $d_{bg} = 1$ and $d_{bg} = 2$, which can already be anticipated from figure 4.3.

The picture is less clear for $d_{tfbs} = 1$ as shown in the second row of figure 4.4. For $d_{bg} = 2$ and $d_{bg} = 1$, MAP even achieves a greater AUC-PR than MSP-D for relative sizes of the training data of 40% and above. Interestingly, MSP-D still performs better than MAP for the smallest training data sets. Hence, we may state that although MSP-D yields a lower maximum AUC-PR for the largest training data sets, it approaches this maximum faster than MAP with increasing relative size of the training data. This observation is not in accordance with the findings of Ng and Jordan (2002), who compare the classification performance of MAP-trained naïve Bayes classifiers and logistic regression for varying sizes of the training data on data sets from the UCI machine learning repository. Ng and Jordan (2002) find that although logistic regression has a lower asymptotic generalization error, i.e. a higher classification performance on independent test data using large training data sets, the naïve Bayes classifier may converge faster to its asymptotic performance. For discrete data, the MAP-trained naïve Bayes classifier corresponds to MAP using two PWM models (cf. figure 4.4(a)), whereas logistic regression is equivalent to a classifier using two PWM models trained by the discriminative MCL principle (see sections 3.2.1 and 3.3.1). The main difference – besides the specific data sets – between the study of Ng and Jordan (2002) and the evaluations presented here is that we use the MSP learning principle instead of MCL in the discriminative setting. Since MAP and MSP-D use



(a) $d_{tfbs} = 0, d_{bg} = 0$ (b) $d_{tfbs} = 0, d_{bg} = 1$ (c) $d_{tfbs} = 0, d_{bg} = 2$ (d) $d_{tfbs} = 0, d_{bg} = 3$

(e) $d_{tfbs} = 1, d_{bg} = 0$ (f) $d_{tfbs} = 1, d_{bg} = 1$ (g) $d_{tfbs} = 1, d_{bg} = 2$ (h) $d_{tfbs} = 1, d_{bg} = 3$

Figure 4.4.: Comparison of MAP (solid line) and MSP-D (dashed line) for different sizes of the training data on the Sp1 data set. We plot AUC-PR against the percentage of sub-sampled data that are used for the training.

equivalent priors with identical hyper-parameters in the study presented here, we may state that the differences we find can be attributed to the different learning principles alone.

4.1.5. Conclusions

We compare the classification performance of Markov models trained by the generative MAP principle and the discriminative MSP principle for the prediction of TFBSs on ten data sets stemming from mammals, *A. thaliana*, and *E. coli*. We find that the MSP principle with Gaussian and transformed product-Dirichlet priors outperforms the MAP principle in the majority of cases. As MSP with a transformed product-Dirichlet prior and MAP use equivalent priors with identical hyper-parameters, this improvement in classification performance can be attributed to the Bayesian discriminative MSP learning principle alone. Our results give indication that the discriminative learning of parameters by MSP might be beneficial for other problems in bioinformatics as well, some of which are scrutinized in the remainder of this work.

4.2. De-novo discovery of cis-regulatory modules

In the last section, we considered the recognition of known transcription factor binding sites. However, often we know neither the exact binding position nor the binding motif of the transcription factor of interest. Instead, we are faced with a number of approximate binding regions or a selection of putative promoters of potentially co-regulated genes. These are determined by wet-lab techniques, which we shortly describe in the next sub-section. De-novo motif discovery aims at inferring the binding motif and corresponding binding sites from the given sequences. Here, we propose a novel approach for de-novo motif discovery that employs the discriminative MSP principle to search for differentially abundant motifs and utilizes the positional preference of binding sites, and we extend this approach to the discovery of cis-regulatory modules comprising binding sites of two different transcription factors.

4.2.1. Wet-lab techniques

DNase footprinting (Galas and Schmitz, 1978) uses Deoxyribonuclease I (DNase I) to digest DNA that is not bound in proteins, e.g. transcription factors. The remaining fragments of undigested DNA are then sequenced by the Maxam-Gilbert method. For obtaining information about binding of a specific transcription factor, this factor must be available in purified form for the incubation of DNA. DNase footprinting is able to identify binding sites with an accuracy of up to 1 bp. However, distinct runs of gel-electrophoresis are necessary for each considered sequence and, hence, DNase footprinting is time-consuming and only suitable for low-throughput experiments.

EMSA (electrophoretic mobility shift assay) (Fried and Crothers, 1981; Hellman and Fried, 2007) utilizes that protein-DNA complexes migrate more slowly than free DNA in electrophoresis and bands of labeled DNA are shifted on the gel, depending on the binding of the studied protein to DNA. By adding different concentrations of a transcription factor to the DNA solution, the binding affinity of transcription factors can be studied, and binding affinities of different factors can be compared. However, EMSA does not elucidate exact binding sites of the transcription factor of interest, but is only capable of measuring the general binding affinity to a (longer) DNA sequence.

For ELISA (enzyme-linked immunosorbent assay) (Benotmane et al., 1997; Mönke et al., 2004), short DNA sequences are immobilized and incubated with the transcription factor of interest. After a washing step, the bound transcription factor is detected using a specific antibody. Since for many transcription factors specific antibodies do not exist, often recombinant variants are employed that exhibit artificial tags at the N- or C-terminus of the transcription factor. These tags can be detected by generic antibodies.

Antibodies are also used for chromatin immunoprecipitation combined with microarrays (ChIP-on-chip or ChIP-chip) (Sun et al., 2003; Wu et al., 2006). For ChIP-chip, DNA is cross-linked to the transcription factor of interest and the free DNA is either digested by nuclease or sheared by sonication. The complexes of DNA and transcription factor are then extracted using antibodies and the linked transcription factor is removed. For detecting the resulting DNA fragments on the microarray, these are labeled by a fluorescent tag. The labeled DNA

is hybridized to the probes of a microarray and the binding to probes is detected as a fluorescence image. This image is further processed resulting in a list of intensities for each of the probes. Similar to microarrays measuring mRNA abundance, these intensities must be normalized and statistical tests or more complex models must be employed to finally decide if the DNA corresponding to a specific probe is bound by the factor of interest. Hence, the annotation of binding regions depends on the computational methods used for the latter step. If genome-wide tiling arrays exist for the studied organism, ChIP-chip can be used to obtain a genome-wide map of binding sites for the factor of interest. However, the spotted probes of the applied tiling arrays often comprise several hundred basepairs (Sun et al., 2003), which limits the resolution of this technique.

The first steps of ChIP-chip, i.e. the immunoprecipitation, are also applied for chromatin immunoprecipitation combined with parallel sequencing (ChIP-seq) (Johnson et al., 2007). However, instead of microarrays, high-throughput parallel sequencing is used to analyze the extracted sequences. The resulting sequence reads are then mapped to the genome, and regions covered by a minimum number of reads and considerably enriched with reads compared to a control experiment are annotated as binding sites of the factor of interest (Johnson et al., 2007). The resolution of ChIP-seq depends on the employed sequencing technique and the length of the DNA fragments selected for sequencing, but is typically considerably higher than using ChIP-chip confining the putative binding sites to tens or a few hundred of basepairs.

The latter techniques all depend on the availability of either a specific antibody for the transcription factor of interest or a tagged, recombinant variant of this factor. Often neither does exists and we must resort to other, more indirect techniques for genome-scale analysis of gene regulation. One such technique is gene expression profiling by microarrays (Lockhart and Winzeler, 2000). Current microarrays can measure mRNA levels of thousands of genes of an organism, e.g. the ATH1 chip for *A. thaliana* contains probes for more than 22500 genes. Microarrays can be used to identify genes that are differentially expressed under certain conditions, e.g. external stress factors, hormone supply, or different stages of development. Assuming that co-expression is related to co-regulation, we expect specific cis-regulatory elements or modules to be enriched in the promoter regions of these genes compared to the promoter regions of the remaining genes.

Given a set of promoter regions, de-novo motif discovery approaches can be used to identify these elements. De-novo motif discovery is also used to infer the binding motif and exact binding sites in the approximate regions determined by ChIP-chip and, depending on the achieved resolution, ChIP-seq experiments.

4.2.2. Related work

One of the first approaches for de-novo motif discovery is proposed by Lawrence and Reilly (1990), who use an OOPS (one occurrence per sequence) model which is learned by the generative ML principle using the expectation-maximization (EM) algorithm. The OOPS model assumes that each sequence in the data set contains exactly one occurrence of the motif of interest. Lawrence and Reilly (1990) use a homogeneous Markov model of order 0 to model the flanking parts of the sequence, i.e. those positions that are not covered by a motif occurrence.

As a motif model, they employ PWM models and more specific models for e.g. palindromic motifs. The EM algorithm is also employed by Bailey and Elkan (1994) in the MEME algorithm for de-novo motif discovery. In contrast to Lawrence and Reilly (1990), Bailey and Elkan (1994) use a two-component mixture model, where the one component models motif occurrences by a PWM model and the other component uses a homogeneous Markov model of order 0 for modeling a background distribution of nucleotides. For learning this model, the input sequences are cut into overlapping sub-sequences of length w, where w denotes the length of the motif to be found. Due to overlapping, the MEME algorithm has the tendency to converge to repetitive sequences, which Bailey and Elkan (1994) compensate for by an additional normalization step. In (Bailey and Elkan, 1995), MEME is extended by an option to choose between the original two-component mixture model and the OOPS model. MEME is still one of the most frequently used programs for de-novo motif discovery, especially via the MEME web server (Bailey et al., 2006).

In (Lawrence et al., 1993), the EM algorithm for learning the OOPS model is replaced by Gibbs sampling. Additionally, the model is extended to multiple different motifs, which is simplified by using sampling instead of an EM algorithm considering all combinations of motif occurrences. Lawrence et al. (1993) also include heuristics to compensate for phase shifts and to automatically adapt the motif length w. The generative MAP principle is used for estimating the parameters of the models. The Gibbs sampler is extended to multiple instances for each of the considered motifs in (Thompson et al., 2003). While the former two Gibbs samplers report the optimal MAP solution after convergence, the centroid Gibbs sampler (Thompson et al., 2007) considers for each sequence the sampled solutions, i.e. the number of sites, the motifs describing these sites and the positions of occurrence, over a large number of iterations after an initial burn-in phase. The reported solution is then the centroid of all sampled solutions, i.e. the solution with the minimum pair-wise distance to all sampled solutions.

Improbizer (Ao et al., 2004) employs a combination of a PWM model and a homogeneous Markov model as well, which are learned by a heuristic procedure similar to the EM algorithm, using 6-mers found in the data to initialize the PWM model. For learning the background distribution represented by the homogeneous Markov model, Improbizer can utilize an additional, typically large, background data set to compensate for organism-specific properties like G/C-content. Optionally, Improbizer may learn a Gaussian position distribution of motif occurrences, which is also used to predict motif occurrences.

Positional information is also utilized by Kim et al. (2008) in conjunction with Gibbs sampling. The proposed algorithm A-GLAM models the position distribution by a Gaussian density similar to Improbizer. Kim et al. (2008) choose a PWM model as the motif model, while the background or flanking sequences are modeled by a third order homogeneous Markov model. Since A-GLAM uses a Bayesian approach, it applies the common choice of a product-Dirichlet prior to the parameters of the sequence models. As a prior for the parameters of the position distribution, (Kim et al., 2008) use the product of a uniform density for the mean parameter and a gamma density for the precision.

In contrast to the previous approaches, Weeder (Pavesi et al., 2001) does not employ a statistical model representing the sequence motif, but searches for over-represented patterns in the sequences of interest. These patterns correspond to common sub-strings that occur in most of

the sequences of interest with a user-specified maximum number of mismatches. The search for common sub-strings with mismatches is accomplished efficiently by generalized suffix trees.

All the former approaches have in common that they search for motifs or patterns that are over-represented in a target data set comprising the sequences of interest. However, such motifs frequently appear over-represented in the entire genome or in all promoter regions and are not specific for the target data set. To overcome this problem, discriminative approaches have been proposed that search for differentially abundant motifs in a set of target sequences compared to a control data set. The control data set may either be chosen specifically, for instance as promoters of genes that are not differentially expressed in experiments, or sampled from all promoter sequences except those in the target data set.

The discriminative matrix enumerator (DME, Smith et al. (2005)) searches for PWM models that are over-represented in the target data set relative to a control data set. To this end, DME enumerates weight matrices that are reasonably different from a uniform distribution, which excludes very degenerate motifs. DME chooses the PWM model that maximizes the log likelihood ratio between the likelihood of all w-mers in the target data set and the likelihood of all w-mers in the control data set. Since the number of matrices that are initially enumerated is limited due to computation time, the chosen matrix is further refined by a local search among slightly deviating matrices in a chosen neighborhood.

Redhead and Bailey (2007) propose discriminatively enhanced motif elicitation (DEME) for motif discovery in protein and DNA sequences. DEME optimizes a ZOOPS (zero or one occurrence per sequence) model with respect to conditional likelihood. However, the frequencies observed from the data are augmented by pseudo counts, resulting in an objective function similar to the supervised posterior utilizing a product-Dirichlet prior. DEME uses a PWM model for representing the motif and a homogeneous Markov model of order 0 for representing flanking sequences. Another homogeneous Markov model of order 0 is used for modeling all sequences in the control data set.

Although the previous approaches can detect binding sites of multiple motifs, typically by removing all occurrences of the first motif from the data and restarting the algorithm, they do not model cis-regulatory modules explicitly. Cis-regulatory modules comprise multiple binding sites of identical or different transcription factors, which bind coordinately to have a regulatory effect. This coordinate binding may entail a specific order of binding sites or preferred distances between the sites.

In a recent review, Loo and Marynen (2009) assign approaches for the detection of cis-regulatory modules (CRMs) into three classes: CRM scanners that search for occurrences of a pre-defined cis-regulatory module for instance in the entire genome, CRM screeners that use a data base of pre-defined PWM models representing the binding sites of different factors and search for over-represented combinations of such binding sites, and CRM builders that learn new PWM models forming a CRM model given promoters of potentially co-regulated genes. The latter class is essentially an extension of the above mentioned approaches for de-novo motif discovery to multiple factors and binding sites. In the following, we consider only CRM builders.

CoBind (GuhaThakurta and Stormo, 2001) models two cooperatively binding transcription factors by two PWM models. Binding sites according to these PWM models may occur in a sequence as non-overlapping sub-sequences. The objective function optimized by CoBind is the log-likelihood of the two PWM models given start positions of motif occurrences, which is normalized to the likelihood of the sequences in a control data set. The optimization of parameters with respect to this objective function is accomplished by a combination of sampling and optimization: start positions of motif occurrences in the sequences of the target data set are drawn similar to Gibbs sampling, and given these start positions, the parameters of the two PWM models are optimized by gradient descent. After a fixed number of iterations, CoBind reports the PWM models that achieved the maximum objective function and corresponding binding sites.

CisModule (Zhou and Wong, 2004) models each sequence by a hierarchical mixture model. At each position of a sequence, this model may open a new CRM with a certain probability or, with the complementary probability, proceed with flanking sequence, which is modeled by a homogeneous Markov model of order 1. To reduce computational complexity, the length of CRMs is fixed. At each position within a CRM, a similar decision is made to either start a binding site of a transcription factor, which is represented by a PWM model, or to consider this position as flanking. After closing a CRM, CisModule may either start the next CRM or proceed with flanking sequence. This process is repeated until the end of the current sequence is reached and CisModule proceeds with the next sequence. The parameters of CisModule are optimized by Gibbs sampling, and positions that are located within a sampled CRM in at least 50% of the sampling iterations are finally predicted as CRM occurrences.

EMCModule (Gupta and Liu, 2005) uses a combination of PWM models for binding sites within CRMs and homogeneous Markov models for flanking sequences. In contrast to the previous approaches, EMCModule initially requires a selection of candidate PWM models, which can be obtained either from databases like Transfac® or Jaspar, or by single motif discovery approaches. For optimizing the selection of PWM models, EMCModule employs a method called evolutionary Monte Carlo, which combines the evolutionary selection of PWM models with sampling, whereas the parameters of the PWM are learned by the MAP principle. EMCModule finally reports the combination of PWM models that yields the maximum posterior and corresponding CRMs.

Valen et al. (2009) propose a discriminative approach called Motif Annealer (MoAn) for the de-novo discovery of CRMs. MoAn considers binding sites of at most two transcription factors, which are represented by PWM models, whereas flanking sequences are modeled by a homogeneous Markov model of order 0. Similar to the approach proposed in this work, MoAn explicitly models the co-occurrence of both motifs, the occurrence of each single motif, and the occurrence of none of the motifs in each of the sequences, resulting in a mixture model over these possibilities. Valen et al. (2009) optimize the parameters of MoAn with respect to conditional likelihood and to this end employ a simulated annealing approach to escape local maxima.

Table 4.2 summarizes the above approaches and classifies these according to the employed learning principle, i.e. either generative or discriminative, and the capability of learning a position distribution of binding sites from the data. Approaches that explicitly model CRMs are

Table 4.2.: Classification of approaches for de-novo motif discovery according to the employed learning principle and the capability to learn a position distribution from the data. Approaches marked with an asterisk are specifically designed for the de-novo discovery of cis-regulatory modules.

		Position distribution	
		fixed	learned from data
Learning principle	generative	Gibbs Sampler MEME Weeder CoBind* CisModule* EMCModule*	Improbizer A-GLAM
	discriminative	DME DEME MoAn*	

marked with an asterisk. Most of the existing approaches use a generative learning principle, i.e. the ML or MAP principle, or a pattern-based approach in case of Weeder, and do not learn a position distribution. For all approaches considered here, the fixed position distribution is a uniform distribution over all admissible start positions. To the best of our knowledge, no approach exists that uses a discriminative approach for discovering differentially abundant motifs in conjunction with learning the position distribution from the data. Here, we propose such an approach that is also capable of modeling cis-regulatory modules comprising at most two different motifs.

4.2.3. Model

This section is structured as follows. We introduce the ZOOPS model including a position distribution that allows for the de-novo discovery of single motifs in section 4.2.3.1, and we extend this model to cis-regulatory modules comprising two different motifs in section 4.2.3.2. We define the position distributions used for these two models in section 4.2.3.3. In section 4.2.3.4, we define priors for the sequence models and the position distribution. Finally, we describe a heuristic employed for an automatic adaption of motif length and the compensation for phase shifts in section 4.2.3.5 and we explain how we use the learned model to predict motif occurrences in a set of given sequences in section 4.2.3.6.

We learn the parameters of the employed models by the discriminative MSP principle (see section 3.2.2, p. 13). To this end, we need the likelihood of sequence $\boldsymbol{x}$ and class c, where $c \in \{target, control\}$. Here, *target* denotes the class of target sequences and *control* denotes the class of sequences stemming from the control data set. We decompose this likelihood into the a-priori probability of class c given parameters β_{target} and $\beta_{control}$, and the probability of

sequence $\boldsymbol{x}$ given class c and class-dependent parameters $\boldsymbol{\beta}_c$, yielding

$$P(\boldsymbol{x}, c|\boldsymbol{\beta}) = P(c|\beta_{target}, \beta_{control})P(\boldsymbol{x}|c, \boldsymbol{\beta}_c), \tag{4.1}$$

where $\boldsymbol{\beta} = (\beta_{target}, \beta_{control}, \boldsymbol{\beta}_{target}, \boldsymbol{\beta}_{control})$. We parameterize the a-priori class probabilities in terms of real-valued parameters β_c as

$$P(c|\beta_{target}, \beta_{control}) = \frac{\exp(\beta_c)}{\sum_{\tilde{c}} \exp(\beta_{\tilde{c}})} \tag{4.2}$$

to allow for unconstrained numerical optimization (cf. sections 3.2.1, p. 12).

4.2.3.1. ZOOPS model with position distribution

We start the derivation of the ZOOPS model employed for the de-novo discovery of single motifs with the model for sequences that do not contain a binding site of the considered motif. This model is also used for the class of control sequences that are assumed to contain no binding sites. Similar to most of the other approaches, we model such sequences by a homogeneous Markov model of order 0. Accordingly, we define the probability $P_0(\boldsymbol{x}|c, \boldsymbol{\beta}_c)$ of sequence $\boldsymbol{x}$ given class c and parameters $\boldsymbol{\beta}_c$ as

$$P_0(\boldsymbol{x}|c, \boldsymbol{\beta}_c) = P_{hMM(0)}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,hMM}), \tag{4.3}$$

where the homogeneous Markov model is parameterized in terms of real-valued parameters as defined in section 3.3.1 and $\boldsymbol{\beta}_{c,hMM}$ denotes the subset of parameters in $\boldsymbol{\beta}_c$ that are used for this Markov model.

To derive the model for those sequences stemming from class *target* that contain a binding site, we first assume that we know the position ℓ at which this site starts. We define the joint probability $P_m(\boldsymbol{x}, \ell|c, \boldsymbol{\beta}_c)$ of sequence $\boldsymbol{x}$ and start position ℓ given class c and parameters $\boldsymbol{\beta}_c$ as the product of

- the probability of position ℓ according to the position distribution $P_{pos}(\ell|c, \boldsymbol{\beta}_{c,pos})$ with parameters $\boldsymbol{\beta}_{c,pos}$,
- the probability of the nucleotides occurring in the flanking sequence before the binding site, which are modelled by the same homogeneous Markov model with parameters $\boldsymbol{\beta}_{c,hMM}$ that is also used in equation (4.3),
- the probability of the nucleotides within the binding site, i.e. $x_\ell, \ldots, x_{\ell+w-1}$ and w denoting the length of the motif, which are represented by a strand model enclosing a PWM model with parameters $\boldsymbol{\beta}_{c,m}$, and
- the probability of the nucleotides in the flanking sequence after the binding site according to the homogeneous Markov model.

We define

$$\begin{aligned} P_m(\boldsymbol{x}, \ell|c, \boldsymbol{\beta}_c) = P_{pos}(\ell|c, \boldsymbol{\beta}_{c,pos})P_{hMM(0)}(x_1, \ldots, x_{\ell-1}|c, \boldsymbol{\beta}_{c,hMM}) \cdot \\ P_m^S(x_\ell, \ldots, x_{\ell+w-1}|c, \boldsymbol{\beta}_{c,m})P_{hMM(0)}(x_{\ell+w}, \ldots, x_L|c, \boldsymbol{\beta}_{c,hMM}), \end{aligned} \tag{4.4}$$

which we may alternatively state as

$$P_m(\boldsymbol{x}, \ell | c, \boldsymbol{\beta}_c) = P_{hMM(0)}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,hMM}) P_{pos}(\ell|c, \boldsymbol{\beta}_{c,pos}) \frac{P_m^S(x_\ell, \ldots, x_{\ell+w-1}|c, \boldsymbol{\beta}_{c,m})}{P_{hMM(0)}(x_\ell, \ldots, x_{\ell+w-1}|c, \boldsymbol{\beta}_{c,hMM})}, \tag{4.5}$$

where $\boldsymbol{\beta}_c$ includes $\boldsymbol{\beta}_{c,pos}$, $\boldsymbol{\beta}_{c,hMM}$, and $\boldsymbol{\beta}_{c,m}$.

The strand model is defined as

$$P_m^S(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,m}) = P(\text{fw}|\boldsymbol{\beta}_{c,m}) P_{\text{PWM}}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,m}) + P(\text{bw}|\boldsymbol{\beta}_{c,m}) P_{\text{PWM}}(\boldsymbol{x}^{rc}|c, \boldsymbol{\beta}_{c,m}), \tag{4.6}$$

where $\boldsymbol{x}^{rc}$ denotes the reverse complement of $\boldsymbol{x}$, and the a-priori probabilities of the forward strand $P(\text{fw}|\boldsymbol{\beta}_{c,m})$ and backward strand $P(\text{bw}|\boldsymbol{\beta}_{c,m})$ are parameterized in analogy to equation (4.2) in terms of real-valued parameters $\beta_{\text{fw}|c,m}$ and $\beta_{\text{bw}|c,m}$. The PWM model $P_{\text{PWM}}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,m})$ is parameterized as defined in section 3.3.1.

Actually, we do not know the start position ℓ of the binding site. Rather this is one information we want to determine by de-novo motif discovery. Hence, we regard the random variable emitting the start positions as a *hidden* variable and determine the probability of sequence $\boldsymbol{x}$ given class c and parameters $\boldsymbol{\beta}_c$ as the marginal probability over all admissible start positions ℓ, i.e.

$$P_m(\boldsymbol{x}|c, \boldsymbol{\beta}_c) = \sum_{\ell=1}^{L-w+1} P_m(\boldsymbol{x}, \ell|c, \boldsymbol{\beta}_c). \tag{4.7}$$

We finally define the probability $P_{\text{ZOOPS}}(\boldsymbol{x}|c, \boldsymbol{\beta}_c)$ of sequence $\boldsymbol{x}$ given the ZOOPS model as a mixture of $P_0(\boldsymbol{x}|c, \boldsymbol{\beta}_c)$ and $P_m(\boldsymbol{x}|c, \boldsymbol{\beta}_c)$

$$P_{\text{ZOOPS}}(\boldsymbol{x}|c, \boldsymbol{\beta}_c) = P(u = 0|c, \boldsymbol{\beta}_c) P_0(\boldsymbol{x}|c, \boldsymbol{\beta}_c) + P(u = 1|c, \boldsymbol{\beta}) P_m(\boldsymbol{x}|c, \boldsymbol{\beta}_c), \tag{4.8}$$

where $P(u = 0|c, \boldsymbol{\beta}_c)$ denotes the a-priori probability that a sequence does not contain a binding site and $P(u = 1|c, \boldsymbol{\beta})$ denotes the complementary a-priori probability. These a-priori probabilities are parameterized in analogy to equation (4.2) in terms of real-valued parameters $\boldsymbol{\beta}_{0|c}$ and $\boldsymbol{\beta}_{1|c}$.

4.2.3.2. Multiple motif model with position distribution

The ZOOPS model can be readily extended to multiple motifs. We assume that we observe a binding site of the first motif m_1 at position ℓ_1 and a binding site of the second motif m_2 at position ℓ_2 of sequence $\boldsymbol{x}$. The numbering of motifs does not imply an order of occurrence and the two motifs may occur at arbitrary positions ℓ_1 and ℓ_2 as long as the two motifs do not overlap. We define the joint probability of sequence $\boldsymbol{x}$ and these two positions in analogy

to equation (4.5) as

$$P_{m_1,m_2}(\boldsymbol{x}, \ell_1, \ell_2 | c, \boldsymbol{\beta}_c) = P_{hMM(0)}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,hMM}) P_{pos}(\ell_1, \ell_2 | c, \boldsymbol{\beta}_{c,pos}) \frac{P^S_{m_1}(x_{\ell_1}, \ldots, x_{\ell_1+w_1-1}|c, \boldsymbol{\beta}_{c,m_1})}{P_{hMM(0)}(x_{\ell_1}, \ldots, x_{\ell_1+w_1-1}|c, \boldsymbol{\beta}_{c,hMM})} \frac{P^S_{m_2}(x_{\ell_2}, \ldots, x_{\ell_2+w_2-1}|c, \boldsymbol{\beta}_{c,m_2})}{P_{hMM(0)}(x_{\ell_2}, \ldots, x_{\ell_2+w_2-1}|c, \boldsymbol{\beta}_{c,hMM})}, \tag{4.9}$$

where $P_{pos}(\ell_1, \ell_2|c, \boldsymbol{\beta}_{c,pos})$ denotes the joint position distribution of ℓ_1 and ℓ_2 given class c and parameters $\boldsymbol{\beta}_{c,pos}$, and $P^S_{m_1}(x_{\ell_1}, \ldots, x_{\ell_1+w_1-1}|c, \boldsymbol{\beta}_{c,m_1})$ and $P^S_{m_2}(x_{\ell_2}, \ldots, x_{\ell_2+w_2-1}|c, \boldsymbol{\beta}_{c,m_2})$ denote the probabilities of the binding sites given the strand models for motif m_1 and m_2, respectively.

Again, we do not know the start position ℓ_1 and ℓ_2 and, hence, marginalize over all admissible start positions, yielding the marginal probability of sequence $\boldsymbol{x}$ given class c and parameters $\boldsymbol{\beta}_c$

$$P_{m1,m2}(\boldsymbol{x}|c, \boldsymbol{\beta}_c) = \sum_{\ell_1=1}^{L-w_1+1} \sum_{\ell_2=1}^{L-w_2+1} P_{m1,m2}(\boldsymbol{x}, \ell_1, \ell_2 | c, \boldsymbol{\beta}_c). \tag{4.10}$$

Here, we do not take into account, that the binding sites of the two motifs are not allowed to overlap. We deal with this problem by assigning positions ℓ_1, ℓ_2 that would result in overlapping binding sites a probability of zero via the position distribution.

In analogy to the ZOOPS model, we define the probability of sequence $\boldsymbol{x}$ given the *multiple motif* (MuMo) model as a mixture model over the different combinatorial possibilities of motif presence and absence, i.e.

$$\begin{aligned} P_{\text{MuMo}}(\boldsymbol{x}|c, \boldsymbol{\beta}_c) = {} & P(u_1 = 1, u_2 = 1|c, \boldsymbol{\beta}_c) P_{m1,m2}(\boldsymbol{x}|c, \boldsymbol{\beta}_c) + \\ & P(u_1 = 1, u_2 = 0|c, \boldsymbol{\beta}_c) P_{m_1}(\boldsymbol{x}|c, \boldsymbol{\beta}_c) + \\ & P(u_1 = 0, u_2 = 1|c, \boldsymbol{\beta}_c) P_{m_2}(\boldsymbol{x}|c, \boldsymbol{\beta}_c) + \\ & P(u_1 = 0, u_2 = 0|c, \boldsymbol{\beta}_c) P_0(\boldsymbol{x}|c, \boldsymbol{\beta}_c), \end{aligned} \tag{4.11}$$

where $P(u_1 = 1, u_2 = 1|c, \boldsymbol{\beta}_c)$ denotes the a-priori probability that a sequence contains binding sites for both motifs, $P(u_1 = 1, u_2 = 0|c, \boldsymbol{\beta}_c)$ and $P(u_1 = 0, u_2 = 1|c, \boldsymbol{\beta}_c)$ denote the a-priori probabilities of single motif occurrences, and $P(u_1 = 0, u_2 = 0|c, \boldsymbol{\beta}_c)$ denotes the a-priori probability that none of the motifs occurs in a sequence. Again, these a-priori probabilities are parameterized in terms of real-valued parameters $\boldsymbol{\beta}_{1,1|c}$, $\boldsymbol{\beta}_{1,0|c}$, $\boldsymbol{\beta}_{0,1|c}$, and $\boldsymbol{\beta}_{0,0|c}$.

Since the model representing motif co-occurrence conceptually contains the same motifs that are also used in the models containing a single motif, we use the same strand models with identical parameters for the first motif in $P_{m1,m2}(\boldsymbol{x}|c, \boldsymbol{\beta}_c)$ and $P_{m_1}(\boldsymbol{x}|c, \boldsymbol{\beta}_c)$, and for the second motif in $P_{m1,m2}(\boldsymbol{x}|c, \boldsymbol{\beta}_c)$ and $P_{m_2}(\boldsymbol{x}|c, \boldsymbol{\beta}_c)$, respectively.

In theory, the extension of this approach to more than 2 motifs is straightforward. However, the runtime of each iteration of the numerical optimization increases exponentially in the number of considered motifs if we use a position distribution modeling correlations between

motif occurrences, since we must explicitly compute the sum over admissible combinations of positions. For this reason, numerical optimization becomes practically infeasible for more than 2 motifs using input sequence of several hundred basepairs.

4.2.3.3. Position distributions

The binding sites of many transcription factors occur at preferred distances from the transcription start site (TSS). The distribution of these positions may be modelled well by a Gaussian density (Kim et al., 2008). However, seldom binding sites may also occur in larger distance of the preferred position and some factors also exhibit a very broad distribution of binding sites. Hence, we decide for a position distribution that is a mixture of a Gaussian density and a uniform distribution. Since the Gaussian density models continuous values, but we consider only discrete positions, we re-normalize the Gaussian density by the sum over all admissible positions. We define the probability of position ℓ given class c and parameters $\boldsymbol{\beta}_{c,pos}$ as

$$P_{pos}(\ell|c,\boldsymbol{\beta}_{c,pos}) = P(\mathcal{N}|\boldsymbol{\beta}_{c,pos})\frac{1}{Z_1(\boldsymbol{\beta}_{c,pos})}\mathcal{N}(\ell|\nu_{c,pos},\kappa_{c,pos}) + P(\mathcal{U}|\boldsymbol{\beta}_{c,pos})\frac{1}{L-w+1}, \quad (4.12)$$

where $Z_1(\boldsymbol{\beta}_{c,pos})$ denotes the normalization constant over all admissible position defined as

$$Z_1(\boldsymbol{\beta}_{c,pos}) = \sum_{\ell=1}^{L-w+1} \mathcal{N}(\ell|\nu_{c,pos},\kappa_{c,pos}). \quad (4.13)$$

We parameterize the Gaussian density in terms of real-valued parameters $\nu_{c,pos}$ and $\kappa_{c,pos}$ as defined in section 3.3.2.1 (p. 25) and we parameterize the mixture probabilities as

$$P(\mathcal{N}|\boldsymbol{\beta}_{c,pos}) = \frac{\exp(\beta_{\mathcal{N}|c,pos})}{\exp(\beta_{\mathcal{N}|c,pos}) + \exp(\beta_{\mathcal{U}|c,pos})} \text{ and } \quad P(\mathcal{U}|\boldsymbol{\beta}_{c,pos}) = \frac{\exp(\beta_{\mathcal{U}|c,pos})}{\exp(\beta_{\mathcal{N}|c,pos}) + \exp(\beta_{\mathcal{U}|c,pos})}.$$

We define the bivariate position distribution of positions ℓ_1 and ℓ_2 in analogy to the univariate case as a mixture of a bivariate Gaussian density (see section 3.3.2.1, p. 25) and a uniform distribution over the admissible positions. We forbid combinations of positions that would lead to overlapping binding sites by explicitly setting the corresponding probabilities to zero. This is formalized by a Kronecker δ, which is equal to 1 if the intersection of the two intervals comprising the positions within the binding site is equal to the empty set and 0 otherwise. We define the probability of positions ℓ_1 and ℓ_2 given class c and parameters $\boldsymbol{\beta}_{c,pos}$ as

$$P_{pos}(\ell_1,\ell_2|c,\boldsymbol{\beta}_{c,pos}) = \delta_{[\ell_1,\ell_1+w_1-1]\cap[\ell_2,\ell_2+w_2-1],\emptyset}\Bigg[P(\mathcal{N}|\boldsymbol{\beta}_{c,pos})\frac{1}{Z_2(\boldsymbol{\beta}_{c,pos})}\mathcal{N}(\ell_1,\ell_2|\boldsymbol{\nu},\boldsymbol{\kappa},r_{1,2}) + P(\mathcal{U}|\boldsymbol{\beta}_{c,pos})\frac{1}{(L-w_1-w_2+1)^2+(L-w_1-w_2+1)}\Bigg], \quad (4.14)$$

where the mixture probabilities are parameterized identically to the univariate case and $Z_2(\boldsymbol{\beta}_{c,pos})$ denotes a normalization constant summing over all admissible combinations of po-

sitions, i.e.

$$Z_2(\boldsymbol{\beta}_{c,pos}) = \sum_{\ell_1=1}^{L-w_1+1} \sum_{\ell_2=1}^{L-w_2+1} \delta_{[\ell_1,\ell_1+w_1-1]\cap[\ell_2,\ell_2+w_2-1],\emptyset}\, \mathcal{N}(\ell_1,\ell_2|\boldsymbol{\nu},\boldsymbol{\kappa},r_{1,2}). \tag{4.15}$$

and assures that $P_{pos}(\ell_1,\ell_2|c,\boldsymbol{\beta}_{c,pos})$ is a proper probability distribution over the space of admissible combinations of positions. The bivariate Gaussian density is parameterized according to section 3.3.2.1, where $\boldsymbol{\nu} = (\nu_1,\nu_2)$, $\boldsymbol{\kappa} = (\kappa_1,\kappa_2)$, and $r_{1,2}$ denotes the parameter of correlation between the positions of the two motifs. In analogy to the motif models, we re-use the parameters ν_1 and κ_1 for the univariate position distribution of the first motif in the single-motif component $P_{m_1}(\boldsymbol{x}|c,\boldsymbol{\beta}_c)$, and ν_2 and κ_2 for the position distribution of the second motif in $P_{m_2}(\boldsymbol{x}|c,\boldsymbol{\beta}_c)$. This entails the assumption that – although occurrences of the two motifs may be correlated – the positions of binding sites of a single motif follow the same marginal distribution regardless of the presence of the second motif.

The normalization constants and, consequently, the position distribution depend on the length L of the sequence. Hence, we consider only target and control data sets that comprise sequences of identical length L. If we used a sequence-dependent normalization instead, the probabilities of identical positions in different sequences would differ and contradict the idea of a common position distribution of binding sites. In real-world applications, we consider this not a major limitation, because in most cases we can elongate all shorter sequences to the length of the longest sequence when extracting the data.

4.2.3.4. Priors

We use transformed Beta priors, which are a special case of the transformed Dirichlet prior (see section 3.4.2, p. 30) for one free parameter, for the class parameters β_c, the mixture parameters $\beta_{\mathrm{fw}|c,m}$ and $\beta_{\mathrm{bw}|c,m}$ of the strand model, the mixture parameters $\beta_{0|c}$ and $\beta_{1|c}$ of the ZOOPS model, and the mixture parameters $\beta_{\mathcal{N}|c,pos}$ and $\beta_{\mathcal{U}|c,pos}$ of the position distribution. For the parameters of the homogeneous Markov model and the parameters of the PWM model enclosed in the strand model, we choose transformed product-Dirichlet priors with hyper-parameters according to the assumption of uniform pseudo data, and we use another transformed Dirichlet prior for the mixture parameters $\boldsymbol{\beta}_{1,1|c}$, $\boldsymbol{\beta}_{1,0|c}$, $\boldsymbol{\beta}_{0,1|c}$, and $\boldsymbol{\beta}_{0,0|c}$ of the MuMo model.

We define the hyper-parameters of these priors based on the hyper-parameters α_c for the class parameters β_c. In the experiments, we set $\alpha_c = 4$. We a-priorily assume that each motif occurs in a fraction p_{motif} of the sequences and set the hyper-parameters for the mixture parameters of the ZOOPS model to $\alpha_{0|c} = p_{\mathrm{motif}} \cdot \alpha_{target}$ and $\beta_{1|c} = (1-p_{\mathrm{motif}}) \cdot \alpha_{target}$, and set the hyper-parameters for the mixture parameters of the MuMo model to $\boldsymbol{\beta}_{1,1|c} = p_{\mathrm{motif}}^2 \cdot \alpha_{target}$, $\boldsymbol{\beta}_{1,0|c} = \boldsymbol{\beta}_{0,1|c} = p_{\mathrm{motif}} \cdot (1-p_{\mathrm{motif}}) \cdot \alpha_{target}$, and $\boldsymbol{\beta}_{0,0|c} = (1-p_{\mathrm{motif}})^2 \cdot \alpha_{target}$, where $p_{\mathrm{motif}} = 0.7$. We further assume a-priorily that the motif occurs in either of the strand orientations with equal probability, resulting in hyper-parameters of $\alpha_{\mathrm{fw}|c,m} = \alpha_{\mathrm{bw}|c,m} = 0.5 \cdot p_{\mathrm{motif}} \cdot \alpha_{target}$. This choice of hyper-parameters is also valid for the MuMo model, since we use the same strand models for the component representing the joint occurrence of motifs, and the corresponding components representing the occurrence of a single motif.

For the PWM model enclosed in the strand model, we use an equivalent sample size (ESS) of $p_{\text{motif}} \cdot \alpha_{target}$, since this PWM model is employed for both strands. For the homogeneous Markov model in the *control* class, we use an ESS of $\alpha_{control}$ and we set the expected length of the sequences to $L_E = L$, whereas we use an ESS of α_{target} for the homogeneous Markov model representing flanking sequences and set the expected length to $L_E = p_{\text{motif}} \cdot (L - w) + (1 - p_{\text{motif}}) \cdot L$ in case of the ZOOPS model and $L_E = p_{\text{motif}}^2 \cdot (L - w_1 - w_2) + p_{\text{motif}} \cdot (1 - p_{\text{motif}}) \cdot (L - w_1) + p_{\text{motif}} \cdot (1 - p_{\text{motif}}) \cdot (L - w_2) + (1 - p_{\text{motif}})^2 \cdot L$ according to the lengths of the parts of the sequence that are modeled by this homogeneous Markov model in the different components and the a-priori probabilities assigned to these components.

Finally, we set the hyper-parameters of the mixture parameters of the position distribution to $\alpha_{\mathcal{N}|c,pos} = p_{\mathcal{N}} \cdot p_{\text{motif}} \cdot \alpha_{target}$ and $\alpha_{\mathcal{U}|c,pos} = (1 - p_{\mathcal{N}}) \cdot p_{\text{motif}} \cdot \alpha_{target}$ in the univariate case and $\alpha_{\mathcal{N}|c,pos} = p_{\mathcal{N}} \cdot p_{\text{motif}}^2 \cdot \alpha_{target}$ and $\alpha_{\mathcal{U}|c,pos} = (1 - p_{\mathcal{N}}) \cdot p_{\text{motif}}^2 \cdot \alpha_{target}$ in the bivariate case. We choose $p_{\mathcal{N}} = 0.2$ in order to assign a high probability to the uniform distribution if the binding sites are spread uniformly over the sequences, instead of learning an artificially small precision to adapt the Gaussian density to this distribution.

For the parameters of the Gaussian densities employed in the position distribution, we use a transformed normal-gamma prior in the univariate case and a transformed normal-Wishart prior in the bivariate case (see section 3.4.3, p. 33). In both cases, we set the a-priori means to the center of the sequence, i.e. $\mu_0 = \frac{L}{2}$ and $\boldsymbol{\mu}_0 = (\frac{L}{2}, \frac{L}{2})$, respectively. Since we are not confident in this a-priori assumption, we set the ESSs for the mean parameters to $\gamma = 10^{-4} \cdot \alpha_{\mathcal{N}|c,pos}$.

We further set the hyper-parameters of the normal-gamma prior to $\tau_1 = 0.5 \cdot \alpha_{\mathcal{N}|c,pos}$ and $\tau_2 = 0.5 \cdot \alpha_{\mathcal{N}|c,pos} \cdot 50^2$, which results in an expected precision of $\frac{1}{50^2}$ corresponding to a standard deviation of 50. Since the normal-Wishart density requires that $\alpha > 3$, we set the corresponding hyper-parameters to $\lambda_{0,d} = 0.5 \cdot p_{\mathcal{N}} \cdot p_{\text{motif}}^2 \cdot (\alpha_{target} + 8) \cdot 50^2$ and $\alpha = 0.5 \cdot p_{\mathcal{N}} \cdot p_{\text{motif}}^2 \cdot (\alpha_{target} + 8)$, which results in the same expected precisions κ_1 and κ_2 as for the normal-gamma prior.

The complete prior on the parameters of the ZOOPS and MuMo model, respectively, is then the product of all the component priors defined above. Since we re-use the parameters ν_1, ν_2, κ_1, and κ_2 of the bivariate Gaussian density in the corresponding univariate Gaussian densities of the MuMo model, these parameters are subject to the normal-gamma and the normal-Wishart prior.

4.2.3.5. Heuristic adaption of motif length and compensation for phase shifts

Similar to other approaches, the log supervised posterior using the ZOOPS or the MuMo model is not a concave function of the parameters and thus numerical optimization may get stuck in local optima or saddle points. A part of these local optima can be attributed to phase shifts, i.e. situation where the algorithm discovers only a shifted variant of the correct motifs which misses some of the relevant positions. Here, we propose a heuristic that compensates for phase shifts and additionally allows for an automatic adaption of the motif length w.

To this end, we determine the number of *irrelevant* positions at the left and right border of the PWM model representing the motif as follows: Let $\boldsymbol{p}_i$ denote the probability distribution at position i of the PWM model, let $\boldsymbol{p}_i^{rc}$ denote the reverse complementary probability distribution, and let $\boldsymbol{q}$ denote the probability distribution of the homogeneous Markov model representing flanking sequences. As a measure for the deviation of $\boldsymbol{p}_i$ from the distribution of the homogeneous Markov model, we use the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951). Since the PWM model is enclosed in a strand model, we compute the mean KL divergence of $\boldsymbol{p}_i$ and $\boldsymbol{p}_i^{rc}$ to $\boldsymbol{q}$. We define

$$D(\boldsymbol{p}_i||\boldsymbol{q}) := P(\text{fw}|\boldsymbol{\beta}_{c,m})D_{KL}(\boldsymbol{p}_i||\boldsymbol{q}) + P(\text{bw}|\boldsymbol{\beta}_{c,m})D_{KL}(\boldsymbol{p}_i^{rc}||\boldsymbol{q}). \tag{4.16}$$

We assess the significance of this deviation by simulations: we draw $1,000,000$ probability distributions $\boldsymbol{p}'$ from a Dirichlet density with parameters $\boldsymbol{\alpha} = (N + p_{\text{motif}} \cdot \alpha_{target}) \cdot \boldsymbol{q}$, where N denotes the number of sequences in the target data set and $p_{\text{motif}} \cdot \alpha_{target}$ is the ESS of the PWM model. The drawn probability distributions $\boldsymbol{p}'$ represent a population of probability distributions that deviate from $\boldsymbol{q}$ only by chance. For each of the distributions $\boldsymbol{p}'$, we compute $D(\boldsymbol{p}'||\boldsymbol{q})$ and use these values to fit a gamma density, which we find appropriate by inspecting histograms of the $D(\boldsymbol{p}'||\boldsymbol{q})$. We then determine the $(1-\alpha)$-percentile of the fitted gamma density and consider all deviations $D(\boldsymbol{p}_i||\boldsymbol{q})$ greater than this percentile significant for a significance level of α. We call positions i with probability distributions $\boldsymbol{p}_i$ that do not deviate significantly from $\boldsymbol{q}$ irrelevant. In the following, we use $\alpha = 10^{-30}$ for single motifs and $\alpha = 10^{-15}$ for multiple motifs.

Let n_ℓ denote the consecutive number of irrelevant positions at the left border of the motif and let n_r denote the consecutive number of irrelevant positions of the right border. We first test if all positions of the PWM model are considered irrelevant by the heuristic, i.e. $n_\ell + n_r \geq w$. If this is the case, we set the length of the motif to 1 , which effectively removes the motif, since a motif of length 1 will most likely approach an uninformative probability distribution. A complete erasure of the motif model is not considered mainly for technical reasons, because it would require extensive post-processing, e.g. the adaption of the bivariate position distribution to the univariate case or, alternatively, the removal of the component comprising both motifs. If $n_\ell + n_r < w$, we shift the parameters of the PWM model to one side such that the larger number of irrelevant positions is excluded from the model, i.e. if $n_\ell \geq n_r$, we shift the parameters to the left, and if $n_\ell < n_r$, we shift the parameters to the right. We set the parameters of the newly included positions at the opposite side to a uniform distribution over the nucleotides. For instance, assume that $n_\ell = 1$ and $n_r = 0$. In this case, we shift the parameters of the PWM model to the left by setting $\forall a \in \Sigma$: $\xi_{\ell,a|c} := \xi_{\ell+1,a|c}, \ell = 1, \ldots, w-1$, and $\xi_{w,a|c} := 0$.

We avoid cyclic sequences of operations by keeping a history of performed operations. If a proposed shift operation would result in a cycle, it is forbidden by the history and we shrink the motif by removing n_ℓ positions from the left border and n_r positions from the right border of the PWM model. Accordingly, the length of the resulting PWM model is adapted to $w - n_\ell - n_r$. If we do not find irrelevant positions at either border of the PWM model, we expand the PWM model by appending additional positions to both sides of the PWM model. If the current PWM model is shorter than the length that was initially specified when starting

the algorithm, we expand the PWM model evenly on both sides such that we obtain the initial length. Otherwise, one position is appended to each side of the PWM model. The newly included positions are again set to a uniform distribution over the nucleotides.

After each heuristic step, i.e. each shift, shrink, or expand operation, we restart the numerical optimization. We repeat with this cycle of heuristic adaption of the PWM model and numerical optimization until either all operations are forbidden by the history or we reach a pre-defined maximum number of heuristic steps. Finally, we report the ZOOPS or MuMo model that achieved the maximum supervised posterior after optimization among all models considered during these iterations.

Even using this heuristic, numerical optimization may get stuck in local optima. Hence, we start the optimization multiple times using random initializations. To exclude less promising initializations, we conduct a pre-selection on a fixed number of random initializations for each run of the optimization. For each random initialization considered, we evaluate the corresponding supervised posterior and finally start the numerical optimization for that random initialization which achieved the maximum supervised posterior. For the ZOOPS model, we test 50 independent starts of the optimization, each selecting the initial parameters from 100 independent random initializations. For the MuMo model, we reduce the number of independent starts to 20 due to the increased runtime, which is partly compensated for by selecting the initial parameters from 1000 independent random initializations for each of the 20 starts.

4.2.3.6. Prediction of binding sites

We predict binding sites based on the joint probabilities $P_m(\boldsymbol{x}, \ell | c, \boldsymbol{\beta}_c)$ of sequence $\boldsymbol{x}$ and position ℓ. In case of the MuMo model, we compute these as marginal distributions $P'_{m_i}(\boldsymbol{x}, \ell | c, \boldsymbol{\beta}_c)$, i.e.

$$P'_{m_1}(\boldsymbol{x}, \ell | c, \boldsymbol{\beta}_c) = \frac{P(u_1 = 1, u_2 = 0 | c, \boldsymbol{\beta}_c)}{P(u_1 = 1, u_2 = 0 | c, \boldsymbol{\beta}_c) + P(u_1 = 1, u_2 = 1 | c, \boldsymbol{\beta}_c)} P_{m_1}(\boldsymbol{x}, \ell | c, \boldsymbol{\beta}_c) \tag{4.17}$$
$$\frac{P(u_1 = 1, u_2 = 1 | c, \boldsymbol{\beta}_c)}{P(u_1 = 1, u_2 = 0 | c, \boldsymbol{\beta}_c) + P(u_1 = 1, u_2 = 1 | c, \boldsymbol{\beta}_c)} \sum_{\ell_2} P_{m1,m2}(\boldsymbol{x}, \ell, \ell_2 | c, \boldsymbol{\beta}_c)$$

and accordingly for $P'_{m_2}(\boldsymbol{x}, \ell | c, \boldsymbol{\beta}_c)$.

We compute these joint probabilities for each admissible position ℓ in each sequence $\boldsymbol{x}$ of the *control* data set, to obtain a background distribution of joint probabilities. We then choose a threshold T on the joint probabilities such that a fraction of α of the positions in all sequences of the control data set achieves a joint probability above this threshold. If we apply this threshold for predicting binding sites at all position ℓ in each sequence $\boldsymbol{x}$ for which $P_m(\boldsymbol{x}, \ell | c, \boldsymbol{\beta}_c) > T$, we predict at most $\alpha \cdot N' \cdot (L - w + 1)$ binding sites in a control data set comprising N' sequences. We use the same threshold to predict significant occurrences of the motif in the sequences of the target data set. Since we assume that the control data set does not contain occurrences of the motif of interest, this choice of the threshold should keep the number of false positive predictions in the target data set low.

4.2.4. Data

Several benchmark data sets for the assessment of de-novo motif discovery algorithms have been proposed over the last years (Tompa et al., 2005; Sandve et al., 2007; Kim et al., 2008). However, these benchmark data sets often comprise only a small number of fairly long sequences. To assess the significance of motifs discovered in such data sets, we extract putative promoter regions of length 2000 from TAIR (Swarbreck et al., 2008) and search for common sub-strings with one mismatch allowed in 5, 10, and 100 randomly selected promoters. With a probability of almost 1, we find such common sub-strings of lengths up to 8 in data sets with at most 100 promoters, and common sub-strings of lengths up to 10 in the data sets comprising 5 promoters. The lengths of these common sub-strings are in the range that can also be expected for transcription factor binding sites, and we anticipate that this problem could be even more severe for fuzzy motifs represented by PWM models. Hence, we decide to create new benchmark data by planting known binding sites of different transcription factors into randomly selected putative promoter regions of length 500.

To this end, we obtain the seven largest data sets of known binding sites from Jaspar (Sandelin et al. (2004), retrieved 10/9/2009). These are binding sites of the transcription factors AGL3 (MA0001) and AG (MA0005) of *Arabidopsis thaliana*, Cf2_II (MA0015) of *Drosophila melanogaster*, NHLH1 (MA0048), MEF2A (MA0052), and SOX9 (MA0077) of *Homo sapiens*, and Myb.PH3 (MA0054) of *Petunia x hybrida*. We plant the binding sites of the factors of *A. thaliana* into randomly selected promoters of the same species obtained from TAIR and use another set of randomly selected promoters as control data set; we apply the same procedure to the binding sites of the factors of *H. sapiens* using randomly selected promoters obtained from the human promoter database[3], and to the binding sites of the factor of *D. melanogaster* using randomly selected promoters of the same species from the eukaryotic promoter database[4]. For the binding sites of the factor stemming from *Petunia*, we also use promoters from TAIR, since no promoter regions for *Petunia* are available.

For each of the target data sets, we randomly select 70% of the promoter regions into which we plant a binding site, while we do not plant a binding sites into the remaining 30%. We plant the binding sites into the promoter regions at positions that are selected either according to a uniform position distribution (denoted as *uniform data sets* in the following) or according to a Gaussian distribution (denoted as *Gaussian data sets*). Each of the binding sites is either implanted on the forward or the backward strand, which is chosen randomly as well. The means of the employed Gaussian distributions are drawn uniformly from the interval $[20, 480]$ and the standard deviations are drawn uniformly from the interval $[20, 80]$. This procedure results in 14 benchmark data sets for discovering single motifs.

Based on the two data sets created for MA0048, we create four additional benchmark data sets. Here, we additionally plant "decoy" binding sites of MA0052 into the sequences of the target *and* the control data sets, once following a uniform position distribution and once following a Gaussian position distribution.

[3] `http://zlab.bu.edu/~mfrith/HPD.html`
[4] `http://www.epd.isb-sib.ch/index.html`

For assessing the predictions of the approaches for the de-novo discovery of cis-regulatory modules, we create four benchmark data sets by planting binding sites of two different transcription factors into the sequences of the target data set. Similar to the benchmark data sets for single motifs, binding sites of each of the two motifs are planted into 70% of the promoters in the target data set, and this subset of promoters is drawn independently for the two motifs. As a consequence, approximately 9% of the promoters do not contain a planted binding site, 42% contain a binding site of only one of the motifs, and the remaining 49% contain binding sites of both motifs. We create two benchmark data sets by planting binding sites of MA0001 and MA0005 into promoters of *A. thaliana*, once following a uniform position distribution and once following a bivariate Gaussian distribution, and we create two additional data sets by applying the same procedure to binding sites of MA0048 and MA0052, and promoters of *H. sapiens*.

We additionally consider two real-world data sets of auxin responsive genes, which are also considered in (Keilwagen et al., 2010a). The first data set comprises 48 promoters of genes that exhibit a two-fold increase of gene expression after auxin exposure for 15, 30, or 60 minutes in a cell suspension culture of *A. thaliana* cells (Paponov et al., 2008). Auxin is a plant hormone that plays a pivotal role in many regulatory processes related to plant growth and development. We choose cell suspension data in this case, since, due to its homogeneity, measurements of gene expression levels are not influenced by additional factors like tissue-specific expression levels. As a control data set, we randomly select 1000 promoters of genes that are not contained in the target data set, but have dedicated probes on the ATH1 microarray chip used for expression profiling.

We use an additional independent test set of 113 promoters of genes that are differentially expressed in seedlings of *A. thaliana* according to the same criteria applied for cell suspension culture (Paponov et al., 2008), but that are not contained in the cell suspension target data set. As a control data set we choose all 21012 promoters of *A. thaliana* genes that are present on the ATH1 chip, but are neither contained in the cell suspension target or control data set nor in the target data set for the seedling data.

4.2.5. Assessment

Several measures have been proposed for the assessment of de-novo motif discovery including nucleotide PPV and nucleotide Sn (Tompa et al., 2005). These measures are defined by regarding de-novo motif discovery as a classification problem: Each position ℓ in each sequence $\boldsymbol{x}$ may either be covered by a binding site (positive class) or not (negative class) according to the annotation. Using these annotations and predictions resulting from de-novo motif discovery, we define true positives (TP), false negatives (FN), false positives (FP), and, consequently Sn and PPV (cf. section 3.5.1, p. 38). As noted in the previous section, the prediction of binding sites by the ZOOPS and MuMo model depends on a threshold T. In analogy to other classification problems, we can vary this threshold and plot the resulting values of PPV against the values of Sn, yielding a PR curve that represents the overall prediction performance of these tools (Keilwagen et al., 2010a).

All other approaches considered in this work apply a pre-defined threshold for predicting binding sites, which complicates the generation of PR curves. However, all approaches report either p-values or scores for each of the predicted binding sites. We use these p-values or scores to compute *partial* PR curves up to the maximum Sn determined by the threshold.

4.2.6. Results & Discussion

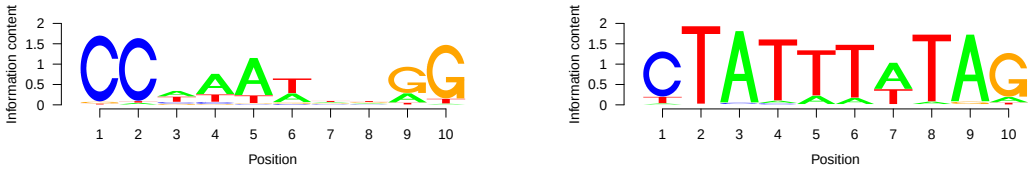
In this section, we compare the ZOOPS model using a position distribution and learned by the discriminative MSP principle as proposed in this work to several other approaches for de-novo motif discovery, namely MEME, Gibbs sampler, Improbizer, Weeder, A-GLAM, DME, and DEME. We also compare the MuMo model for de-novo discovery of cis-regulatory modules proposed in this work to other approaches that are specifically designed for this task, namely CoBind, CisModule, and MoAn. We also tested EMCModule but stopped computations after two weeks of runtime without any output. We refer to the approach proposed in this work as *MuMFi*, which is an acronym for "multiple motif finder".

We run all of the programs using default parameters with the following exceptions: if available and not the default, we use switches for searching on both strands, for enabling a position distribution, and for using a ZOOPS model instead of OOPS. We start each of the programs – including the implementation of MuMFi – once specifying correct length of the motif and once with switches for the automatic adaption of motif length. If such a switch is not available, we set the length of the motif to 15. A list of the calls for all programs is given in appendix A.2.

4.2.6.1. Benchmark for single motifs

We give an overview of the results of the comparison for all uniform and all Gaussian data sets in figure 4.6. Instead of complete PR curves, we use a condensed presentation that displays the achieved PPV for different values of Sn as bars. Here, we choose Sns of 0.1, 0.3, 0.5, 0.7, and 0.9, which gives a reasonable overview of the complete PR curve. As mentioned in the previous section, all approaches to which we compare MuMFi use internal thresholds for their predictions. Hence, the maximum Sn reached by these approaches may remain below some of the chosen values of Sn, which we represent as missing bars in the barplots. For those approaches that do not achieve a Sn of at least 0.1 on a given data set, the corresponding block comprises only missing bars.

Figure 4.6(a) presents the results of all approaches on the uniform data sets if these are started with given correct length of the motif. We find that the discriminative approaches, namely DEME, DME and MuMFi discover the correct motifs for the largest number of data sets. DEME and DME find the correct motif for 8 of the 9 data sets, while MuMFi is successful in all 9 cases. However, MuMFi yields a lower PPV than DEME and DME for MA0052 and the MA0048 data sets. A-GLAM discovers the correct motif for 3 data sets, Gibbs sampler for 1 data set, Improbizer for 4 data sets, MEME for 3 data sets, and Weeder for 5 data sets. Scrutinizing the achieved values of PPV, we observe that some of the motifs appear to be more challenging than others: for instance, all approaches except A-GLAM discover the correct motif for the MA0052 data set, whereas only DEME, Weeder, and MuMFi are



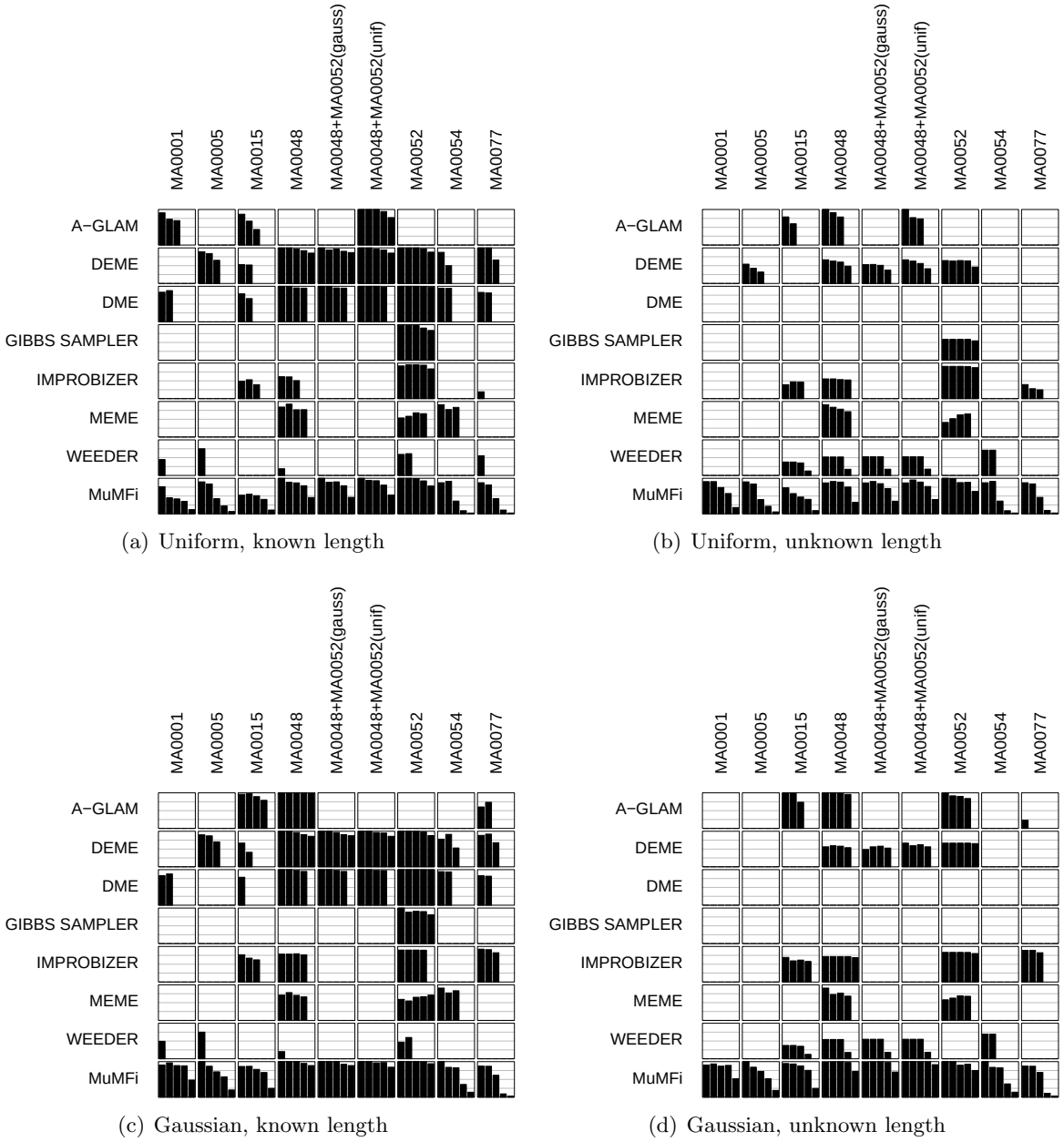
(a) MA0005 (b) MA0052

Figure 4.5.: Sequence logos of the binding sites of MA0005 (a) and MA0052 (b). The binding motif of MA0052 is highly conserved at all positions, whereas the motif of MA0005 exhibits many less conserved positions.

successful for MA0005. The sequence logos of the motifs of MA0005 and MA0052, which are presented in figure 4.5, show that this might be attributed to a different level of conservation of the two motifs. While MA0052 is highly conserved across all positions of the motif, MA0005 is considerably conserved only at the bordering positions 1, 2, 9, and 10, and at position 5.

Turning to the results for initially unknown lengths of the motifs in figure 4.6(b), we observe that the accuracy of some of the considered approaches is greatly decreased. The maximum PPV yielded by DEME decreases from values of almost 1 in figure 4.6(a) to values that are consistently below 0.75 for unknown motif length. The deterioration of performance is even more dramatic for DME, which is not capable of discovering any of the motifs without the length of the motif specified. Since neither DEME nor DME adapt the motif length automatically, this means that DME did not succeed in determining an elongated variant of the correct motif, whereas the reduced accuracy of DEME for MA0005, MA0052, and the MA0048 data sets can presumably be attributed to the incorrect lengths of the discovered motifs. Other approaches, namely A-GLAM, Improbizer, MEME, and Weeder, are less sensitive to the specification of the motif length. However, these approaches did already fail to discover the correct motif for many data sets with known motif length. Interestingly, MuMFi still discovers all 9 motifs and achieves an even improved accuracy for MA0001 and MA0015 compared to the results for known motif length. We will scrutinize this phenomenon in a few paragraphs.

Considering the results for the Gaussian data sets and known motif length, which are presented in figure 4.6(c), we find a similar picture as for the uniform data sets. The results of DEME, DME, Gibbs sampler, MEME, and Weeder are virtually unchanged regarding the data sets for which the correct motif could be discovered and regarding accuracy on these data sets. This behavior can be expected, since neither of these approaches learns a position distribution from the data (cf. table 4.2), and we planted identical binding sites into the same promoters for the uniform and Gaussian data sets. In contrast, the results of those approaches that learn a position distribution, namely A-GLAM, Improbizer, and MuMFi, profit from the appropriate modeling of the position distribution to different degrees. On the one hand, A-GLAM achieves increased values of PPV for MA0015 and, in contrast to the uniform data sets, correctly discovers MA0048 and MA0077 for the Gaussian data sets. On the other hand, it does not discover the motif for the MA0001 data set, which is discovered by A-GLAM for the uniform position distribution. The accuracy of Improbizer is increased for most data sets for which a motif is discovered at all, namely MA0015, MA0048, and MA0077. Finally, MuMFi profits



(a) Uniform, known length

(b) Uniform, unknown length

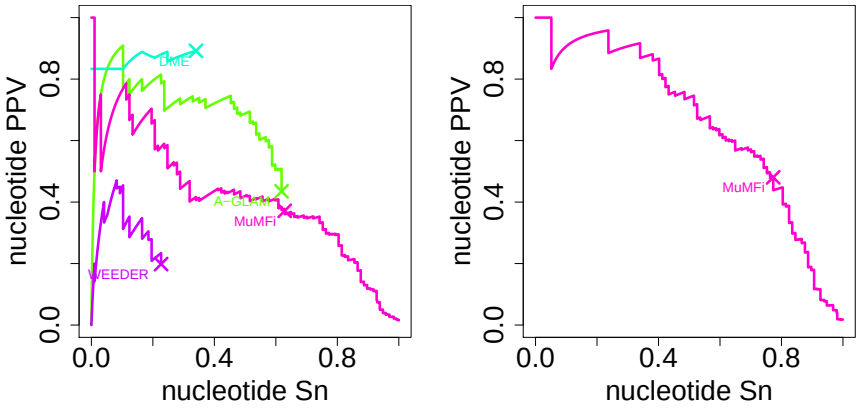
(c) Gaussian, known length

(d) Gaussian, unknown length

Figure 4.6.: Overview of the prediction performance of all approaches on all data sets. The results in the first row are determined on the uniform data sets, and those in the second row are determined on the Gaussian data sets. In the left column, we present the result for the experiments where we start all approaches with the correct length of the motif, and in the right column the length of the motif is either fixed to 15 or – if available – learned from the data. In each block of each sub-figure, we plot the values of PPV for a Sn of 0.1, 0.3, 0.5, 0.7, and 0.9, i.e. specific points on the PR curve, as bars. For some of the approaches, the maximum Sn is limited to a value below 0.9 due to an internally applied threshold, which results in missing bars for the corresponding values of Sn. In each sub-figure, the columns labeled "MA0048+MA0052" correspond to the data sets with additional decoy motif.

from the Gaussian position distribution for all data sets except MA0077 and, in contrast to the uniform data sets, achieves similar values of PPV as DEME and DME for MA0048 and MA0052.

As a last overview, we examine the results of all approaches for the Gaussian data set if these are started with unknown motif length in figure 4.6(d). As for the uniform data sets, the performance of DEME and DME suffers from the lack of an automatic adaption of the motif length. Compared to known motif length, A-GLAM additionally discovers the correct motif for MA0052, whereas its achieves an decreased accuracy for the MA0077 data set. While the results of Improbizer are virtually unchanged compared to the known motif length, Gibbs sampler does not find the correct motifs for any of the 9 Gaussian data sets if it is not provided the correct motif length. The results of MEME are similar to those for the known motif length regarding MA0048 and MA0052, but it now fails to discover the correct motif for MA0054. Weeder finds the correct motif for 5 of the Gaussian data sets with unknown length of the motif as opposed to 4 correctly identified motifs for known motif length. Finally, the results of MuMFi are virtually identical for known and unknown motif length on the Gaussian data sets. This indicates that MumFi could successfully adapt the length of the motif by the heuristic described in section 4.2.3.5. We consider this last benchmark the most realistic, since the binding sites of most transcription factors exhibit a non-uniform position distribution and in real-world applications we seldom know the correct length of the motif in advance.

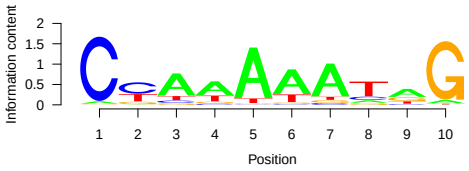


(a) Uniform, known length

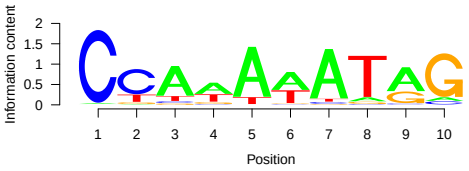
(b) Uniform, unknown length

Figure 4.7.: PR curves for the uniform MA0001 data set comprising binding sites and promoters stemming from *Arabidopsis thaliana*. Approaches with Sn and PPV below 0.1 are omitted for clarity.

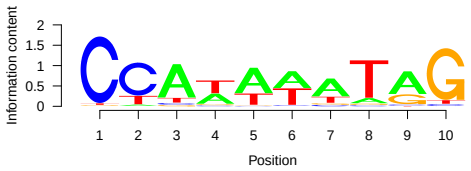
As one specific example, we consider the complete PR curves of MuMFi for the uniform MA0001 data set, and we compare these to the partial PR curves achieved by the other approaches. In figure 4.7 and the following plots, we omit approaches that did neither reach a Sn nor a PPV above 0.1 to avoid an uninformative cluster of curves in the lower left corner of the plots. Comparing figure 4.7(a) to figure 4.7(b), we find in accordance with figure 4.6 that DME and A-GLAM achieve greater values of PPV than MuMFi if the correct motif length is specified in advance, whereas MuMFi is the only approach that discovers the correct motif of MA0001 for unknown motif length. As already noted for the overview, MuMFi achieves an even improved PR curve if the motif length is not specified, which seems counter-intuitive.



(a) MuMFi, known length



(b) MuMFi, unknown length



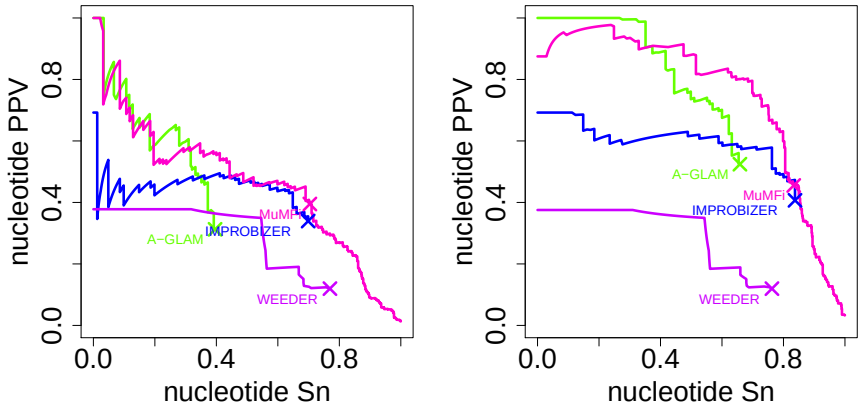
(c) Sequence logo according to annotation

Figure 4.8.: Sequence logos of the binding sites predicted by MuMFi with given (a) and adapted (b) length of the motif compared to the sequence logo of the annotated binding sites (c).

To investigate potential reasons for this observation, we plot the sequence logos of the binding sites predicted by MuMFi for given motif length and unknown motif length in figure 4.8, and compare these to the sequence logo of the annotated binding sites. Here and in the following, we generate the sequence logos for MuMFi by predicting occurrences of the discovered motifs using a p-value of $\alpha = 10^{-4}$ (see section 4.2.3.6), estimating a PWM model by the generative ML principle from these predictions, and plotting the sequence logo for the estimated PWM model. We find that both sequence logos of the predictions of MuMFi are similar to the sequence logo of the annotated binding sites and comprise the correct number of positions. However, the sequence logo for unknown motif length shows a greater conformity to the sequence logo of figure 4.8(c), especially at positions 8 and 9. One possible explanation for these differences might be that additional iterations of numerical optimization during the adaption of the motif length (cf. section 4.2.3.5) helped MuMFi to escape local optima in case of unknown motif length.

As another example, we study the effects of the position distribution of the planted binding sites on the accuracy of the different approaches. To this end, we present the results of the approaches studied on the uniform and Gaussian data sets for MA0015 in figure 4.9. Comparing figures 4.9(a) and 4.9(b), we find that approaches that explicitly model a position distribution, namely A-GLAM, Improbizer, and MuMFi, yield considerably improved PR curves for the Gaussian data set. Among these approaches, MuMFi achieves the best values of PPV for the larger values of Sn, whereas A-GLAM performs slightly better than MuMFi for a Sn below approximately 0.35. In contrast, Weeder obtains almost identical PR curves on the uniform and the Gaussian data set, as it can not exploit the position distribution for predicting binding sites.

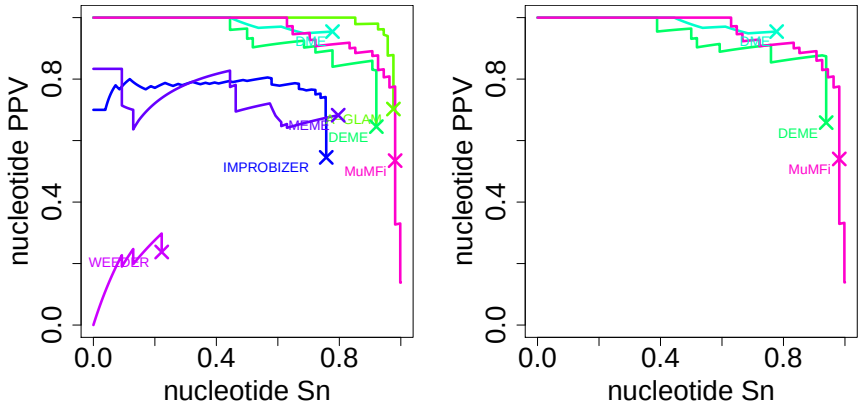
Finally, we demonstrate the advantages of discriminative approaches for discovering differentially abundant motifs as opposed to over-represented motifs. To this end, we compare the performance of all approaches on the Gaussian data set for MA0048 to that on the Gaussian data set for MA0048 with an additionally planted decoy motif, namely MA0052. The drawn



(a) Uniform, unknown length

(b) Gaussian, unknown length

Figure 4.9.: PR curves for the uniform (left) and Gaussian (right) MA0015 data set and unknown length of the motif. Approaches with Sn and PPV below 0.1 are omitted for clarity.



(a) Gaussian, unknown length

(b) Gaussian, unknown length, with decoy motif

Figure 4.10.: PR curves for binding sites of the TF MA0048 of *Homo sapiens* placed in randomly selected promoters of the same species. In the right plot, an additional decoy motif (MA0052) is planted into the target and control sequences. Tools with Sn and PPV below 0.1 are omitted for clarity.

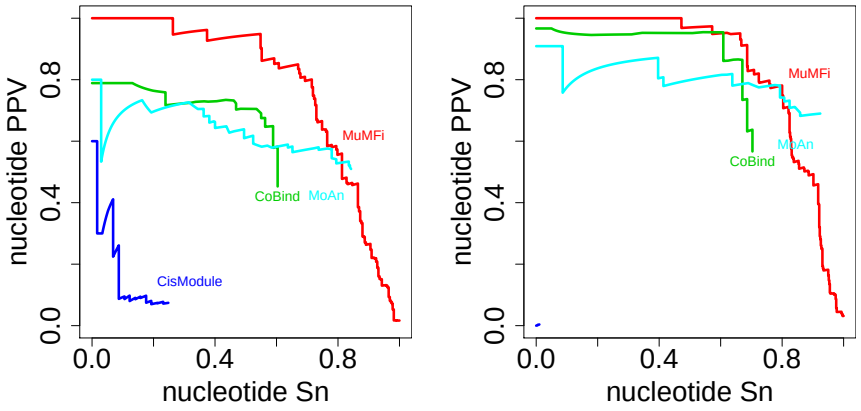
positions for the binding sites of MA0048, the binding sites, and the promoter sequences into which these binding sites are planted are identical in both cases to assure that differences in accuracy can be attributed solely to the presence of the decoy motif. The results of this experiment are depicted in figure 4.10.

Many of the studied approaches discover the correct motif of MA0048 in the data set without decoy motif. A-GLAM, MEME, Improbizer, DEME, DME, and MuMFi achieve satisfactory PR curves, whereas Weeder does not yield a PPV or Sn above 0.4. On the data set with planted decoy motif, this picture changes considerably. None of the generative approaches is able to discover the correct motif in this case, most likely due to the high conservedness of the decoy motif MA0052 (cf. figure 4.5), which is hence preferred to MA0048 by the generative approaches. In contrast, the PR curves of the discriminative approaches, namely DEME,

DME, and MuMFi, for the data sets with and without decoy motif differ only slightly.

4.2.6.2. Benchmark for multiple motifs

We also compare MuMFi in its variant for the de-novo discovery of cis-regulatory modules to other approaches that are specifically designed for this task, namely CoBind, CisModule, and MoAn. Since all studied approaches adapt the length of motifs, we only consider the case of an unknown motif length in the following.

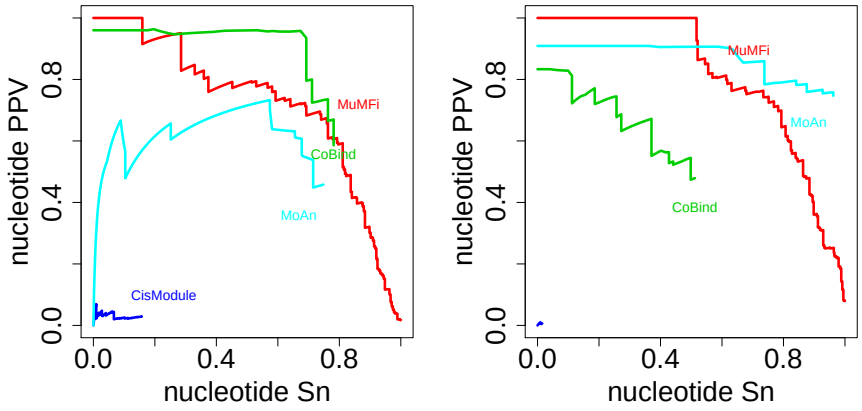


(a) MA0048 (b) MA0052

Figure 4.11.: PR curves regarding the binding sites of MA0048 (a) and MA0052 (b) for the uniform data set comprising binding sites of both factors.

As a first benchmark data set, we consider the uniform data set with planted binding sites of MA0048 and MA0052. We compute the PR curves separately for each of the motifs, and we resolve label switching by reporting the best curves of the two possible combinations of annotated and predicted motifs. The results of this analysis are illustrated in figure 4.11. We find that CoBind, MoAn, and MuMFi achieve a satisfactory accuracy for both motifs, whereas CisModule performs considerably worse for MA0048 and fails to identify the correct motif of MA0052. For MA0048, MuMFi achieves considerably larger values of PPV than CoBind and MoAn for a broad range of Sn. Since MuMFi can not profit from a non-uniform position distribution in this case, and MoAn and MuMFi both use a discriminative objective function, we might speculate that the improvement of PPV gained by MuMFi over MoAn can be attributed to the heuristic compensating for phase shifts and the parameter prior employed in the MSP principle. For MA0052, MuMFi achieves slightly improved values of PPV compared to MoAn for a Sn below 0.8, whereas MoAn yields a sightly larger PPV for Sn between 0.8 and 0.9. Although the PR curve of CoBind stays below that of MuMFi for most valued of Sn, the differences between the two approaches are only minor for a Sn below 0.6.

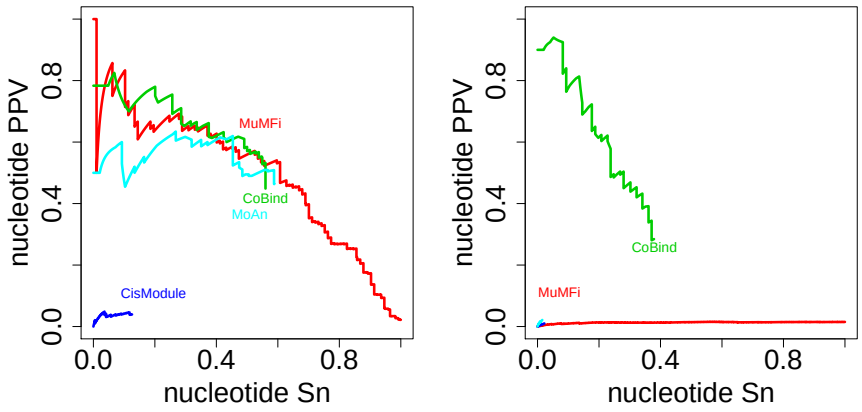
Turning to the corresponding results on the Gaussian data sets for MA0048 and MA0052, which are displayed in figure 4.12, we find a slightly different picture for CoBind. While CoBind achieves a considerably improved accuracy for the binding sites of MA0048, the PR curve for MA0052 is notably lowered. Comparing MuMFi to CoBind and MoAn, we observe



(a) MA0048 (b) MA0052

Figure 4.12.: PR curves regarding the binding sites of MA0048 (a) and MA0052 (b) for the Gaussian data set comprising binding sites of both factors.

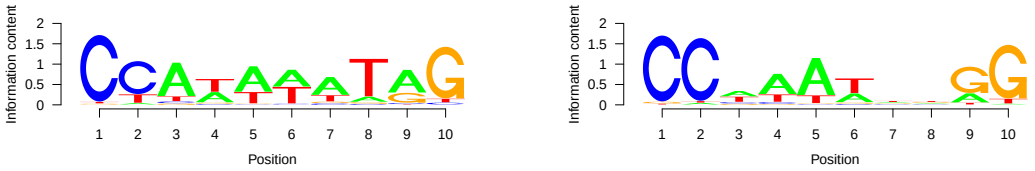
that CoBind yields a better accuracy than MuMFi for MA0048 and MoAn yields a better accuracy for MA0052 than MuMFi. However, neither of the two approaches does consistently outperform MuMFi on this data set. Like for the uniform data set, CisModule does not achieve reasonable accuracies for any of the two motifs.



(a) MA0001 (b) MA0005

Figure 4.13.: PR curves regarding the binding sites of MA0001 (a) and MA0005 (b) for the uniform data set comprising binding sites of both factors.

In addition, we test the approaches on two other benchmark data sets, for which binding sites of MA0001 and MA0005 are planted into promoters of *A. thaliana* using a uniform and a Gaussian position distribution. We first consider the uniform data set. The PR curves of the studied approaches are presented in figure 4.13. We find that CoBind, MoAn, and MuMFi discover the binding sites of MA0001 with similar accuracy, whereas CisModule again fails to find the correct motif. Turning to the binding sites of MA0005, all approaches except CoBind essentially fail to discover the correct motif. We investigate potential reasons of this observation in the following.



(a) MA0001 (b) MA0005

Figure 4.14.: Sequence logos of the annotated binding sites of MA0001 (a) and MA0005 (b).

The sequence logos of MA0001 and MA0005 are depicted in figure 4.14. These show two properties of the motifs: first, both are highly conserved only at a fraction of positions and most of these are located at the borders of the motifs, and second, both motifs are fairly similar at the conserved positions. This is an example for possible pitfalls of discriminative approaches like MoAn and MuMFi, which focus on motifs that discriminate target and control sequences best. Since the motifs of both factors are similar and the binding sites of both factors exhibit a uniform position distribution, one joint representation of both motifs may be sufficient for discrimination. Hence, a second motif, which may be sightly differentially abundant between target and control data set but not a motif of interest, is learned if this improves the discrimination between the two data sets.

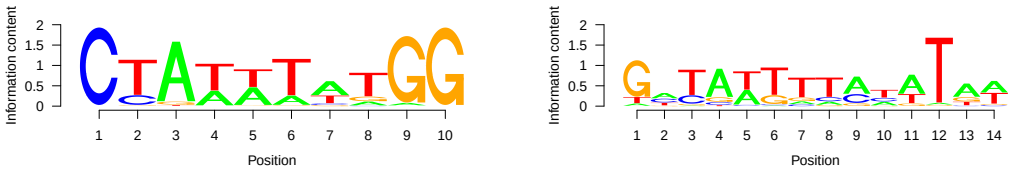
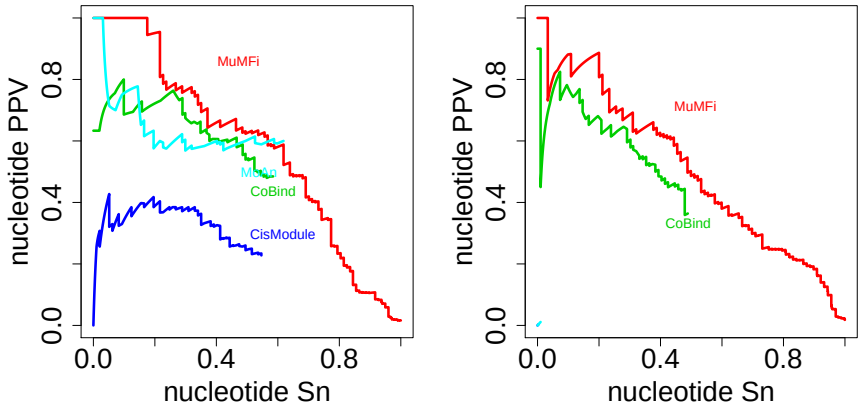


Figure 4.15.: Sequence logos of the binding sites predicted by MuMFi on the uniform data set comprising binding sites of MA0001 and MA0005.

The sequence logos of the binding sites predicted by MuMFi, which are presented in figure 4.15, support this assumption. The motif on the left is a mixture of the reverse complementary motif of MA0001 and the motif of MA0005. In contrast, the second motif of binding sites discovered by MuMFi is conserved only at position 14 and shows no similarity to the motifs of interest.

Turning to the Gaussian data set for the same factors, we observe that CisModule identified the correct motif of MA0001 in this case. However, the remaining three approaches yield significantly larger values of PPV across the range of Sn than CisModule. While the accuracy of MuMFi is only slightly increased for MA0001 compared to the results on the uniform data set, MuMFi is now able to discover the correct motif and binding sites of MA0005. In contrast, MoAn still fails to identify the motif of MA0005. This might be an indication that the position distribution modeled by MuMFi but not by MoAn might be responsible for the increased performance.

We scrutinize this assumption in figure 4.17, where we present the sequence logos of the



(a) MA0001 (b) MA0005

Figure 4.16.: PR curves regarding the binding sites of MA0001 (a) and MA0005 (b) for the Gaussian data set comprising binding sites of both factors.

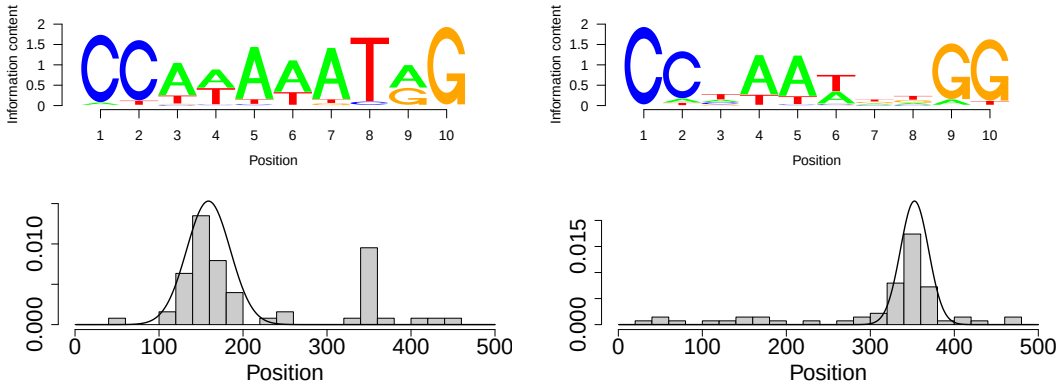


Figure 4.17.: Sequence logos and position distribution of the binding sites predicted by MuMFi on the Gaussian data set comprising binding sites of MA0001 and MA0005. The plots of the position distribution show a histogram of the positions of the binding sites predicted by MuMFi and the position distribution learned by MuMFi as a solid black line.

binding sites predicted by MuMFi. Additionally, we plot a histogram of the positions of the predicted binding sites and the position distribution learned by MuMFi. We observe that both sequence logos represent the specialties of MA0001 and MA0005 well and even slightly amplify conserved nucleotides. Considering the position distribution of the predictions, we find that the binding sites of MA0001 and MA0005 are clustered at distinct regions of the promoter sequence. However, MuMFi still predicts some occurrences of MA0001 in the region around position 350, which is the center of the cluster of MA0005 binding sites, most likely due to the similarity of the two motifs. This last benchmark endorses that discriminative learning of parameters and learning the position distribution from the data support the de-novo discovery of relevant motifs and corresponding binding sites.

4.2.6.3. Applying MuMFi to promoters of auxin responsive genes

In a final study, we investigate the utility of MuMFi on real-world data. To this end, we learn the parameters of the sequence models and the position distribution on the cell suspension target data set comprising promoters of auxin responsive genes and the corresponding control data set. We first learn MuMFi with a ZOOPS model, i.e. with a single motif model. The motif and position distribution discovered by MuMFi are depicted in figure 4.18. MuMFi finds a motif that is similar to the canonical auxin response element (ARE) TGTCTC (Paponov et al., 2008), but nonetheless exhibits some interesting differences. First, the consensus Cs of the canonical ARE at position 4 and 6 are not fully conserved, but may also be replaced by G. Second, MuMFi discovers a highly conserved C at position 8 of the motif, which is not part of the canonical ARE.

Turning to the position distribution learned by MuMFi, we find a strong positional preference of occurrences of the discovered motif. Most of the predicted binding sites are located at most 250 bp upstream of the TSS and – on the cell suspension data set comprising 48 sequence – no predicted binding site is located more than 300 bp upstream of the TSS. In figure 4.18, we additionally plot the position distribution learned by MuMFi as a solid black line. We find that the position distribution is in good accordance with the predicted binding sites. The mean of the Gaussian density is located 128 bp upstream of the TSS with a precision of $\sim 2.1 \times 10^{-4}$, which corresponds to a standard deviation of ~ 70. The Gaussian component of the position distribution obtains a mixture probability of 0.94.

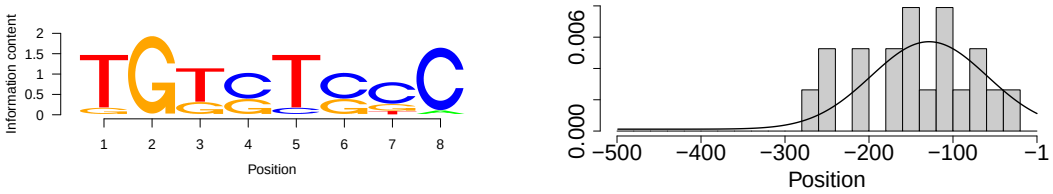


Figure 4.18.: Sequence logo and histogram of positions of the binding sites detected by MuMFi on the cell suspension data set with one allowed motif. In addition to the histogram of binding site positions, we plot the position distribution learned by MuMFi as a solid black line.

We assess the relevance of the motif and position distribution on the independent data set of auxin responsive genes in seedlings and the control data set comprising the promoters of all genes that are on the ATH1 chip but neither in the cell suspension training data nor in the seedling target data set. As a reference, we search for perfect matches of the canonical ARE in these two data sets. In addition to the region from -500 to -1 relative to the TSS, we search for the canonical ARE in a shortened region from -250 to -1, to evaluate the contribution of the positional preference discovered by MuMFi. For the seedling target data set and the control data set, we report the number of sequences that exhibit at least one occurrence of the canonical ARE and the number of sequences that contain at least one binding site predicted by MuMFi in table 4.3. Given these numbers, we additionally report the p-value of the enrichment of the canonical ARE and the discovered motif, respectively, relative to the control data set according to Fisher's exact test.

Considering perfect matches of the canonical ARE in the $[-500, -1]$ region, we find at least

one occurrence in 36 of the 113 promoters in the seedling target data set compared to 4741 out of 21012 promoters for the control data set. This corresponds to a ~ 1.4-fold enrichment of the canonical ARE in the target promoters relative to the control promoters, which results in a p-value of 1.5×10^{-2}. Hence, we consider the enrichment of the canonical ARE statistically significant. Turning to the search for the canonical ARE in the $[-250, -1]$ region, the number of promoters in the target data set containing the ARE consensus is reduced to 26, whereas only 2564 promoters of the control data set contain the ARE within the shortened region. Consequently, we find a ~ 1.9-fold enrichment of the canonical ARE in the shortened region, which corresponds to a p-value of 1.0×10^{-3}. Since the p-value according to Fisher's exact test decreases more than 10-fold due to the restriction of the considered region, we may conclude that the position distribution learned by MuMFi also contributes considerably to the specificity of the canonical ARE for auxin responsive genes.

Table 4.3.: Number of sequences stemming from the seedling and control data set that contain at least one occurrence of the canonical auxin response element (ARE) TGTCTC or at least one occurrence of the motif predicted by MuMFi, respectively. For the canonical ARE, we determine occurrences in the promoter region up to 500 bp upstream of the TSS and occurrences restricted to at most 250 bp upstream of the TSS. For each combination of the predictions within the seedling and control data set, we report p-values according to Fisher's exact test.

	TGTCTC			MuMFi		
	seedling	control	p-value	seedlings	control	p-value
total	113	21012		113	21012	
predicted [-500,-1]	36	4741	1.5×10^{-2}	26	2137	6.0×10^{-5}
predicted [-250,-1]	26	2564	1.0×10^{-3}			

The predictions of MuMFi depend on a threshold T, which we choose such that we obtain the same number of sequences with a least one predicted binding site as for the canonical ARE in the region $[-250, -1]$ of the target data set. Using this threshold, MuMFi predicts at least one binding site in only 2137 of the 21012 promoters of the control data set. This corresponds to a ~ 2.3-fold enrichment of the motif discovered by MuMFi in the target data set relative to the control data set, which results in a p-value of 6×10^{-5}. Since, we observe a 16-fold decrease of the p-value compared to the canonical ARE in the $[-250, -1]$ region and a 250-fold decrease compared to the canonical ARE in the $[-500, -1]$ region, we may conclude that the combination of the motif and the position distribution learned by MuMFi is highly specific for auxin responsive genes.

In (Keilwagen et al., 2010a), another approach called Dispom is proposed, which differs from MuMFi for single motifs only in the determination of irrelevant positions for the heuristic and in the choice of the position distribution. Dispom determines irrelevant positions by means of the number of sequences that are predicted to contain a binding sites before and after a potential modification. On the one hand, this heuristic leads to slightly improved PR curves compared to MuMFi (Keilwagen et al., 2010a). On the other hand, the heuristic of Dispom requires exhaustive testing for each position considered, which involves the computation of p-values for determining the number of sequences bound.

Dispom uses a skew normal position distribution. Since the extension of the skew normal

distribution to the multivariate case modeling correlations is not straightforward, we stick to the bivariate Gaussian distribution for MuMFi. In (Keilwagen et al., 2010a), Dispom is applied to the cell suspension and seedling data sets as well. On the cell suspension data set, Dispom discovers a motif and position distribution which is highly similar to those learned by MuMFi. Keilwagen et al. (2010a) use this motif to define a refined consensus sequence of AREs as TGTSTSBC. Searching for the refined ARE in the $[-250, -1]$ region results in at least one occurrence in 21 of the 113 promoters of the seedling target data set and at least one occurrence in 1252 of the 21012 promoters of the control data set, yielding an even decreased p-value of 3.5×10^{-6}.

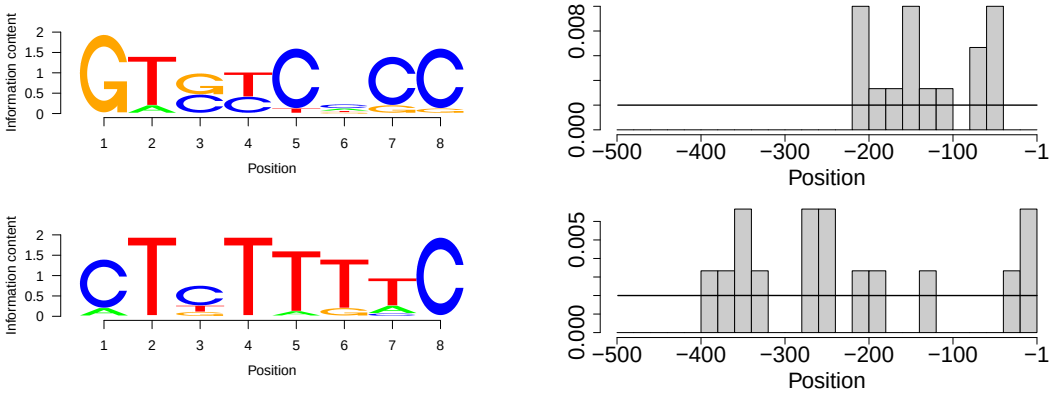


Figure 4.19.: Sequence logos and histograms of positions of the binding sites detected by MuMFi on the cell suspension data set with two motifs allowed. In addition to the histogram of binding site positions, we plot the marginal position distribution learned by MuMFi as a solid black line.

Although the motif and position distribution learned by MuMFi and the refined ARE defined by (Keilwagen et al., 2010a) are highly specific for auxin responsive genes, only a fraction of the promoters of the target data sets contains these elements. To investigate if an additional motif or a combination of motifs might explain the differential expression of a larger number of the auxin responsive genes, we learn MuMFi for cis-regulatory modules on the cell suspension data as well. The two motifs and corresponding position distributions learned by MuMFi are depicted in figure 4.19. The first motif discovered by MuMFi might be a shifted variant of the motif presented in figure 4.18, which is also supported by the histogram of positions of occurrences of this motif. In contrast, the binding sites of the second motif exhibit a less stringent position distribution. Considering the position distribution learned by MuMFi, we find a mixture probability of almost 1 for the uniform component, although this is contradicted by the positions of predicted binding sites of the first motif. For the first motif, we obtain a considerably higher p-value of 6.1×10^{-4} on the seedling data than for the motif and position distribution discovered by MuMFi using a single motif, whereas the second motif yields a lower p-value of 5.3×10^{-5}.

This indicates a potential shortcoming of MuMFi in its current implementation: The position distributions of the component comprising two motifs and the components comprising single motifs share the parameters for mean, precision, and mixture probabilities. If one of the motifs occurs uniformly in the target promoters, this may affect the position distribution of a second, non-uniformly distributed motif as well, leading to less accurate predictions for the second

motif. Hence, it may be worthwhile to also consider a variant of MuMFi without shared parameters in the future.

4.2.7. Conclusions

In this work we propose MuMFi, a novel approach for the prediction of cis-regulatory modules comprising binding sites of at most two different motifs. This approach incorporates a model for the position distribution of binding sites, which is a mixture of a uniform and a Gaussian distribution and is able to capture correlations between the binding sites of the two motifs. The parameters of the statistical models for motifs and flanking sequence and the parameters of the position distribution are learned by the discriminative MSP principle.

We compare MuMFi to seven other approaches for de-novo discovery of single motifs on 18 benchmark data sets with single planted motif, and find that it outperforms the other approaches with regard to the total number of motifs discovered. In most cases, MuMFi also achieves a comparable or even improved accuracy compared to the other approaches studied as measured by the PR curve. We find that the strengths of MuMFi are the combination of learning a position distribution from data, learning parameters discriminatively, and using a heuristic to compensate for phase shift and to automatically adapt the length of the motif.

We also demonstrate the utility of MuMFi on benchmark data sets comprising binding sites of two different motifs. For these data, we show that the accuracy of MuMFi greatly profits from the combination of discriminative learning and learning the position distribution from the data.

Applying MuMFi to the promoters of auxin responsive genes, we find a motif that may be interpreted as a refined and elongated variant of the canonical auxin response element. This motif, combined with a strong positional preference discovered by MuMFi, is highly specific for auxin responsive genes and yields a 250-fold decreased p-value in Fisher's exact test compared to the canonical auxin response element without positional preference.

4.3. Prediction of nucleosome positioning

4.3.1. Background

Nucleosomes are building-blocks of eukaryotic chromatin organization. Besides their importance for the compaction of eukaryotic genomes, nucleosome positioning influences the binding of transcription factors by steric hindrance (see also section 2.2). Hence, we are interested in the prediction of nucleosome positioning from sequence to reduce the number of false-positives when predicting TFBSs.

4.3.1.1. Sources of verified nucleosome positions

Most techniques for the experimental determination of nucleosome positions share the same initial step (Yuan et al., 2005; Lee et al., 2007; Field et al., 2008): Genomic DNA is extracted from cell cultures, cross-linked, and then digested by micrococcal nuclease. Micrococcal nuclease (MNase) is an endo-/exo-nuclease with a relatively low binding specificity and preferentially cuts A/T-dinucleotides (see also figure 4.36). Regions of DNA bound in nucleosomes are not accessible to MNase and consequently are protected from digestion, whereas linker sequences, i.e. the regions between the nucleosomes, are digested. After the digestion is stopped, the histone cores are removed by proteinase.

The detection of the undigested nucleosomal DNA is then accomplished either by hybridization to tiling microarrays (Yuan et al., 2005; Lee et al., 2007) or by parallel sequencing (Field et al., 2008). In the latter case, we directly obtain the sequences of nucleosomal DNA, whereas the evaluation of tiling microarrays requires the mapping from hybridization intensities to nucleosome positions. Additionally, the resolution achievable depends on the spacing of the tiling microarray. Yuan et al. (2005) use 50 nt probes tiled every 20 bp, whereas (Lee et al., 2007) achieve a resolution of 4 bp on an Affymetrix tiling microarray. The mapping from intensities to positions is commonly conducted employing hidden Markov models (HMMs) (Yuan et al., 2005; Lee et al., 2007; Yassour et al., 2008).

Field et al. (2008) obtain approximately $503,000$ reads of nucleosomal DNA by parallel sequencing of a pool of eight independent biological replicates of *Saccharomyces cerevisiae*. The 454 pyrosequencing technique used by Field et al. (2008) is capable of sequencing fragments up to ~ 200 bp, which is sufficient to reliably detect nucleosomal DNA with a length of ~ 147 bp. The reads are then mapped to the genome excluding those reads that map to repetitive regions and filtering for a required length of $127 - 177$ bp. The resulting set of approximately $380,000$ uniquely mapped reads corresponds to a five-fold coverage of the yeast genome. These data are also used in this work to learn a model of nucleosome positioning.

4.3.1.2. Related work

Several attempts have been made to predict nucleosome positioning from DNA sequence. For yeast (*S. cerevisiae*), Ioshikhes et al. (2006) calculate the relative frequency of AA and TT

dinucleotides along the sequences of approximately 200 well-positioned nucleosomes, i.e. nucleosomes with a low mobility. This pattern of length 139 is then slid along genomic sequences to scan for potential nucleosome positions. Ioshikhes et al. (2006) convert the genomic sequence under the sliding window to a sequence of relative AA/TT occurrences as well and measure its correlation to the pre-defined pattern. The resulting correlations are used as scores to predict non-overlapping nucleosome positions.

A similar approach is proposed by Segal et al. (2006), who use a WAM model smoothed over three neighboring positions combined with a homogeneous Markov model of order 0 as background model to scan for nucleosome positioning signals in yeast. The resulting scores serve as the input of a dynamic programming approach similar to the forward-backward algorithm for HMMs, which computes the final probabilities of nucleosome occupancy. Segal et al. (2006) find ~ 10 bp periodic A/T dinucleotides for sequences bound in nucleosomes and similar periodicities in nucleosome-bound sequence from chicken.

Nucleosome positioning in yeast is also studied by Peckham et al. (2007), who employ support vector machines (SVMs) with a linear kernel. The kernel considers the number of occurrences of all 1 to 6-mers of the input sequence. Each input sequence is converted into a vector of the number of k-mer occurrences for each possible k-mer in Σ^k, where $k = 1, \ldots, 6$ and k-mers and their reverse complement are identified, e.g. ACG and CGT are considered as the same k-mer. The approach of Peckham et al. (2007) is adapted by Gupta et al. (2008) to predict nucleosome positions in human DNA sequences.

Lee et al. (2007) use a linear model on features selected by the Lasso method (Tibshirani, 1994) to predict nucleosome positions according to data obtained by tiling microarrays, which are presented in the same publication. The features selected by Lasso are the helical properties tip, tilt, propeller twist, and roll (see section 4.3.2.6), several k-mer occurrences, and the known binding sites of three transcription factors.

Liu et al. (2008) use the occurrences of A/T-dinucleotides and the computationally determined curvature of the DNA helix as input of a wavelet analysis to predict nucleosome positions in DNA sequences stemming from chromosome 1 and 2 of yeast. In accordance to Segal et al. (2006), they find periodic signals of A/T dinucleotides along the nucleosomes.

Wavelets are also employed by Yuan and Liu (2008), who use a wavelet transform with Haar wavelets. The wavelet coefficients are computed on each of 16 binary sequences representing dinucleotide occurrences for each of the 16 dinucleotide. An automatically selected subset of these coefficients serves as input of a logistic regression (see section 3.2). The final assignment of nucleosome positions is accomplished by an HMM working on the probabilities that are the results of logistic regression. This HMM accounts for steric hindrance between adjacent nucleosomes.

All of the previous approaches derive their ground truth from the analysis of tiling microarrays or low-throughput sequencing. Field et al. (2008) are the first, who use MNase digest in conjunction with high-throughput parallel sequencing to obtain $\sim 380,000$ uniquely mapped nucleosome positions. They train a slightly modified version of the model of Segal et al. (2006) using a homogeneous Markov model of order 4 on these data and show that their model performs best compared to the approaches of (Ioshikhes et al., 2006), (Segal et al., 2006),

(Lee et al., 2007), (Peckham et al., 2007), and (Yuan and Liu, 2008). The data of (Field et al., 2008) are also considered in this work. In (Lubliner and Segal, 2009) this approach is combined with an explicit model of dependencies between adjacent nucleosomes and the authors show that prediction performance can be improved for *in vitro* as well as *in vivo* nucleosome positioning.

Morozov et al. (2009) employ a biophysical model for predicting DNA geometry within nucleosomes and use this model to predict genome-wide nucleosome occupancy as well. The geometrical properties modelled are twist, roll, tilt, slide, shift, and rise (see also section 4.3.2.6), and they show that their predictions are highly correlated with experimental results using the first four properties.

4.3.2. Model

The approach presented here is driven by two main ideas: Firstly, coding and non-coding sequences differ in general properties like G/C-content or the presence of coding potential, which might superimpose the signals of nucleosome positioning. Hence, nucleosome positioning in coding and non-coding regions should be modelled separately. Secondly, nucleosome positioning might be influenced by a number of different features of the affected stretches of DNA, which all contribute to the final probability of nucleosome formation. Additionally, we utilize preferred lengths of the linker sequences between nucleosome in a post-processing step.

4.3.2.1. Voting of components

We model the differentiation between coding and non-coding sequences by a two-stage process. First, we learn a classifier that discriminates coding from non-coding sequences. Second, we learn independent *component classifiers* for discriminating DNA bound in nucleosomes from linkers given that this stretch of DNA is either coding or non-coding. In order to prevent a blurring of the distinctive properties of nucleosome-bound sequences in coding and non-coding DNA, we introduce another category called *border*, which refers to stretches of DNA that are at a border between coding and non-coding regions and thus exhibit properties of both types of DNA.

The full setup is depicted in figure 4.20. The colored boxes on the left side represent the three component classifiers that model the probability of nucleosome formation given sequence $\boldsymbol{x}$ is coding (c, blue box), non-coding (n, green box), or at a border region (b, red box). Each component classifier comprises a number of *elementary classifiers*, which are represented by the inner boxes enclosed in the boxes of the component classifiers.

In the component classifier for region $m \in \{c, n, b\}$, the votings $P(\text{nuc}|\boldsymbol{x}, \text{m}, t, \boldsymbol{\beta}_{m,t})$ of the set $\mathcal{T}_m$ of elementary classifiers are combined by weighted voting. The weights $P(t|\boldsymbol{\beta}_m)$ of the elementary classifiers are a-priori probabilities and do not depend on the sequence $\boldsymbol{x}$. Each elementary classifier considers a feature or a set of closely related features of a sequence. The specific elementary classifiers in $\mathcal{T}_m$ and the number of elementary classifiers T_m may be different for each of the component classifiers. The parameters $\boldsymbol{\beta}_m$ of component classifier m

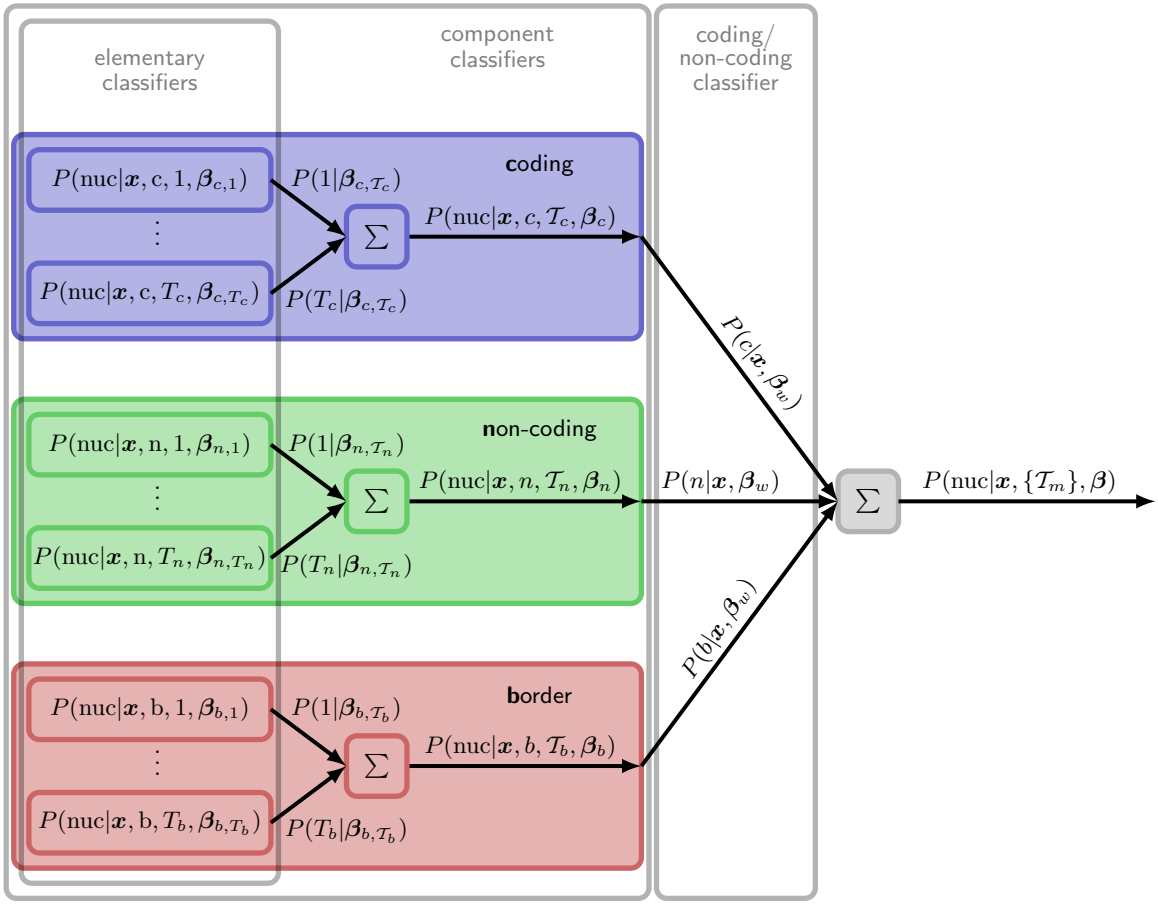


Figure 4.20.: Voting of components. The three component classifiers for coding, non-coding and border sequences are represented by blue, green, and red boxes, respectively. Each component classifier consists of a number of elementary classifiers, which are illustrated by inner boxes enclosed in the boxes of the component classifiers. The elementary classifiers employ specific features of the DNA sequence $\boldsymbol{x}$. The t-th elementary classifier uses its features to compute the probability $P(\text{nuc}|\boldsymbol{x}, m, t, \boldsymbol{\beta}_{m,t})$ of nucleosome formation given region m. These are combined by weighted voting using a-priori weights $P(t|\boldsymbol{\beta}_{m,\mathcal{T}_m})$ yielding the voting $P(\text{nuc}|\boldsymbol{x}, m, \mathcal{T}_m, \boldsymbol{\beta}_m)$ of the component classifier given the elementary classifiers in set $\mathcal{T}_m$. These votings are combined by another weighted voting using the probabilities $P(m|\boldsymbol{x}, \boldsymbol{\beta}_w)$ as weights, which are the probabilities that $\boldsymbol{x}$ is a coding, non-coding, or border sequence. Finally, we obtain the probability $P(\text{nuc}|\boldsymbol{x}, \{\mathcal{T}_m\}, \boldsymbol{\beta})$ of nucleosome formation given the sets $\{\mathcal{T}_m\}$ of selected elementary classifiers.

comprise all sets of parameters $\boldsymbol{\beta}_{m,t}$ of the employed elementary classifiers and the parameters for the a-priori probabilities of these elementary classifiers.

The probabilities $P(\text{nuc}|\boldsymbol{x}, m, \mathcal{T}_m, \boldsymbol{\beta}_m), m \in \{c, n, b\}$, of nucleosome formation according to the component classifiers are combined by another weighted voting. In contrast to the weighted voting of elementary classifiers, the probabilities $P(m \,|\, \boldsymbol{x}, \boldsymbol{\beta}_w)$ used as weights depend on the sequence. These correspond to the probabilities that $\boldsymbol{x}$ is a coding, non-coding, or border sequence, and are obtained from the classifier discriminating these three classes, where $\boldsymbol{\beta}_w$ denotes the parameters of this classifier. We define the combined probability of nucleosome

formation $P(\text{nuc} \mid \boldsymbol{x}, \{\mathcal{T}_m\}, \boldsymbol{\beta})$ as

$$P(\text{nuc} \mid \boldsymbol{x}, \{\mathcal{T}_m\}, \boldsymbol{\beta}) = \sum_{m \in \{c,n,b\}} P(m \mid \boldsymbol{x}, \boldsymbol{\beta}_w) P(\text{nuc} \mid \boldsymbol{x}, m, \mathcal{T}_m, \boldsymbol{\beta}_m), \tag{4.18}$$

where $\boldsymbol{\beta} = (\boldsymbol{\beta}_w, \boldsymbol{\beta}_c, \boldsymbol{\beta}_n, \boldsymbol{\beta}_b)$. This voting scheme shows some analogies to boosting (Freund and Schapire, 1996; Jing et al., 2005) and other ensemble approaches (Cerquides and de Mántaras, 2005; Kim and Pavlovic, 2005). However, in contrast to previous approaches the weights $P(m \mid \boldsymbol{x}, \boldsymbol{\beta}_w)$ of the voting depend on the current sequence $\boldsymbol{x}$. In the following, we consider sequences $\boldsymbol{x}$ of length $L = 200$.

We learn the parameters $\boldsymbol{\beta}_m$ of the component classifiers and the parameters $\boldsymbol{\beta}_w$ of the classifier discriminating coding, non-coding, and border sequences by the discriminative MSP principle. In case of the component classifiers, the classes considered are nucleosome-bound sequences and linker sequences, and the corresponding class posteriors required for the definition of conditional likelihood (see section 3.2.1, p. 12) are $P(\text{nuc}|\boldsymbol{x}, m, \mathcal{T}_m, \boldsymbol{\beta}_m)$ and $1-P(\text{nuc}|\boldsymbol{x}, m, \mathcal{T}_m, \boldsymbol{\beta}_m)$, respectively. For the classifier discriminating coding, non-coding, and border sequences, we consider these very classes, and the class posteriors correspond to $P(m \mid \boldsymbol{x}, \boldsymbol{\beta}_w), m \in \{c, n, b\}$.

The MSP principle also requires the definition of priors on the parameters $\boldsymbol{\beta}_m$ and $\boldsymbol{\beta}_w$. The specific priors used for the parameters of the elementary classifiers, the parameters of the weights used in the component classifiers, and the parameters of the classifier discriminating coding, non-coding, and border sequences are presented in the following sections, where we also concretize these classifiers. For the specification of the hyper-parameters of all priors, we adhere to the assumption of uniform pseudo-data. In most cases, we can determine hyper-parameters according to this assumption analytically, while in some cases we must resort to simulations.

The remainder of this section is structured as follows. In the next sub-section, we present the classifier that discriminates coding, non-coding, and border sequences. We give more detail on the weighted voting of elementary classifiers employed by the component classifiers in sub-section 4.3.2.3. We introduce the elementary classifiers employing Markov models in sub-section 4.3.2.4 and 4.3.2.5, and those employing numerical properties of DNA sequences in sub-section 4.3.2.6. For learning the component classifiers, we map coverage by nucleosome reads to probabilities of nucleosome formation as described in sub-section 4.3.2.7 and we select elementary classifiers by a greedy approach presented in sub-section 4.3.2.8. Finally, we describe the post-processing step for utilizing preferred linker lengths in sub-section 4.3.2.9.

4.3.2.2. Discriminating coding from non-coding sequences

Discriminating coding from non-coding sequences is closely related to gene finding, which involves the prediction of transcription starts or exon-intron boundaries aside from more general properties like base composition, codon usage, or the presence of stop codons. Here, we concentrate on the latter properties, because these utilize neither the location of the sequence on the chromosome nor its neighborhood, and, hence, can be employed for isolated, short (200 bp) sequences that we consider for the prediction of nucleosome positioning.

Class posterior

We define the class posterior $P(m \mid \boldsymbol{x}, \boldsymbol{\beta}_w)$ based on the a-priori probability $P(m|\boldsymbol{\beta}_w)$ of class m, where $m \in \{c, n, b\}$, and a score $S(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$ of sequence $\boldsymbol{x}$ given class m and the parameters $\boldsymbol{\beta}_w$ as

$$P(m \mid \boldsymbol{x}, \boldsymbol{\beta}_w) = \frac{P(m|\boldsymbol{\beta}_w)S(\boldsymbol{x}|m, \boldsymbol{\beta}_w)}{\sum_{\tilde{m}} P(\tilde{m}|\boldsymbol{\beta}_w)S(\boldsymbol{x}|\tilde{m}, \boldsymbol{\beta}_w)}. \tag{4.19}$$

The score $S(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$ can be normalized to a proper likelihood. However, normalization is not necessary and can be omitted, since the class posterior as defined in equation (4.19) is always normalized, i.e. $\sum_{m\in\{c,n,b\}} P(m \mid \boldsymbol{x}, \boldsymbol{\beta}_w) = 1$.

We parameterize the a-priori probabilities as

$$P(m|\boldsymbol{\beta}_w) = \frac{\exp(\beta_m)}{\sum_{\tilde{m}} \exp(\beta_{\tilde{m}})}, \tag{4.20}$$

where the $\beta_m \in \mathbb{R}$ are a subset of $\boldsymbol{\beta}_w$. We omit the index w here and in the following, when we refer to the parameters for one specific class to reduce notational complexity.

The score $S(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$ is composed of the likelihood $P_{\text{Strand}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$ that basically models base composition and codon usage, and a score $S_{\text{Stop}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$ that models the distribution of stop codons over the potential reading frames:

$$S(\boldsymbol{x}|m, \boldsymbol{\beta}_w) = P_{\text{Strand}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)S_{\text{Stop}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w), \tag{4.21}$$

As we will see in a few paragraphs, the distribution of stop codons is a proper probability distribution over the number of reading frames exhibiting a stop codon, but not over all possible sequences $\boldsymbol{x} \in \Sigma^L$ and, hence, referred to as a score. Although the independence assumption, implied by using the product of the likelihood $P_{\text{Strand}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$ and the score $S_{\text{Stop}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$, is clearly not valid in general, we consider the degree of dependency low and thus neglectable.

Base composition in non-coding sequences should not depend on the strand considered. Additionally, we do not consider the annotated strand orientation of the genes from that we extract coding sequences. Hence, we utilize a *strand model* for the first likelihood $P_{\text{Strand}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$, which is a mixture model over the forward and backward strand applying the same component likelihood once to the original sequence $\boldsymbol{x}$ and once to its reverse complement $\boldsymbol{x}^{rc}$, i.e.

$$P_{\text{Strand}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w) = P(\text{fw}|\boldsymbol{\beta}_w)P_{\text{MM}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w) + P(\text{bw}|\boldsymbol{\beta}_w)P_{\text{MM}}(\boldsymbol{x}^{rc}|m, \boldsymbol{\beta}_w), \tag{4.22}$$

where the index MM indicates that we employ Markov models for the component likelihood $P_{\text{MM}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$ as described in the next paragraph. We parameterize the mixture probabilities in terms of real valued parameters $\beta_{\text{fw}|m}, \beta_{\text{bw}|m} \in \mathbb{R}$ as

$$P(\text{fw}|\boldsymbol{\beta}_w) = \frac{\exp(\beta_{\text{fw}|m})}{\exp(\beta_{\text{fw}|m}) + \exp(\beta_{\text{bw}|m})} \text{ and } \quad P(\text{bw}|\boldsymbol{\beta}_w) = \frac{\exp(\beta_{\text{bw}|m})}{\exp(\beta_{\text{fw}|m}) + \exp(\beta_{\text{bw}|m})}, \tag{4.23}$$

where $\beta_{\text{fw}|m}$ and $\beta_{\text{bw}|m}$ are again included in $\boldsymbol{\beta}_w$.

For the component likelihood $P_{\mathrm{MM}}(\boldsymbol{x}|m,\boldsymbol{\beta}_w)$, we split the sequence $\boldsymbol{x}$ in halves to adequately model sequences stemming from the border region, which comprise coding as well as non-coding parts with different properties. Each of the two halves is modelled by a mixture of a homogeneous Markov model of order 3 (hMM) and a 3-periodic Markov model of order 3 (pMM) as introduced in section 3.3.1 (p. 17), resulting in the combined likelihood

$$\begin{aligned} P_{\mathrm{MM}}(\boldsymbol{x}|m,\boldsymbol{\beta}_w) \;=\; & \big[P_1(\mathrm{hMM}|\boldsymbol{\beta}_w)P_{\mathrm{hMM},1}(x_1,\ldots,x_{L/2}|m,\boldsymbol{\beta}_w)+ \\ & \quad P_1(\mathrm{pMM}|\boldsymbol{\beta}_w)P_{\mathrm{pMM},1}(x_1,\ldots,x_{L/2}|m,\boldsymbol{\beta}_w)\big]\cdot \\ & \big[P_2(\mathrm{hMM}|\boldsymbol{\beta}_w)P_{\mathrm{hMM},2}(x_{L/2+1},\ldots,x_L|m,\boldsymbol{\beta}_w)+ \\ & \quad P_2(\mathrm{pMM}|\boldsymbol{\beta}_w)P_{\mathrm{pMM},2}(x_{L/2+1},\ldots,x_L|m,\boldsymbol{\beta}_w)\big]\,, \end{aligned} \tag{4.24}$$

where the mixture probabilities are parameterized in analogy to those of the strand model in terms of parameters $\beta_{\mathrm{hMM},i|m},\beta_{\mathrm{pMM},i|m}\in\mathbb{R}$ as

$$P_i(\mathrm{hMM}|\boldsymbol{\beta}_w)=\frac{\exp(\beta_{\mathrm{hMM},i|m})}{\exp(\beta_{\mathrm{hMM},i|m})+\exp(\beta_{\mathrm{pMM},i|m})} \tag{4.25}$$

$$P_i(\mathrm{pMM}|\boldsymbol{\beta}_w)=\frac{\exp(\beta_{\mathrm{pMM},i|m})}{\exp(\beta_{\mathrm{hMM},i|m})+\exp(\beta_{\mathrm{pMM},i|m})}. \tag{4.26}$$

The 3-periodic Markov model explicitly represents the codon structure of coding sequences, where the probability of a certain nucleotide x_ℓ at position ℓ depends on its localization within the codon.

We anticipate that the mixture probabilities $P_i(\mathrm{hMM}|\boldsymbol{\beta}_w)$ and $P_i(\mathrm{pMM}|\boldsymbol{\beta}_w)$ adapt to the characteristics of coding, non-coding, and border sequences. For coding sequences, we expect a high probability for the periodic Markov model in both halves, whereas the homogeneous Markov model should obtain a higher probability for non-coding sequences. For sequences at the borders between coding and non-coding regions, the model should favor the periodic Markov model in one of the halves and the homogeneous Markov model in the other. Due to the enclosing strand model, it does not matter, which of the two models is favored for which of the two halves.

As second component of the score $S(\boldsymbol{x}|m,\boldsymbol{\beta}_w)$ we use a model for the distribution of stop codons (TAA, TGA, and TAG) over the three reading frames on the forward strand and the three reading frames on the backward strand. It is known (Nicorici and Astola, 2004; Creanza et al., 2009) that the number of reading frames containing at least one stop codon discriminates well between coding and non-coding sequences: Considering one strand of coding sequences of length 160, Nicorici and Astola (2004) find that with high probability (~ 0.97) only two of the three potential reading frames do contain a stop codon, whereas for non-coding sequences of the same length, they observe stop codons in all three reading frames with a probability of ~ 0.86. These observations can be explained by the fact that only one stop codon exists in the correct frame, which may not be contained in the 160 bp considered, whereas the number of stop codons is not controlled in the two incorrect reading frames. Stop codons in the incorrect reading frames may even be favorable, as they stop translation in a wrong frame early. Finally, the occurrence of stop codons in non-coding sequences depends only on the base composition and the length of the sequence, as stop codons are meaningless in non-coding DNA.

Let $n_{\text{Stop}}(\boldsymbol{x})$ be the number of reading frames of sequence $\boldsymbol{x}$ on either of the two strands that contain at least one stop codon. We define the score of sequence $\boldsymbol{x}$ with respect to the number of stop codons as

$$S_{\text{Stop}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w) \quad = \quad P(n_{\text{Stop}}(\boldsymbol{x})|m, \boldsymbol{\beta}_w), \tag{4.27}$$

where $P(n_{\text{Stop}}(\boldsymbol{x})|m, \boldsymbol{\beta}_w)$ is a proper probability distribution over the number of reading frames that contain at least one stop codon, i.e. $\sum_{n=1}^{6} P(n|m, \boldsymbol{\beta}_w) = 1$, but is not normalized over all sequences $\boldsymbol{x} \in \Sigma^L$. Hence, it is only considered a score $S_{\text{Stop}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$ with regard to a sequence $\boldsymbol{x}$. The probability of observing at least one stop codon in n of the six potential reading frames is parameterized as

$$P(n|m, \boldsymbol{\beta}_w) = \frac{\exp(\beta_{\text{Stop},n|m})}{\sum_{\tilde{n}=1}^{6} \exp(\beta_{\text{Stop},\tilde{n}|m})}, \tag{4.28}$$

where the $\beta_{\text{Stop},n|m} \in \mathbb{R}$ are a subset of the parameters in $\boldsymbol{\beta}_w$.

We learn all parameters of the classifier distinguishing coding, non-coding, and border sequences including the parameters of the a-priori probabilities of the three classes, the mixture parameters, the parameters of the homogeneous and periodic Markov models, and the parameters of the score modeling stop codons. We optimize the parameters $\boldsymbol{\beta}_w$ by the discriminative MSP principle, i.e.

$$\boldsymbol{\beta}_w^* \quad = \quad \underset{\boldsymbol{\beta}_w}{\text{argmax}} \left[\prod_{n=1}^{N} P\left(c_n \,|\, \boldsymbol{x}_n, \boldsymbol{\beta}_w\right) \right] q\left(\boldsymbol{\beta}_w \,|\, \boldsymbol{\alpha}_w\right), \tag{4.29}$$

where $c_n \in \{c, n, b\}$ denotes the correct class of sequence $\boldsymbol{x}_n$, and $q\left(\boldsymbol{\beta}_w \,|\, \boldsymbol{\alpha}_w\right)$ denotes the prior on the parameters $\boldsymbol{\beta}_w$ with hyper-parameters $\boldsymbol{\alpha}$, which has yet to be defined.

Prior and hyper-parameters

We start the definition of prior and hyper-parameters with the definition of equivalent sample sizes (ESS) α_m (cf. section 3.4) for the three classes. These are used as hyper-parameters of a transformed Dirichlet prior (see section 3.4.2, p. 30) on the parameters of the a-priori class probabilities β_m. The same ESS is used for the parameters of the strand model $P_{\text{Strand}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$ and the distribution of stop codons $S_{\text{Stop}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$. For the latter, we use another transformed Dirichlet prior, and we equally distribute the ESS over the six possible outcomes resulting in hyper-parameters $\alpha_{\text{Stop},n|m} = \frac{\alpha_m}{6}$ for parameter $\beta_{\text{Stop},n|m}$.

The assumption of uniformly distributed pseudo data also leads to an even distribution of the ESS over the two strands, resulting in hyper-parameters $\alpha_{\text{fw}|m} = \alpha_{\text{bw}|m} = \frac{\alpha_m}{2}$, which are used in a transformed Beta prior on the parameters $\beta_{\text{fw}|m}$ and $\beta_{\text{bw}|m}$. As the mixture of the Markov models $P_{\text{MM}}(\boldsymbol{x}|m, \boldsymbol{\beta}_w)$ is used for both strands, once for the original sequence $\boldsymbol{x}$ and once for its reverse complement $\boldsymbol{x}^{rc}$, the ESS remains α_m for the mixture. This ESS is then evenly distributed over the two components of the mixture, and we use hyper-parameters $\alpha_{\text{hMM},i|m} = \alpha_{\text{pMM},i|m} = \frac{\alpha_m}{2}$ for another transformed Beta prior on the mixture parameters $\beta_{\text{hMM},i|m}$ and $\beta_{\text{pMM},i|m} = \frac{\alpha_m}{2}$.

The priors on the parameters of the homogeneous and periodic Markov models are product-Dirichlet priors defined in section 3.4.2 using an ESS of $\frac{\alpha_m}{2}$ and an expected length of $L_E = 200$.

The complete prior $q\left(\boldsymbol{\beta}_w \,|\, \boldsymbol{\alpha}_w\right)$ is then the product of all the component priors described above. Throughout the experiments, we use $\alpha_m = 4$.

4.3.2.3. Component classifiers

As mentioned in the beginning of this section, we assume that different features contribute to nucleosome positioning. We represent this assumption by defining the component classifiers as a weighted voting of *elementary classifiers*, each considering a specific feature or a set of closely related features of a sequence. In contrast to the enclosing voting of components, the weights applied to the votings of elementary classifiers are a-priori weights, i.e. do not depend on the sequence, in this case. The elementary classifiers are selected by a greedy algorithm presented in section 4.3.2.8 and the set of elementary classifiers $\mathcal{T}_m$ selected may be different for coding, non-coding, and border sequences.

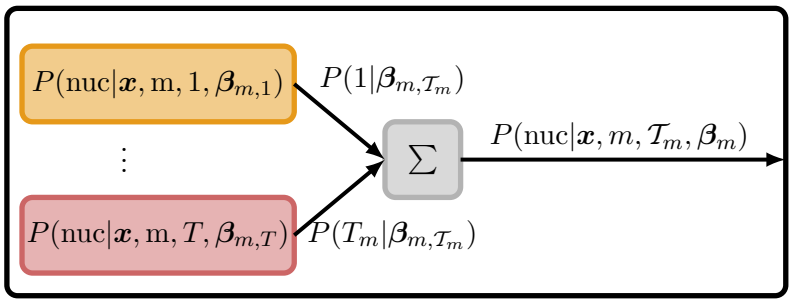


Figure 4.21.: Weighted voting of elementary classifiers used as component classifiers (cf. figure 4.20). The votings $P\left(\text{nuc} \,|\, \boldsymbol{x}, m, t, \boldsymbol{\beta}_{m,t}\right)$ of elementary classifiers are weighted by $P(t|\boldsymbol{\beta}_{m,\mathcal{T}_m})$ and added to the final voting $P\left(\text{nuc} \,|\, \boldsymbol{x}, m, \mathcal{T}_m, \boldsymbol{\beta}_m\right)$ of the component classifier.

A graphical representation of the voting is given in figure 4.21, which corresponds to one of the boxes representing component classifiers in figure 4.20. The t-th elementary classifier in $\mathcal{T}_m$ votes with probability $P\left(\text{nuc} \,|\, \boldsymbol{x}, m, t, \boldsymbol{\beta}_{m,t}\right)$ for $\boldsymbol{x}$ being bound in a nucleosome. The votings of the elementary classifiers are weighted by a-priori probabilities $P(t|\boldsymbol{\beta}_{m,\mathcal{T}_m})$ and combined to a final probability of nucleosome formation $P\left(\text{nuc} \,|\, \boldsymbol{x}, m, \mathcal{T}_m, \boldsymbol{\beta}_m\right)$ given the sequence $\boldsymbol{x}$, the type of the component $m \in \{c, n, b\}$, and parameters $\boldsymbol{\beta}_m$. We formalize this weighted voting as

$$P\left(\text{nuc} \,|\, \boldsymbol{x}, m, \mathcal{T}_m, \boldsymbol{\beta}_m\right) \;=\; \sum_{t \in \mathcal{T}_m} P(t|\boldsymbol{\beta}_{m,\mathcal{T}_m}) P\left(\text{nuc} \,|\, \boldsymbol{x}, m, t, \boldsymbol{\beta}_{m,t}\right), \tag{4.30}$$

where $\boldsymbol{\beta}_m = (\boldsymbol{\beta}_{m,\mathcal{T}_m}, \boldsymbol{\beta}_{m,1}, \ldots, \boldsymbol{\beta}_{m,T_m})$. The probabilities used as weights are parameterized in terms of parameters $\beta_{t|m} \in \mathbb{R}$ as

$$P(t|\boldsymbol{\beta}_{m,\mathcal{T}_m}) = \frac{\exp(\beta_{t|m})}{\sum_{\tilde{t} \in \mathcal{T}_m} \exp(\beta_{\tilde{t}|m})}, \tag{4.31}$$

and we denote by $\boldsymbol{\beta}_{m,\mathcal{T}_m}$ the vector of all $\beta_{t|m}$. Again, we apply a transformed Dirichlet prior to the parameters $\boldsymbol{\beta}_{m,\mathcal{T}_m}$ using an ESS of α_m, which is equally distributed over the T_m hyper-parameters $\alpha_{t|m}$.

The class posterior $P\left(\text{nuc} \mid \boldsymbol{x}, m, t, \boldsymbol{\beta}_{m,t}\right)$ given sequence $\boldsymbol{x}$ of type m in the elementary classifier t with parameters $\boldsymbol{\beta}_{m,t}$ is defined as

$$P\left(\text{nuc} \mid \boldsymbol{x}, m, t, \boldsymbol{\beta}_{m,t}\right) = \frac{P(\text{nuc}|\boldsymbol{\beta}_{m,t}) S_{m,t}(\boldsymbol{x}|\text{nuc}, \boldsymbol{\beta}_{\text{nuc},m,t})}{P(\text{nuc}|\boldsymbol{\beta}_{m,t}) S_{m,t}(\boldsymbol{x}|\text{nuc}, \boldsymbol{\beta}_{\text{nuc},m,t}) + P(\text{link}|\boldsymbol{\beta}_{m,t}) S_{m,t}(\boldsymbol{x}|\text{link}, \boldsymbol{\beta}_{\text{link},m,t})}, \tag{4.32}$$

where $P(\text{nuc}|\boldsymbol{\beta}_{m,t})$ denotes the a-priori probability of nucleosome formation parameterized in analogy to equation (4.20) and $\boldsymbol{\beta}_{m,t} = (\beta_{\text{nuc}|m,t}, \beta_{\text{link}|m,t}, \boldsymbol{\beta}_{\text{nuc},m,t}, \boldsymbol{\beta}_{\text{link},m,t})$. $S_{m,t}(\boldsymbol{x}|\text{nuc}, \boldsymbol{\beta}_{\text{nuc},m,t})$ and $S_{m,t}(\boldsymbol{x}|\text{link}, \boldsymbol{\beta}_{\text{link},m,t})$ are scores of sequence $\boldsymbol{x}$ given it is a nucleosome-bound or linker sequence, respectively. The functional form of these scores depends on the sequence features that are used in elementary classifier t. We consider two fundamental types of scores. The first is composed of Markov models capturing general k-mer frequencies, and is actually a proper likelihood over the sequences $\boldsymbol{x} \in \Sigma^L$. The second type comprises the densities of several numerical sequence features.

4.3.2.4. Homogeneous Markov models

We illustrate in figure 4.22 how homogeneous Markov models are arranged in the elementary classifiers. We divide the input sequence of length $L = 200$ into four quarters, where the first quarter is modelled by a homogeneous Markov model of order d_1, the second quarter by a homogeneous Markov model of order d_2, which is re-used for the reverse complement of the third quarter as indicated by the inverse direction of the second green box, and, finally, the fourth quarter is modelled by another homogeneous Markov model of the same order as the first one. The Markov model in the center depicted as green box scores 100 bp in total, which are assumed to be located at the core of nucleosome-bound sequences. The two Markov models at the borders depicted in red and blue, respectively, model the transition from nucleosome-bound sequence to linker, which should be localized approximately in these 50 bp regions. By using different Markov models with different parameters at the two borders, we allow for variability in this transition. The combination of homogeneous Markov models is enclosed in a strand model, which allows for inversion of the sequence of Markov models and is visualized by the outer grey box in figure 4.22

Score We formalize the strand model using a combination of homogeneous Markov models with order d_1 at the borders and order d_2 in the center given class $c \in \{\text{nuc}, \text{link}\}$ and parameters $\boldsymbol{\beta}_{m,t}, m \in \{c, n, b\}$, as

$$S_{m,t}^{\text{hMM}(d_2,d_1)}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,m,t}) = P(\text{fw}|c, \boldsymbol{\beta}_{c,m,t}) P_{\text{hMM}}(\boldsymbol{x}|c, d_2, d_1, \boldsymbol{\beta}_{c,m,t}) + \tag{4.33}$$

$$P(\text{bw}|c, \boldsymbol{\beta}_{c,m,t}) P_{\text{hMM}}(\boldsymbol{x}^{rc}|c, d_2, d_1, \boldsymbol{\beta}_{c,m,t}), \tag{4.34}$$

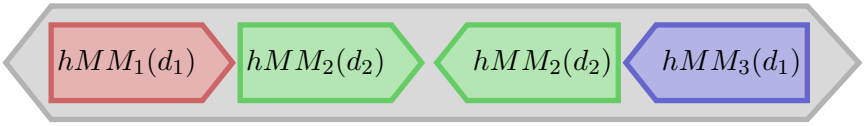


Figure 4.22.: Arrangement of homogeneous Markov models in the elementary classifier. The first and the last Markov model share the same order but may employ different probability distributions as indicated by the differing colors. The Markov model in the center is applied once to the second quarter to the sequence and once to the reverse complement of the third quarter, which is represented by the orientation of boxes. The combination of these three Markov models is enclosed in a strand model, which is illustrated by the outer grey box.

where $\boldsymbol{\beta}_{c,m,t}$ denotes the subset of parameters in $\boldsymbol{\beta}_{m,t}$ that are used for class c and the mixture probabilities of the strand model $P(\text{fw}|c, \boldsymbol{\beta}_{c,m,t})$ and $P(\text{bw}|c, \boldsymbol{\beta}_{c,m,t})$ are parameterized in analogy to equation (4.23). The combination of homogeneous Markov models amounts to the product of the corresponding likelihoods parameterized according to section 3.3.1 and applied to the first quarter of the sequence $x_1, \ldots, x_{L/4}$, the second quarter $x_{L/4+1}, \ldots, x_{L/2}$, and the reverse complement of the third quarter $[x_{L/2+1}, \ldots, x_{3L/4}]^{rc}$ and the fourth quarter $[x_{3L/4+1}, \ldots, x_L]^{rc}$, i.e.

$$P_{\text{hMM}}(\boldsymbol{x}|c, d_2, d_1, \boldsymbol{\beta}_{c,m,t}) = P_{\text{hMM}_1(d_1)}(x_1, \ldots, x_{L/4}|\boldsymbol{\beta}_{c,m,t,1}) P_{\text{hMM}_2(d_2)}(x_{L/4+1}, \ldots, x_{L/2}|\boldsymbol{\beta}_{c,m,t,2}) \cdot P_{\text{hMM}_2(d_2)}([x_{L/2+1}, \ldots, x_{3L/4}]^{rc}|\boldsymbol{\beta}_{c,m,t,2}) P_{\text{hMM}_3(d_1)}([x_{3L/4+1}, \ldots, x_L]^{rc}|\boldsymbol{\beta}_{c,m,t,3}), \quad (4.35)$$

where $\boldsymbol{\beta}_{c,m,t,k}$ denotes the subset of parameters in $\boldsymbol{\beta}_{c,m,t}$ that are used for the homogeneous Markov model $k \in \{1, 2, 3\}$. Here, we consider orders $d_2 \in \{0, 1, 2, 3\}$ at the center and orders $d_1 \in \{0, \ldots, d_2\}$ at the borders, i.e. the order of the Markov models at the borders is at most the order of the Markov model at the center.

Prior and hyper-parameters We use a transformed beta prior on the parameters of the mixture probabilities of the strand model. As hyper-parameters, we use $\alpha_{\text{fw}|c,m,t} = \alpha_{\text{bw}|c,m,t} = \frac{\alpha_m}{T \cdot 2 \cdot 2}$, where T is the number of elementary classifiers in the component classifier m. This choice of hyper-parameters follows from the assumption of uniform pseudo data (see section 3.4.2) and assumes a uniform a-priori distribution of nucleosome-bound sequences and linkers as well as the two strand orientations. Accordingly, we set the equivalent sample sizes of the homogeneous Markov models to $\frac{\alpha_m}{T \cdot 2}$ at the borders, since this models are used for both strands, and to $\frac{\alpha_m}{T}$ for the homogeneous Markov model at the center, which is used for both strands and the second and third quarter of the input sequence. We set the expected length (see section 3.4.2, p. 30) of all homogeneous Markov models to $L_E = \frac{L}{4}$.

4.3.2.5. Inhomogeneous Markov model

Score In addition to the homogeneous Markov models, we define another score that uses an inhomogeneous Markov model of order 1 (see section 3.3.1) for the class of nucleosome-bound sequences, and a homogeneous Markov model of order 4 for modelling the linker sequences, which is similar to the heuristically learned model employed by (Field et al., 2008). We denote

this score as

$$S_{m,t}^{\text{iMM}}(\boldsymbol{x}|c,\boldsymbol{\beta}_{c,m,t}) = \begin{cases} P_{iMM(1)}(\boldsymbol{x}|\text{nuc},\boldsymbol{\beta}_{c,m,t}), & \text{if} \quad c = \text{nuc} \\ P_{hMM(4)}(\boldsymbol{x}|\text{link},\boldsymbol{\beta}_{c,m,t}), & \text{if} \quad c = \text{link}. \end{cases} \tag{4.36}$$

Prior and hyper-parameters We use two transformed product-Dirichlet priors with equivalent sample size $\frac{\alpha_m}{T \cdot 2}$ as defined in section 3.4.2 for the parameters of the inhomogeneous Markov model and the parameters of the homogeneous Markov model.

4.3.2.6. Numerical properties of DNA sequences

Besides homogeneous Markov models, we also employ a number of numerical properties computed from the sequence $\boldsymbol{x}$, some of which are motivated by previous findings of specific properties of nucleosome-bound sequences and linkers. These properties include the entropy of k-mer frequencies, the number of CTG trinucleotides, the length of consecutive tracts of A or T nucleotides, and the wavelet energies of geometrical and physicochemical properties of the DNA helix. For each of the properties introduced in the following, we first describe how the property is determined from DNA sequence. We then define the score for modeling the property, and finally introduce the prior on the parameters of the score and the associated hyper-parameters under the assumption of uniform pseudo data.

Entropy

Entropy measures the deviation of a given probability distribution from the uniform distribution. Here, we apply it to the relative frequencies of k-mer occurrences, which gives an overall rating of the over- or under-representation of k-mers. The entropy $H_k(\boldsymbol{x})$ estimated from the relative frequencies of k-mers in sequence $\boldsymbol{x}$ amounts to

$$H_k(\boldsymbol{x}) \quad = \quad - \sum_{\boldsymbol{b} \in \Sigma^k} \frac{n_{\boldsymbol{b}}(\boldsymbol{x})}{\sum_{\tilde{\boldsymbol{b}}} n_{\tilde{\boldsymbol{b}}}(\boldsymbol{x})} \log \left(\frac{n_{\boldsymbol{b}}(\boldsymbol{x})}{\sum_{\tilde{\boldsymbol{b}}} n_{\tilde{\boldsymbol{b}}}(\boldsymbol{x})} \right), \tag{4.37}$$

where $n_{\boldsymbol{b}}(\boldsymbol{x})$ denotes the number of occurrences of k-mer $\boldsymbol{b}$ in sequence $\boldsymbol{x}$. Here, we consider the entropies of 1- to 4-mers.

Additionally, we consider the entropy of k-mers for a reduced alphabet that identifies nucleotides and their complements, i.e. we define a symbol W that matches the nucleotides A and T in $\boldsymbol{x}$, and another symbol S that matches G and C. The estimated entropy of k-mers for this reduced alphabet amounts to

$$H_k^c(\boldsymbol{x}) \quad = \quad - \sum_{\boldsymbol{b} \in \{W,S\}^k} \frac{n_{\boldsymbol{b}}(\boldsymbol{x})}{\sum_{\tilde{\boldsymbol{b}}} n_{\tilde{\boldsymbol{b}}}(\boldsymbol{x})} \log \left(\frac{n_{\boldsymbol{b}}(\boldsymbol{x})}{\sum_{\tilde{\boldsymbol{b}}} n_{\tilde{\boldsymbol{b}}}(\boldsymbol{x})} \right). \tag{4.38}$$

Scores As the entropy is always positive, we model the distribution of values of the entropies by transformed gamma densities (see section 3.3.2.2) with shape $\gamma_{c,m,t,k}$ and rate $\beta_{c,m,t,k}$. The index t indicates here and in the following that these parameters are used in one of the elementary classifiers indexes by t.

We define the score of sequence $\boldsymbol{x}$ with respect to the entropy of k-mers as

$$S^H_{m,t}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,m,t}) = \prod_{k=1}^{4} \mathcal{G}(H_k(\boldsymbol{x})|\gamma_{c,m,t,k}, \beta_{c,m,t,k}), \tag{4.39}$$

where in this case $\boldsymbol{\beta}_{c,m,t} = (\gamma_{c,m,t,1}, \beta_{c,m,t,1}, \ldots, \gamma_{c,m,t,4}, \beta_{c,m,t,4})$.
Although each single density $\mathcal{G}(H_k(\boldsymbol{x})|\gamma_{c,m,t,k}, \beta_{c,m,t,k})$ is normalized, $S^H_{m,t}(\boldsymbol{x}|c, \boldsymbol{\beta}_{m,t})$ is not normalized over all possible input sequences $\boldsymbol{x}$ and hence referred to as a score.

We define the score of sequence $\boldsymbol{x}$ with respect to the entropy for the reduced alphabet as

$$S^{H^c}_{m,t}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,m,t}) = \prod_{k=1}^{4} \mathcal{G}(H^c_k(\boldsymbol{x})|\gamma_{c,m,t,k}, \beta_{c,m,t,k}), \tag{4.40}$$

where $\boldsymbol{\beta}_{c,m,t} = (\gamma_{c,m,t,1}, \beta_{c,m,t,1}, \ldots, \gamma_{c,m,t,4}, \beta_{c,m,t,4})$.

Priors and hyper-parameters In section 3.4.4, we defined a conjugate prior for the transformed gamma density. Besides the equivalent sample size α, this prior requires the specification of the logarithm of the expected geometric mean $\chi_{1,c,m,t,k}$ and the expected arithmetic mean $\chi_{2,c,m,t,k}$. We set the equivalent sample size to $\frac{\alpha_m}{T \cdot 2}$ in analogy to the elementary classifier using homogeneous Markov models. The expected arithmetic mean under the assumption of uniform pseudo data can be determined analytically. This assumption entails that all k-mers $\boldsymbol{b} \in \Sigma^k$ occur with identical relative frequencies $q_{\boldsymbol{b}}$ that amount to the product of the relative frequencies of single nucleotides $q = \frac{1}{|\Sigma|}$, and we can determine the expected arithmetic mean as

$$\begin{aligned}
\chi_{2,c,m,t,k} &= \sum_{\boldsymbol{x} \in \Sigma^L} \frac{1}{|\Sigma|^L} \sum_{\boldsymbol{b} \in \Sigma^k} q_{\boldsymbol{b}} \log q_{\boldsymbol{b}} && (4.41)\\
&= \sum_{\boldsymbol{b} \in \Sigma^k} \left[\prod_{i=1}^{k} q\right] \log \left[\prod_{i=1}^{k} q\right] && (4.42)\\
&= k \cdot |\Sigma| \cdot q \log q. && (4.43)
\end{aligned}$$

We know that the geometric mean is always less than or equal to the arithmetic mean. As we cannot determine the geometric mean analytically, we set $\chi_{1,c,m,t,k} := \log(\chi_{2,c,m,t,k} \cdot 0.9995)$, which appears to be an appropriate value in simulations.

Number of CTG trinucleotides

CTG trinucleotides and the reverse complement CAG have been reported to be a main determinant of nucleosome formation (Wang et al., 1994; Lee et al., 2007; Gupta et al., 2008). However, the results of Peckham et al. (2007) indicate that the occurrence of CTG/CAG alone does not discriminate well between nucleosome-bound sequences and linkers. Nonetheless, we include the number of CTG/CAG trinucleotides into the set of numerical properties. We define

$$CTG(\boldsymbol{x}) := n_{\mathrm{CTG}}(\boldsymbol{x}) + n_{\mathrm{CAG}}(\boldsymbol{x}) + 1, \tag{4.44}$$

where $n_{\text{CTG}}(\boldsymbol{x})$ and $n_{\text{CAG}}(\boldsymbol{x})$ count the number of CTG and CAG trinucleotides in $\boldsymbol{x}$, respectively. We add a constant of 1 to the counts, as we want to model $CTG(\boldsymbol{x})$ by a transformed gamma density, which is not defined in case of $CTG(\boldsymbol{x}) = 0$.

Score We define the score for the number of CTG/CAG trinucleotides in sequence $\boldsymbol{x}$ as

$$S_{m,t}^{CTG}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,m,t}) = \mathcal{G}(CTG(\boldsymbol{x})|\gamma_{c,m,t}, \beta_{c,m,t}), \tag{4.45}$$

where $\boldsymbol{\beta}_{c,m,t} = (\gamma_{c,m,t}, \beta_{c,m,t})$.

Prior and hyper-parameters For obtaining the hyper-parameters of the conjugate prior of the gamma density, we approximate the distribution of CTG/CAG trinucleotides by a binomial distribution under the assumption of uniform pseudo data. Since we cannot observe overlapping occurrences of CTG or CAG, we count at most $N = \lfloor \frac{L}{3} \rfloor$ of these trinucleotides. Let k denote the number of occurrences of CTG or CAG trinucleotides. Let $p := P(C) \cdot (P(A) + P(T)) \cdot P(G)$, where under the assumption of uniform pseudo data $P(A) = P(C) = P(G) = P(T) = \frac{1}{4}$, i.e. $p = \frac{1}{32}$. We set the hyper-parameters of the prior to the expectation of $\log(k+1)$ and $(k+1)$ with respect to the binomial distribution, respectively, i.e.

$$\chi_{1,c,m,t} = \sum_k \log(k+1) \binom{N}{k} p^k (1-p)^{N-k} \tag{4.46}$$

$$\chi_{2,c,m,t} = \sum_k (k+1) \binom{N}{k} p^k (1-p)^{N-k} \tag{4.47}$$

As before, we set the equivalent sample size to $\frac{\alpha_m}{T \cdot 2}$.

AT tracts

Long poly-A or poly-T tracts, briefly termed as poly-A/T tracts, are widely reported to prevent formation of nucleosomes (Suter et al., 2000; Yuan et al., 2005; Peckham et al., 2007; Segal and Widom, 2009). Hence, we include the length of the longest, second, third, and fourth longest poly-A/T tract, and the number of poly-A/T tracts with a length greater than or equal to 3, 5, and 7 into the set of numerical properties. We formalize the number of poly-A/T tracts of minimum length n as

$$N_{AT}^n(\boldsymbol{x}) = \left| \{i | \exists k \geq n : x_i \dots x_{i+k-1} \in \{A^k, T^k\} \wedge x_{i-1} \notin \{A, T\} \wedge x_{i+k} \notin \{A, T\}\} \right| + 1$$

and we define the length of the k-th longest poly-A/T tract as

$$L_{AT}^k(\boldsymbol{x}) = \max\{n | \exists i : x_i \dots x_{i+n-1} \in \{A^n, T^n\} \wedge N_{AT}^n(\boldsymbol{x}) \geq k\} + 1,$$

Again, we add a constant of 1 to these values to ensure that the transformed gamma density is always defined.

Scores We use $N_{AT}^{n}(\boldsymbol{x})$ and $L_{AT}^{k}(\boldsymbol{x})$ to define two scores, one for the number of poly-A/T tracts of minimum lengths 3, 5, and 7

$$S_{m,t}^{N_{AT}}(\boldsymbol{x}|c,\boldsymbol{\beta}_{c,m,t}) = \prod_{n\in\{3,5,7\}} \mathcal{G}(N_{AT}^{n}(\boldsymbol{x})|\gamma_{c,m,t,n},\beta_{c,m,t,n}), \tag{4.48}$$

where $\boldsymbol{\beta}_{c,m,t} = (\gamma_{c,m,t,3},\beta_{c,m,t,3},\ldots,\gamma_{c,m,t,7},\beta_{c,m,t,7})$, and one for the length of the k-th longest poly-A/T tract, $k \in \{1,2,3,4\}$

$$S_{m,t}^{L_{AT}}(\boldsymbol{x}|c,\boldsymbol{\beta}_{c,m,t}) = \prod_{k=1}^{4} \mathcal{G}(L_{AT}^{k}(\boldsymbol{x})|\gamma_{c,m,t,k},\beta_{c,m,t,k}), \tag{4.49}$$

where $\boldsymbol{\beta}_{c,m,t} = (\gamma_{c,m,t,1},\beta_{c,m,t,1},\ldots,\gamma_{c,m,t,4},\beta_{c,m,t,4})$.

Prior and hyper-parameters Again, we use the conjugate prior of the gamma density and need to determine hyper-parameters under the assumption of uniform pseudo data. In this case, we can determine the joint cumulative distribution function $P(K \geq k, N \geq n|L)$ of observing at least k poly-A/T tracts of minimum length n in a sequence of length L analytically (see appendix A.3). From the joint cumulative distribution function, we compute the probability that k poly-A/T tracts of a length of at least n occur in a sequence of length L as

$$P(K = k, N \geq n|L) = P(K \geq k, N \geq n|L) - P(K \geq k+1, N \geq n|L), \tag{4.50}$$

and we compute the probability that the k-th longest poly-A /T tract is of length n as

$$P(K \geq k, N = n|L) = P(K \geq k, N \geq n|L) - P(K \geq k, N \geq n+1|L). \tag{4.51}$$

These two probabilities can be used to determine the expectations of the logarithm of the geometric mean and the arithmetic mean required as hyper-parameters of the conjugate prior for the transformed gamma density. We define the hyper-parameters for the prior on the parameters of the score $S_{m,t}^{N_{AT}}(\boldsymbol{x})$ for the number of poly-A/T tracts of minimum length n as

$$\chi_{1,c,m,t,n} = \sum_{k=0}^{L} \log(k+1) P(K = k, N \geq n|L) \tag{4.52}$$

$$\chi_{2,c,m,t,n} = \sum_{k=0}^{L} (k+1) P(K = k, N \geq n|L), \tag{4.53}$$

and we define the hyper-parameters for the prior on the parameters of the score $S_{m,t}^{L_{AT}}(\boldsymbol{x})$ for

the length of the k-th longest poly-A/T tract as

$$\chi_{1,c,m,t,n} = \sum_{n=0}^{L} \log(n+1) P(K \geq k, N = n|L) \tag{4.54}$$

$$\chi_{2,c,m,t,n} = \sum_{n=0}^{L} (n+1) P(K \geq k, N = n|L). \tag{4.55}$$

Again, the equivalent sample size is set to $\frac{\alpha_m}{T \cdot 2}$.

Wavelets

Wavelet energies for the prediction of nucleosome positioning have been proposed by Yuan and Liu (2008), who use the wavelet energies of Haar wavelets in a logistic regression (section 3.2). Here, we extend this approach to the more general MSP learning principle, and we use Mexican hat wavelets instead of Haar wavelets. The mexican hat wavelet $\psi(y)$ is defined as the normalized, negative second derivative of the Gaussian density (see section 3.3.2.1) with a standard deviation of 1 and mean 0, i.e.

$$\psi(t) = c \cdot (1 - t^2) \exp(-\frac{1}{2} t^2), \tag{4.56}$$

where c is a normalization constant. The wavelet function of $t \in [0, 1]$ for scale $i \in \mathbb{R}^+$ and shift $k \in 0, 1, \ldots, 2^i - 1$ is defined as (Yuan and Liu, 2008)

$$\psi(t, i, k) = 2^{\frac{i}{2}} \psi(2^i t - k), \tag{4.57}$$

and results in a scaled and shifted variant of the original mexican hat. The wavelet function is then used in a convolution with the numerical signal $f(\boldsymbol{x}, \ell)$ computed from sequence $\boldsymbol{x}$ at position ℓ to obtain wavelet coefficients $\phi_f(\boldsymbol{x}, i, k)$

$$\phi_f(\boldsymbol{x}, i, k) = \sum_{\ell=1}^{L} f(\boldsymbol{x}, \ell) \psi(\frac{\ell}{L}, i, k). \tag{4.58}$$

The index f indicates that the coefficients depend on the function f. We obtain a large wavelet coefficient $\phi_f(\boldsymbol{x}, i, k) > 0$ if the signal $f(\boldsymbol{x}, \ell)$ matches the wavelet function with scale i shifted by k well, a low coefficient $\phi(\boldsymbol{x}, i, k) < 0$ if the signal matches the negative wavelet function, and a coefficient of 0, if both are uncorrelated.

Here we focus on two scales, namely 3 and 64. The corresponding wavelet functions are depicted in figure 4.23. The wavelet function with scale 3 is chosen as it matches sinusoidal signals with approximately 10 bp periodicities well, i.e. we obtain large absolute values of the wavelet coefficients every ~ 5 bp for such signals. Periodic signals of approximately 10 bp have been reported as important features of nucleosome-bound sequences (Richmond and Davey, 2003; Segal et al., 2006; Liu et al., 2008; Field et al., 2008), e.g. for A/T and G/C dinucleotides or helical properties like tip. In contrast, the wavelet function with scale 64 is not capable of capturing periodicities in signals of length $L = 200$. Instead, it matches a general characteristic of transition between the center and the borders of a signal.

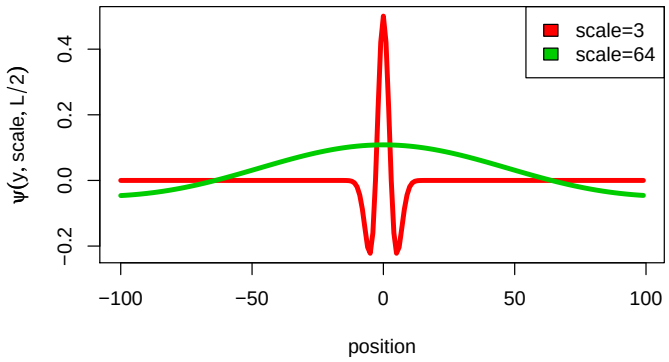


Figure 4.23.: Wavelet function of the mexican hat wavelets for scales 3 and 64 and a sequence of length $L = 200$.

Based on the wavelet coefficients, we define two kinds of wavelet energies. The first is the sum of wavelet coefficients over all possible shifts k

$$E_{f,1}(\boldsymbol{x}, i) = \sum_{k=1}^{L} \phi_f(\boldsymbol{x}, i, k), \tag{4.59}$$

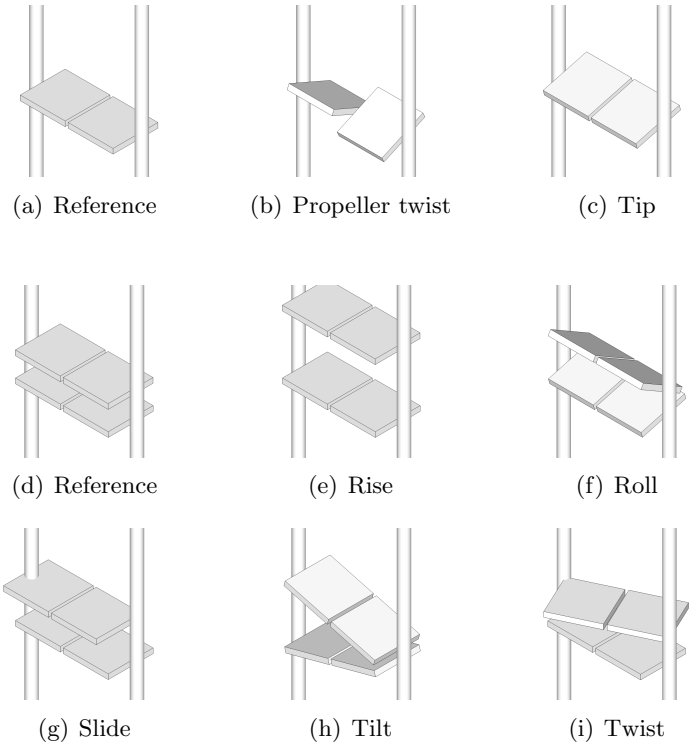
which is an aggregate measure for how well the wavelet function matches the signal. However, it considers the sign of the coefficient, and consequently may lead to the extinction of high absolute coefficients with opposite sign. The magnitude of the absolute values of coefficients are measured by the second wavelet energy, which is defined in analogy to (Yuan and Liu, 2008) as

$$E_{f,2}(\boldsymbol{x}, i) = \sum_{k=1}^{L} \phi_f(\boldsymbol{x}, i, k)^2. \tag{4.60}$$

In the following, we introduce numerical properties of DNA sequences that are considered as numerical signals $f(\boldsymbol{x}, \ell)$ for obtaining the wavelet energies. We convert the discrete sequences $\boldsymbol{x}$ to numerical values according to experimentally determined values of physicochemical and geometrical properties of di- or trinucleotides. These are either obtained from SRS[5] or extracted from the specified publication. We employ the following physicochemical properties determined for dinucleotides:

- The change of *free energy* of helix formation measured in units of $kcal/mol$ (Sugimoto et al., 1996);
- The *melting temperature* of the DNA helix in degree Celsius, i.e. the temperature at which the DNA helix dissociates (Sugimoto et al., 1996);
- The *base stacking energy* of adjacent bases in $kcal/mol$, which results from interaction of the aromatic rings of nucleotides (values of Rein (1973) as reported in (Ornstein et al., 1978)).

[5] `http://srs6.bionet.nsc.ru/srs6bin/cgi-bin/wgetz?-page+LibInfo+-lib+PROPERTY`



(a) Reference (b) Propeller twist (c) Tip

(d) Reference (e) Rise (f) Roll

(g) Slide (h) Tilt (i) Twist

Figure 4.24.: Geometrical properties of the DNA-helix. Sub-figures (a) and (d) depict the reference for the geometry of single and adjacent basepairs, respectively.

We also employ the following geometrical properties of the DNA helix, which are illustrated in figure 4.24:

- The *persistence length*, which is the length of DNA in *nm* for which the DNA molecule can be considered as an elastic rod, and is a measure for the stiffness of the DNA (Sivolob and Khrapunov, 1995);
- The *propeller twist*, which is the rotational angle in degrees between the two bases in a basepair (Gorin et al., 1995);
- *Rise*, which is the distance between adjacent basepairs in Ångström and influences the pitch of the DNA helix (Karas et al., 1996);
- *Roll*, i.e. the opening angle between adjacent basepairs orthogonal to the axis through the two bases of a basepair, of free and complexed DNA (Suzuki et al., 1996);
- *Slide* of free and complexed DNA, which is the translation in Ångström along this axis (Suzuki et al., 1996);
- *Tilt* of free and complexed DNA, which is the opening angle between adjacent basepairs along this axis (Suzuki et al., 1996);
- *Tip*, which is the angle of the conjoint rotation of the two bases in a basepair orthogonal to the helical axis (Karas et al., 1996);
- *Twist*, i.e. the rotation of one basepair against the adjacent one around the helical axis (Karas et al., 1996);
- V_{step}, which is a measure for the size of the space of admissible basepair conformations with unit $\text{deg}^3\text{Å}^3$ (Olson et al., 1998);
- *Bendability*, which measures the flexibility of the DNA helix and has been experimentally

determined for trinucleotides (Brukner et al., 1995).

Additionally, we consider a numerical representation of the occurrence of A/T and G/C dinucleotides, which is neither a physicochemical nor a geometrical property.

Each property determined for dinucleotides can be represented by a matrix $\boldsymbol{r}$, where the rows are indexed by the first nucleotide and the columns are indexed by the second nucleotide of a dinucleotide. As a comprehensible example, we present the dinucleotide matrix $\boldsymbol{r}$ for the occurrence of A/T and G/C dinucleotides, which is

$$\boldsymbol{r} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & -1 & -1 & 0 \\ 0 & -1 & -1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \tag{4.61}$$

where the rows correspond to A, C, G, and T at the first position of the dinucleotide, and the columns correspond to A,C,G, and T at the second position of the dinucleotide. We refer to the entries of the matrix $\boldsymbol{r}$ as $r_{x_\ell,x_{\ell+1}}$, e.g. $r_{C,G}$ for a C at the first position and a G at the second position of the dinucleotide, which can be found in the third column of the second row of the matrix. The matrices for the remaining properties are given in appendix A.4. Bendability, which has been determined for trinucleotides, can be represented by a tensor $\boldsymbol{r}$ with entries $r_{x_\ell,x_{\ell+1},x_{\ell+2}}$, where the three dimensions are indexed by the first, second, and third nucleotide of the trinucleotide, respectively.

In analogy to (Yuan and Liu, 2008), we smooth the resulting sequence of numerical values by computing the mean value over a window of length 3, yielding the final numerical signal

$$f_{\boldsymbol{r}}(\boldsymbol{x}, \ell) \quad = \quad \frac{1}{3} \sum_{j=0}^{2} r_{x_{\ell+j},\ldots x_{\ell+j+R-1}}, \tag{4.62}$$

where $\ell \in \{1, \ldots, L - R + 1\}$, and $R = 2$ for the properties based on dinucleotides and $R = 3$ for bendability.

Scores We model the wavelet energies $E_1(\boldsymbol{x}, i) \in \mathbb{R}$ by a transformed Gaussian density

$$S_{m,t}^{E_{f,1}(i)}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,m,t}) \quad = \quad \mathcal{N}(E_{f,1}(\boldsymbol{x}, i)|\mu_{c,m,t}, \kappa_{c,m,t}), \tag{4.63}$$

where $i \in \{3, 64\}$ denotes the scale and $\boldsymbol{\beta}_{c,m,t} = (\mu_{c,m,t}, \kappa_{c,m,t})$, and we model the wavelet energies $E_2(\boldsymbol{x}, i) \in \mathbb{R}^+$ by a transformed gamma density

$$S_{m,t}^{E_{f,2}(i)}(\boldsymbol{x}|c, \boldsymbol{\beta}_{c,m,t}) \quad = \quad \mathcal{G}(E_{f,2}(\boldsymbol{x}, i)|\gamma_{c,m,t}, \beta_{c,m,t}), \tag{4.64}$$

where $\boldsymbol{\beta}_{c,m,t} = (\gamma_{c,m,t}, \beta_{c,m,t})$.

Priors and hyper-parameters We use a normal-gamma prior (see section 3.4.3) on the parameters of the transformed Gaussian density. For the specification of hyper-parameters, we need the expected mean μ_0 for the Gaussian part, and the expected mean and variance of the

gamma-part of the normal-gamma density. Due to the complexity of the mapping from the discrete input sequences $\boldsymbol{x}$ to the wavelet energies, these expected values are not determined analytically, but obtained from simulations. To this end, we draw $100,000$ sequences according to the assumption of uniform pseudo data and compute the wavelet energies $E_{f,1}(\boldsymbol{x}, i)$ from these random sequences. In turn, we use these values to compute the mean $\hat{\mu}$ and variance $\hat{\sigma^2}$ of the obtained wavelet energies. We set the a-priori mean $\mu_0 = \hat{\mu}$, and choose the hyper-parameters τ_1 and τ_2 such that the expectation and variance of the gamma-part of the normal-gamma density are equal to the estimated precision $\frac{1}{\hat{\sigma^2}}$.

The hyper-parameters of the conjugate prior of the gamma density are determined in the same manner. From the $100,000$ uniformly drawn sequences, we compute the wavelet energies $E_{f,2}(\boldsymbol{x}, i)$ and use these to determine estimates of the logarithm of the expected geometric and the arithmetic mean required to define the hyper-parameters χ_1 and χ_2.

In all cases, we use an equivalent sample size of $\frac{\alpha_m}{T \cdot 2}$.

4.3.2.7. Mapping coverage to probabilities

Although the nucleosome positioning obtained by Field et al. (2008) using parallel sequencing and subsequent mapping to the genome is more reliable than previous data from tiling microarrays, we do not anticipate that it is error-free. One potential source of errors is incomplete digestion by MNase, which leads to false positives in the identification of nucleosomal DNA. Additionally, the number of reads - though large - is limited and, hence, sequences actually bound in nucleosomes might be overlooked. We also assume that the number of reads that cover a specific position on a chromosome gives an indication of the strength of nucleosome formation. We should observe many reads for stretches of DNA that are tightly bound in nucleosome, whereas loosely bound stretches of DNA should be covered by a low number of observed reads, since these are bound in nucleosomes only for a fraction of analyzed yeast cells.

We want to utilize information about the certainty and strength of nucleosome formation for learning the parameters of the component classifiers. To this end, we aim at a mapping from the coverage by nucleosome reads to a probability of nucleosome formation. These probabilities can be used for parameter learning by the weighted variant of the MSP principle for the case of soft-labelling, which we introduced in section 3.2.3 (p. 14). In the following, we describe how we derive such a mapping under the assumption of a random experiment.

We define a random variable B with realizations $\ell \in [1, L]$ corresponding to positions on the genomic region considered, where L denotes the length of this genomic region. This random variable models the situation where a histone core approaches a genomic position and is available for nucleosome formation. We further define a random variables R_ℓ with realization $r \in \{T, F\}$ for observing a nucleosome read (T) or not (F), and random variables M_ℓ with realizations $m \in \{T, F\}$ for finding the center of a nucleosome-bound sequence at a given position. We are interested in the probability $P(M_\ell = T | K = k, B = \ell, N)$ of position ℓ being the center of a nucleosome-bound sequence given that we observe k of N total reads at this position.

We start the derivation of this probability with considering a single read and the probability $P(B = \ell, R_\ell = T|M_\ell = m)$ of drawing position ℓ and observing a read at this position given $M_\ell = m$. We decompose this probability according to the assumption that positions are drawn from B independent of the realization of M_ℓ, yielding

$$P(B = \ell, R_\ell = T|M_\ell = m) = P(B = \ell)P(R_\ell = T|M_\ell = m, B = \ell). \tag{4.65}$$

$P(R_\ell = T|M_\ell = T, B = \ell)$ is the probability to observe a read given a histone core is available at position ℓ and given that this position is the center of a nucleosome-bound sequence, which should be close to 1 if we sequence a sufficient total number of reads N. $P(R_\ell = T|M_\ell = F, B = \ell)$ is the probability to observe a read, although this position is not bound in a nucleosome, due to some source of error in the experiment. We define the probabilities of drawing a position ℓ as $P(B = \ell) = p_\ell$, and we define $P(R_\ell = T|M_\ell = T, B = \ell) = p_T$ and $P(R_\ell = T|M_\ell = F, B = \ell) = p_F$.

Extending the experiment to multiple independent reads, the probability of observing k reads at position ℓ in N independent drawings given it is the center of a nucleosome-bound sequence, i.e. $M_\ell = T$, amounts to

$$P(K_\ell = k, B = \ell|M_\ell = T, N, \boldsymbol{p}) = \binom{N}{k} (p_\ell p_T)^k (1 - p_\ell p_T)^{N-k}, \tag{4.66}$$

where K_ℓ is a random variable representing the number of observed reads at position ℓ with realizations $k \in [0, N]$, and $\boldsymbol{p} = (p_\ell, p_T, p_F)$. The corresponding probability given $M_\ell = F$ amounts to

$$P(K_\ell = k, B = \ell|M_\ell = F, N, \boldsymbol{p}) = \binom{N}{k} (p_\ell p_F)^k (1 - p_\ell p_F)^{N-k}. \tag{4.67}$$

Additionally, we introduce an a-priori probability $P(M_\ell = T|p_{\text{nuc}}) := p_{\text{nuc}}$ of nucleosome formation at a given position ℓ. We can now express the probability $P(M_\ell = T|K_\ell = k, B = \ell, N, p_{\text{nuc}}, \boldsymbol{p})$ of nucleosome formation given that k of N reads are observed at position ℓ and given parameters $\boldsymbol{p}$ and p_{nuc} in terms of the a-priori probability $P(M_\ell = T|p_{\text{nuc}})$ of nucleosome formation and the probability $P(K_\ell = k, B = \ell|M_\ell = F, N, \boldsymbol{p})$ of observing k reads at the center of a nucleosome ($M_\ell = T$) or a linker ($M_\ell = F$), yielding

$$\begin{aligned} &P(M_\ell = T|K_\ell = k, B = \ell, N, p_{\text{nuc}}, \boldsymbol{p}) \\ &\quad = \frac{P(M_\ell = T|p_{\text{nuc}})P_\ell(K_\ell = k, B = \ell|M_\ell = T, N, \boldsymbol{p})}{P(K_\ell = k, B = \ell|N, p_{\text{nuc}}, \boldsymbol{p})} \\ &\quad = \frac{p_{\text{nuc}} (p_\ell p_T)^k (1 - p_\ell p_T)^{N-k}}{p_{\text{nuc}} (p_\ell p_T)^k (1 - p_\ell p_T)^{N-k} + (1 - p_{\text{nuc}}) (p_\ell p_F)^k (1 - p_\ell p_F)^{N-k}}. \end{aligned} \tag{4.68}$$

Here, we set $\forall \ell : p_\ell = \frac{1}{L}$, where L denotes the length of the considered genomic region. We set $p_F = 0.1, p_T = 1.0$, and $p_{\text{nuc}} = 0.8 \cdot \frac{147}{20}$, which reflects the average length of a nucleosome-bound sequence of 147 bp and the uncertainty due to the sequencing of reads longer than 147 bp of ~ 20 bp (Yair Field, personal communication). The resulting mapping from the number of reads to probabilities of nucleosome formation is illustrated in figure 4.25. We observe that

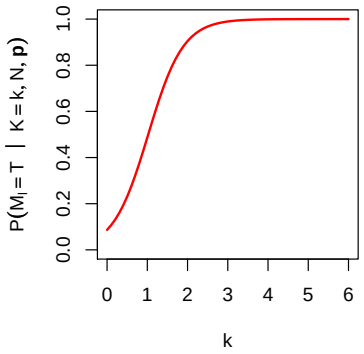


Figure 4.25.: Mapping from number of reads k to probabilities of nucleosome formation $P(M_\ell = T|K = k, N, \boldsymbol{p})$.

using this mapping even a coverage by 0 reads leads to a probability $P(M_\ell = T|K_\ell = 0, B = \ell, N, p_{\text{nuc}}, \boldsymbol{p})$ of nucleosome formation slightly different from 0, a coverage of 1 results in a probability $P(M_\ell = T|K_\ell = 1, B = \ell, N, p_{\text{nuc}}, \boldsymbol{p})$ of approximately 0.5, and a coverage of 3 and above corresponds to a probability of almost 1.

For each chromosome of *Saccharomyces cerevisiae* we compute the number k of reads centered at each position l from the mapped reads (see section 4.3.3) and apply a Gaussian smoothing with a standard deviation of 20 to the obtained coverages. The resulting smoothed coverages are then mapped to probabilities according to equation (4.68). These probabilities serve as weights for the weighted variant of the MSP principle (see equation (3.25)) when learning the component classifiers in the next section. We denote the weights for a sequence $\boldsymbol{x}_n$ centered at position ℓ by $\boldsymbol{w}_n = (P(M_\ell = T|K_\ell = k_\ell, B = \ell, N, p_{\text{nuc}}, \boldsymbol{p}), 1 - P(M_\ell = T|K_\ell = k_\ell, B = \ell, N, p_{\text{nuc}}, \boldsymbol{p}))$, and we denote the vector of all weights by $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N)$ in analogy to the vector of correct classes $\boldsymbol{c}$.

4.3.2.8. Learning of component classifiers

The learning of the component classifiers is conducted for each type of DNA, i.e. coding, non-coding, and border sequences, independently. We use a three-stage learning procedure, which first learns the parameters of the elementary classifiers independently, greedily selects the elementary classifiers to be employed in a component classifier, and, finally, jointly learns the parameters of the selected elementary classifiers and the mixture probabilities used in the weighted voting.

The pseudo code of this learning algorithm is presented in figure 4.26 and explained in the following. We assess elementary classifiers and preliminary component classifiers by a two-fold cross validation on the training data. In order to reduce variations of performance during one learning procedure due to different partitionings, we partition the data set of training sequences $\boldsymbol{X}_m$ and the associated weights $\boldsymbol{w}_m$ beforehand. We denote the set of elementary classifiers that may be selected for the component classifiers by $\mathcal{E}$. Initially, the set $\mathcal{E}$ comprises all elementary classifiers introduced above. We denote the elementary classifiers by the employed score $S_{m,t}$ in the following.

Partition $\boldsymbol{X}_m$ and $\boldsymbol{w}_m$ into $\boldsymbol{X}_{m,1}, \boldsymbol{X}_{m,2}$ and $\boldsymbol{w}_{m,1}, \boldsymbol{w}_{m,2}$
/* Elementary classifiers */
$\mathcal{E} := \bigcup_{d_2=0}^{3} \bigcup_{d_1=0}^{i} \{S_{m,t}^{\mathrm{hMM}(d_2,d_1)}\} \cup \{S_{m,t}^{\mathrm{iMM}}, S_{m,t}^{H}, S_{m,t}^{H^c}, S_{m,t}^{CTG}, S_{m,t}^{N_{AT}}, S_{m,t}^{L_{AT}}\}$
$\bigcup_{i\in\{3,64\}} \bigcup_{\boldsymbol{r}} \{S_{m,t}^{E_{f_{\boldsymbol{r}},1}(i)}, S_{m,t}^{E_{f_{\boldsymbol{r}},2}(i)}\}$
for $S_{m,t} \in \mathcal{E}$ **do**
  for $i \in \{1,2\}$ **do**
    // optimize parameters of elementary classifier $S_{m,t}$
    $\boldsymbol{\beta}_{m,t}^{i} = \underset{\boldsymbol{\beta}_{m,t}}{\operatorname{argmax}} \left[\log \mathrm{CL}(\boldsymbol{w}_i|\boldsymbol{X}_i, \boldsymbol{\beta}_{m,t}) + \log q\left(\boldsymbol{\beta}_{m,t} \,|\, \boldsymbol{\alpha}_{m,t}\right)\right]$
  done
done
/* Select best elementary classifier */
$\mathcal{T}_m := \left\{ \underset{S_{m,t}\in\mathcal{E}}{\operatorname{argmax}} \left\{ \left[\log \mathrm{CL}(\boldsymbol{w}_{m,1}|\boldsymbol{X}_{m,1}, \boldsymbol{\beta}_{m,t}^{2}) + \log \mathrm{CL}(\boldsymbol{w}_{m,2}|\boldsymbol{X}_{m,2}, \boldsymbol{\beta}_{m,t}^{1})\right] \right\} \right\}$
$last := \max_{S_{m,t}\in\mathcal{E}} \left\{ \left[\log \mathrm{CL}(\boldsymbol{w}_{m,1}|\boldsymbol{X}_{m,1}, \boldsymbol{\beta}_{m,t}^{2}) + \log \mathrm{CL}(\boldsymbol{w}_{m,2}|\boldsymbol{X}_{m,2}, \boldsymbol{\beta}_{m,t}^{1})\right] \right\}$
/* Select further elementary classifiers */
do
  $opt :=$ NIL; $best := -\infty$
  $\mathcal{E} := \mathcal{E} \setminus \mathcal{T}_m$
  for $S_{m,t} \in \mathcal{E}$ **do**
    $\mathcal{T}'_m := \mathcal{T}_m \cup \{S_{m,t}\}$
    for $i \in \{1,2\}$ **do**
      // optimize parameters $\boldsymbol{\beta}_{m,\mathcal{T}'_m}$ of weights of voting
      $\boldsymbol{\beta}_{m,\mathcal{T}'_m}^{i} := \underset{\boldsymbol{\beta}_{m,\mathcal{T}'_m}}{\operatorname{argmax}} \big[\log \mathrm{CL}(\boldsymbol{w}_i|\boldsymbol{X}_i, \boldsymbol{\beta}_m^i \setminus \boldsymbol{\beta}_{m,\mathcal{T}_m}^{i} \cup \boldsymbol{\beta}_{m,\mathcal{T}'_m} \cup \boldsymbol{\beta}_{m,t}^{i})$
      $+ \log q\left(\boldsymbol{\beta}_m^i \setminus \boldsymbol{\beta}_{m,\mathcal{T}_m}^{i} \cup \boldsymbol{\beta}_{m,\mathcal{T}'_m} \cup \boldsymbol{\beta}_{m,t}^{i} \,\middle|\, \boldsymbol{\alpha}_m \setminus \boldsymbol{\alpha}_{m,\mathcal{T}_m} \cup \boldsymbol{\alpha}_{m,\mathcal{T}'_m} \cup \boldsymbol{\alpha}_{m,t}\right)\big]$
      $\boldsymbol{\beta}_m^{i\,\prime} := \boldsymbol{\beta}_m^i \setminus \boldsymbol{\beta}_{m,\mathcal{T}_m}^{i} \cup \boldsymbol{\beta}_{m,\mathcal{T}'_m}^{i} \cup \boldsymbol{\beta}_{m,t}^{i}$
    done
    $curr := \left[\log \mathrm{CL}(\boldsymbol{w}_{m,1}|\boldsymbol{X}_{m,1}, \boldsymbol{\beta}_m^{2\,\prime}) + \log \mathrm{CL}(\boldsymbol{w}_{m,2}|\boldsymbol{X}_{m,2}, \boldsymbol{\beta}_m^{1\,\prime})\right]$
    if $curr > best$ **then**
      $best := curr$
      $opt := S_{m,t}$
    fi
  done
  $\mathcal{T}'_m := \mathcal{T}_m \cup \{opt\}$
  for $i \in \{1,2\}$ **do**
    // optimize all parameters for $\mathcal{T}'$
    $\boldsymbol{\beta}_m^{i\,\prime} := \underset{\boldsymbol{\beta}_m}{\operatorname{argmax}} \left[\log \mathrm{CL}(\boldsymbol{w}_i|\boldsymbol{X}_i, \boldsymbol{\beta}_m) + \log q\left(\boldsymbol{\beta}_m \,|\, \boldsymbol{\alpha}_m\right)\right]$
  done
  $best := \left[\log \mathrm{CL}(\boldsymbol{w}_{m,1}|\boldsymbol{X}_{m,1}, \boldsymbol{\beta}_m^{2\,\prime}) + \log \mathrm{CL}(\boldsymbol{w}_{m,2}|\boldsymbol{X}_{m,2}, \boldsymbol{\beta}_m^{1\,\prime})\right]$
  if $best > last$ **then**
    // retain $\mathcal{T}'_m$
    $\mathcal{T}_m := \mathcal{T}'_m$
    $last := best$
  fi
while $\mathcal{T}_m$ changed
/* optimize all parameters for $\mathcal{T}_m$ on complete data */
$\boldsymbol{\beta}_m^* := \underset{\boldsymbol{\beta}_m}{\operatorname{argmax}} \left[\log \mathrm{CL}(\boldsymbol{w}_m|\boldsymbol{X}_m, \boldsymbol{\beta}_m) + \log q\left(\boldsymbol{\beta}_m \,|\, \boldsymbol{\alpha}_m\right)\right]$

Figure 4.26.: Pseudo code of the algorithm for learning the component classifiers.

For each of the elementary classifiers $S_{m,t}$, we learn parameters $\boldsymbol{\beta}^i_{m,t}$ from the sequences in partition $\boldsymbol{X}_{m,i}$ and associated weights $\boldsymbol{w}_{m,i}$ by the weighted variant of the discriminative MSP principle (see section 3.2.3, p. 14). We do not explicitly denote the dependency of the conditional likelihood on the score $S_{m,t}$, whenever the employed score can be derived from the denotation of parameters $\boldsymbol{\beta}_{m,t}$.

We select the best elementary classifiers by means of the weighted variant of conditional likelihood in a two-fold cross validation. We do not include the value of the prior into the evaluation, because training and test data do not overlap and thus over-fitting effects should not be relevant. For each elementary classifier $S_{m,t}$, we consider parameters $\boldsymbol{\beta}^1_{m,t}$ and $\boldsymbol{\beta}^2_{m,t}$ that are learned from the first and second partition of the data, respectively. We use the parameter values $\boldsymbol{\beta}^2_{m,t}$ for the evaluation of the conditional likelihood on the first partition, i.e. $\boldsymbol{X}_{m,1}$ and $\boldsymbol{w}_{m,1}$, and we use the parameter values $\boldsymbol{\beta}^1_{m,t}$ for the evaluation on the second partition. We include into the set of selected elementary classifiers $\mathcal{T}_m$ that classifier $S_{m,t}$ yielding the maximum sum of these two values of conditional likelihood. This sum is stored in a variable *last* after the selection.

In the following *while*-loop, we greedily augment the set of selected elementary classifier, as long as the conditional likelihood in the evaluation increases. Again, we test each elementary classifier in $\mathcal{E}$ – except those already selected and consequently included into $\mathcal{T}_m$. In a learning step, we adjust only the parameters $\boldsymbol{\beta}^i_{m,\mathcal{T}'_m}$ responsible for the weights of the voting, but retain all other parameters of the previously trained preliminary component classifier and the newly selected elementary classifier. We evaluate the augmented component classifier with parameters $\boldsymbol{\beta}^1_m{}'$ and $\boldsymbol{\beta}^2_m{}'$ learned on the first and second partition, respectively, using the sum of conditional likelihoods. If the current preliminary component classifier achieves a better result *curr* than the previously selected, we remember $S_{m,t}$ as the new selection *opt* together with the corresponding sum of conditional likelihoods *best*.

After one iteration of the selection, we include the final selection *opt* into a preliminary set of selected classifiers $\mathcal{T}'_m$, and now optimize[6] all parameters $\boldsymbol{\beta}^i_m{}'$ on the corresponding partitions of the data. We again evaluate the augmented selection using these optimized parameters. If the resulting sum of conditional likelihoods *best* is greater than the previously optimal value *last*, the selection of elementary classifiers in $\mathcal{T}'_m$ is retained. If we do not observe an improvement, we discard the augmented selection in $\mathcal{T}'_m$ and $\mathcal{T}_m$ remains unchanged, which terminates the while-loop.

Finally, we learn the optimal parameters $\boldsymbol{\beta}^*_m$ of the component classifier using the elementary classifiers in the final set $\mathcal{T}_m$ on the complete training data $\boldsymbol{X}_m, \boldsymbol{w}_m$.

4.3.2.9. Utilizing preferences of linker lengths

In the previous sections, we explained, how the parameters of the classifier distinguishing coding, non-coding, and borders sequences are learned, how we select the elementary classifiers of component classifiers and learn their parameters, and how the votings of the component classifiers are combined in weighted voting to obtain the probability of nucleosome formation

[6] In the implementation, we stop the optimization after at most 50 steps to reduce runtime.

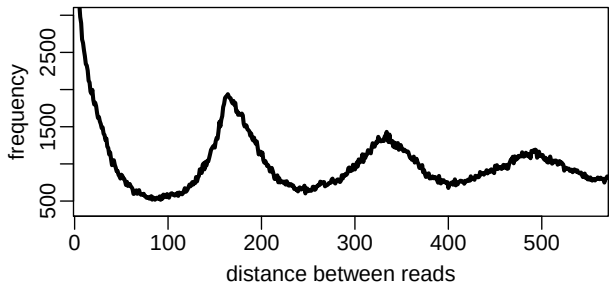


Figure 4.27.: Frequencies of observed distances between centers of nucleosome reads computed on the mapped reads of (Field et al., 2008).

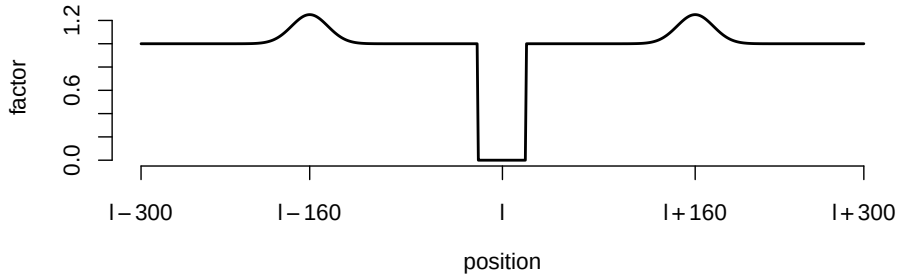


Figure 4.28.: Factors used to re-weight probabilities of nucleosome formation. A region of ±20 bp round the nucleosome predicted at position l is masked by setting probabilities to 0, while scores in the preferred distance of ~ 160 bp are up-weighted.

$P(\text{nuc} \mid \boldsymbol{x}, \{\mathcal{T}_m\}, \boldsymbol{\beta})$ for a sequence $\boldsymbol{x}$. After having obtained these probabilities for all subsequences of, for instance, a chromosome of *S. cerevisiae*, we add another post-processing step to obtain the final prediction of nucleosome positioning, which is motivated in the following.

The binding of nucleosomes to DNA is often illustrated as "beads on a string". The arrangement of these "beads", i.e. the nucleosomes, is not arbitrary, but preferred distances between nucleosomes and, as a consequence, preferred lengths of the linker sequences between the nucleosomes can be observed (Wang et al., 2008; Lubliner and Segal, 2009). In figure 4.27, we plot the number of pairs of nucleosome reads that exhibit a certain distance between their centers as observed from the mapped reads of Field et al. (2008). We find that the largest number of pairs occurs for very short distance (20 and below). These distances can be attributed to the expected impreciseness of the experimental method, since different reads belonging to the same nucleosome position may be shortened to a different degree by MNase. Another peak of observed frequency can be observed for a distance of ~ 160 bp. Under the assumption that ~ 147 bp of DNA are bound in a nucleosome, this corresponds to a very short linker length. Since we do not restrict the analysis to directly neighboring reads, the preferred distance of 160 bp between nucleosome centers re-occurs as lower peaks at ~ 320 bp and ~ 480 bp.

With the goal of utilizing these preferred linker lengths for the prediction of nucleosome positions, we apply a simple post-processing step to the probabilities obtained from the voting of components. As a first step, we apply a Gaussian smoothing with standard deviation of 25 to the obtained probabilities to avoid local discontinuities. We consider all positions along a chromosome and select that position ℓ exhibiting the highest probability $P(\text{nuc} \mid x_{\ell-100}, \ldots, x_{\ell+99}, \{\mathcal{T}_m\}\boldsymbol{\beta})$. For ± 20 bp around this position ℓ, we set the probabilities of nucleosome formation to 0. This does not imply that we consider a slightly shifted positioning impossible, but reflects that we can not determine the positioning of nucleosomes more accurately experimentally. Additionally, we slightly upweight the probabilities in the preferred distance by a bell-shaped function, where the maximum corresponds to a 1.25-fold increase of probability. This procedure can be perceived as a point-wise multiplication of the original vector of probabilities with the function depicted in figure 4.28. In a window of ± 20 bp around the center this function is equal to 0, whereas in a distance of $\sim \pm 160$ bp we find factors greater than 1 with a maximum of 1.25. We determine the next maximum from these re-weighted probabilities and report the original probability as probability of nucleosome formation at this position. If multiple positions in the direct vicinity of the chosen position exhibit the same or an even higher original probability, we relocate the reported position to the average of these positions. Although this procedure does not modify the probabilities of nucleosome formation, it affects the order of reporting and, consequently, allows for slight shifts in the predicted positioning of nucleosomes accounting for preferred distances between adjacent nucleosomes.

4.3.3. Data & Evaluation

We use the data of Field et al. (2008) for the subsequent experiments. We obtain the mapped reads of (Field et al., 2008) from `http://genie.weizmann.ac.il/pubs/field08/data/YeastMappedReads.tab.gz`. The filtered reads of length between 127 and 177 bp are further post-processed using the following protocol (Yair Field, personal communication): First all reads are extended to a minimum length of 147 bp. The resulting reads are then shrunken by 63 bp on either side, resulting in intervals of length 21 for reads of length 147 and in larger intervals for longer reads. Each read contributes to the coverage at the positions within this associated interval with a weight of one, i.e. the coverage at position ℓ is equal to the number of reads with a shrunken interval overlapping this position. Field et al. (2008) define regions that exhibit a coverage above a number of thresholds to reflect different levels of nucleosome stability. The thresholds considered are 1, i.e. all regions occupied by nucleosomes, 2, 4, 8, and 16. For each of the thresholds, consecutive regions with a minimum coverage according to this threshold are determined, and the resulting intervals of minimum coverage are reduced to ± 20 bp around their center. We present the number of such regions for the different coverages in table 4.4.

Linker regions are defined as consecutive regions of length 50 to 500 bp without mapped reads, excluding repetitive regions that are not considered in the initial mapping. In analogy to the regions occupied by nucleosomes, the linker regions are reduced to their center ± 20 bp. The resulting lists of chromosomal intervals of length 41 bp are kindly provided by Yair Field (personal communication) and these are used for the experiments presented here.

Table 4.4.: Number of nucleosome-bound sequences and linkers across all 16 chromosomes of *S. cerevisiae* as determined from the mapped reads of (Field et al., 2008).

coverage	1	2	4	8	16
number of nucleosome-bound regions	84410	69703	38787	12076	1601
number of linkers	8017				

We evaluate the classification performance of the voting of components in a cross validation experiment over the 16 chromosomes of *S. cerevisiae*. For each iteration of the cross validation, we exclude the data, i.e. the chromosomal intervals of nucleosome-bound sequences and linkers, of one chromosome from the training data, and test the learned classifier on the excluded chromosome. We use all regions with a minimum coverage of 1 and all linkers in the training data for learning the classifier, whereas we independently test the learned classifier for minimum coverages of 1, 2, 4, 8, and 16. However, due to the mapping from coverage to probabilities (see section 4.3.2.7) the stability of nucleosome formation is still reflected in the training data. Since the number of nucleosome-bound sequences and linkers varies considerably for the different chromosome due to their length, we average the considered performance measures over the 16 chromosomes, weighted by the number of nucleosome-bound sequences and linkers.

Like Field et al. (2008), we use AUC-ROC as performance measure. However, for low minimum coverages, the test data contain approximately 10 times as many annotated nucleosome-bound sequences as linkers. On the other hand, the number of annotated nucleosome-bound sequences with a minimum coverage of 16 is approximately a fifth of the number of linkers. Hence, we also consider AUC-PR and AUC-PRI. AUC-PR is especially appropriate for high coverages, while AUC-PRI gives a good impression of classification performance for low coverages (see section 3.5.1).

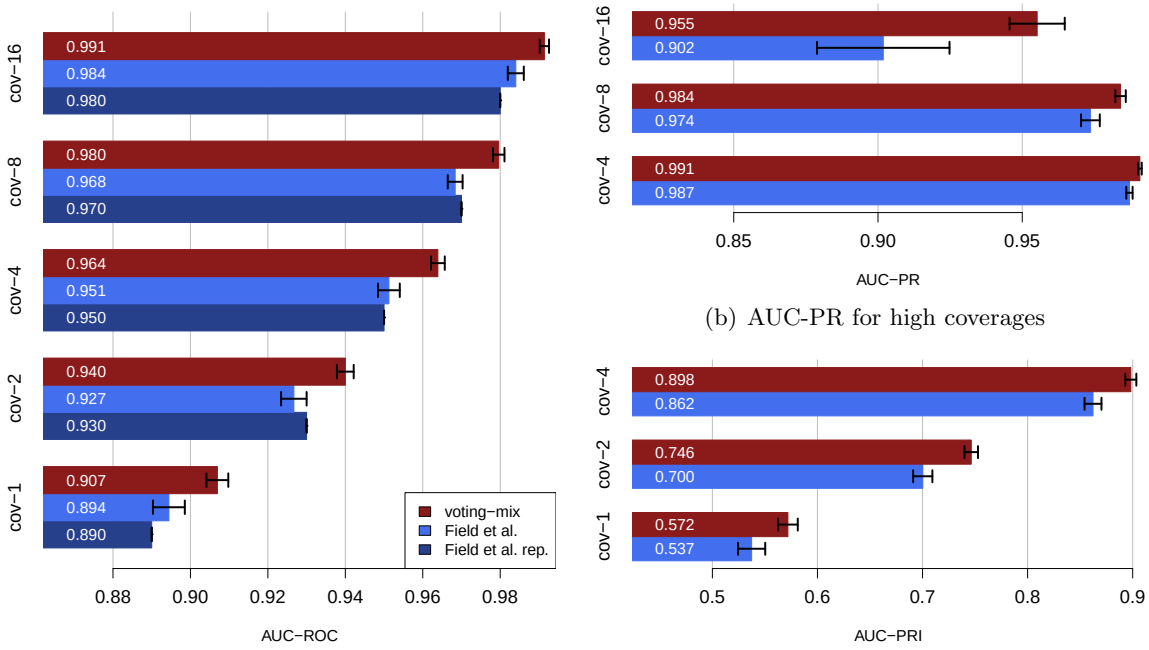
4.3.4. Results & Discussion

We start the evaluation of the proposed approach with a comparison of classification performance to that of the approach of Field et al. (2008). We then analyze to which degree the different aspects of voting of components, namely the elementary classifiers used in component classifiers, weighting of data, and post-processing, contribute to classification performance. We scrutinize some of the selected classifiers for features of nucleosome binding, and we investigate which role periodicities play in nucleosome positioning. Finally, we survey predictions in their genomic context and compare the predictions of both approaches considered.

4.3.4.1. Comparison to Field et al. (2008)

First, we compare the classification performance of voting of components with mixtures of elementary classifiers as component classifiers (voting-mix) to the approach of Field et al. (2008) by means of AUC-ROC, which was also chosen in their study. Since Field et al. (2008) could show that their approach yields a superior classification performance compared to the approaches of (Ioshikhes et al., 2006), (Segal et al., 2006), (Peckham et al., 2007), and (Yuan

and Liu, 2008) on the data obtained by parallel sequencing, we consider only this approach in the following comparison. In addition to the originally reported values of AUC-ROC, we compute AUC-ROC on the scores available at `http://genie.weizmann.ac.il/pubs/field08/field08_genomes.html` as *model score*, which differs slightly from the reported values due to rounding (see figure 4.29(a)). We also test the *Genomica*[7] file available on the same web-page, which result in lower values of AUC-ROC than using the model scores. Field et al. (2008) use a different procedure for obtaining the average AUC-ROC over all chromosomes, where they first merge the scores for all 16 test data sets and use these merged scores to determine the ROC curve. However, applying this procedure to the average occupancy scores results in only slightly different values of AUC-ROC compared to the weighted averaging. We stick to the variant of weighted averaging, because it allows for computing standard errors along with the mean values, which give an estimate of the significance of observed differences between the considered approaches. Here, we consider a deviation of two-fold the standard error significant.



(a) AUC-ROC for different coverages

(b) AUC-PR for high coverages

(c) AUC-PRI for low coverages

Figure 4.29.: Comparison of the classification performance of voting of components using mixtures of selected elementary classifiers as component classifiers (voting-mix, red) to that of the approach of (Field et al., 2008). For AUC-ROC (a), we show the performance as reported in (Field et al., 2008) (dark blue) and as computed from the model scores available at `http://genie.weizmann.ac.il/pubs/field08/field08_genomes.html` (light blue). Since AUC-PR (b) and AUC-PRI (c) are not evaluated in (Field et al., 2008), we resort to the computation on the available score in these cases. Error bars indicate two-fold standard error in both directions.

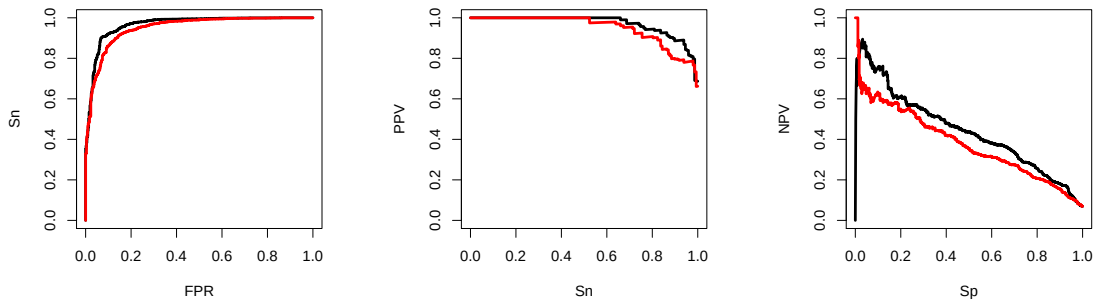
The results considering AUC-ROC are presented in figure 4.29(a) for levels of coverage between 1 and 16. We find that voting-mix (red) consistently yields a higher AUC-ROC compared to the values reported in (Field et al., 2008) (dark blue) and the values of AUC-ROC computed from the average occupancy scores (light blue). We conclude from the error bars, which

[7]Genomica is a browser for genome annotations published by the same group

indicate two-fold the standard error in both directions, that this improvement is significant in all five cases. Voting-mix yields an AUC-ROC of 0.907 even for a coverage of 1 compared to 0.89 reported in (Field et al., 2008) and 0.894 computed on the model scores. Interestingly, voting-mix can achieve a considerably higher classification accuracy than the approach of (Field et al., 2008) for coverages of at least 16 as well, improving AUC-ROC from 0.98 and 0.984, respectively, to 0.991. Since AUC-ROC can be interpreted as the probability that a randomly chosen nucleosome-bound sequence obtains a higher score than a randomly chosen linker sequence (Fawcett, 2006), this improvement means that using voting-mix the probability of false predictions is reduced almost by half.

In addition to AUC-ROC, we consider AUC-PR as performance measure for the higher levels of coverage, namely 4, 8, and 16 as presented in figure 4.29(b). Again, voting-mix significantly outperforms the approach of (Field et al., 2008) for all levels of coverage considered. Interestingly, the approach of (Field et al., 2008) achieves an average AUC-PR of only 0.902 for a coverage of 16 and above, whereas classification accuracy rapidly increases with lowering the level of coverage. For a minimum coverage of 8, this approach already yields an AUC-PR of 0.974, and for a minimum coverage of 4, AUC-PR further increases to 0.987. This can partly be explained by the number of nucleosome-bound regions defined for these coverages, which increases by a factor of ~ 7.5 from coverage 16 to 8. We might speculate that many of these additional regions obtain a high probability of nucleosome formation and, hence, the overall PPV increases. Interestingly, we do not observe this deterioration of performance from coverage 8 to 16 for voting-mix. One reason for this observation might be that we assign high weights to sequences exhibiting a high coverage when learning the component classifiers of voting-mix and, hence, the parameters of voting-mix reflect the properties of highly-covered sequences better than those of the approach of (Field et al., 2008).

Considering AUC-PRI as performance measure for regions with a low minimum coverage in figure 4.29(c), we find an improved classification performance of voting-mix compared to the approach of (Field et al., 2008) as well. In contrast to AUC-PR for high coverages, the magnitude of improvement gained by voting-mix is similar for all coverages from 1 to 4. Again, all observed differences in AUC-PRI are significant considering the two-fold standard error.



(a) ROC curve, cov-4 (b) PR curve, cov-16 (c) PRI curve, cov-1

Figure 4.30.: ROC curve for a coverage of 4, PR curve for a coverage of 16, and PRI curve for a coverage of 1 comparing the approach of (Field et al., 2008) (red line) to voting mix (black line).

The areas under the ROC curve, the PR curve, and the PRI curve are aggregate measures

and, hence, only give an overall picture of classification performance. In order to investigate if the observed differences between voting-mix and the approach of (Field et al., 2008) can be attributed to deviations of classification performance in practically relevant regions of the curves, we present the three curves for chromosome 4 in figure 4.30. We choose chromosome 4, because it is the largest chromosome of *S. cerevisiae* and for this reason the corresponding test data set comprises the largest number of supporting points for plotting the curves. However, the general picture remains the same for the other chromosomes (data not shown). For a coverage of 4 we observe a difference of 0.014 in AUC-ROC between voting-mix and the approach of (Field et al., 2008). Considering the ROC curve for the same coverage presented in figure 4.30(a), we find that voting-mix yields the same or an even better Sn than the approach of (Field et al., 2008) across the whole range of FPR. The greatest differences between the two approaches can be observed for an FPR between 0.05 and 0.25, where both approaches achieve an Sn above 0.7. Since lower values of Sn would leave a great fraction of nucleosome-bound sequences unrecognized and a higher FPR would result in many erroneously classified linker regions, we may state that the improvement of AUC-ROC gained by voting-mix can be attributed to a practically relevant region of the ROC curve.

If we consider the same interval of Sn for the PR curve for a minimum coverage of 16 in figure 4.30(b), we find an almost consistently greater PPV for voting-mix than for the approach of (Field et al., 2008). However, both approaches yield a PPV above 0.5 for the whole range of Sn indicating a generally reasonable classification performance. An FPR between 0.05 and 0.25, as considered for the ROC curve, corresponds to a Sp between 0.75 and 0.95 for the PRI curve for minimum coverage 1, since the linker regions considered do not change for the different levels of coverage. We observe from figure 4.30(c) that in this region of the PRI curve, both approaches achieve a low NPV of at most 0.3, which means that for every correctly predicted linker sequence we obtain approximately two additional false negatives, i.e. two nucleosome-bound sequences that are erroneously predicted as linkers. However, since the number of nucleosome-bound sequences for a coverage of 1 is approximately 10 times as high as the number of linker sequences, this ratio of true negatives to false negatives is more acceptable than it appears at first sight.

Summarizing the above results, we find that voting-mix significantly outperforms the approach of (Field et al., 2008) considering AUC-ROC, AUC-PR, and AUC-PRI, and we ascertain that the observed differences in the AUC values can be attributed to practically relevant regions of the corresponding curves. In the following, we investigate the contributions of the different aspects of voting-mix to this improved classification performance.

4.3.4.2. Selected elementary classifiers

As a first analysis, we investigate which of the elementary classifiers are selected for the component classifiers for coding, non-coding, and border sequences. The results of this analysis are visualized in figure 4.31, where we consider the classification performance, the number of iterations of the cross validation in which an elementary classifier is selected, and the average weight assigned to selected classifiers. We exclude from this illustration all elementary classifiers that are never selected by the greedy selection procedure.

In the left block of figure 4.31, we present the classification performance of component classifiers that consist only of the elementary classifier specified by the row name. We measure classification performance by the average AUC-PRI for a coverage of 1 over all 16 iterations of cross validation, i.e. over all 16 chromosomes. Classification for a coverage of 1 is the most challenging classification task of all coverages considered. In addition to AUC-PRI on all types of sequences presented in the first column of this block, we also measure classification performance separately considering the coding, non-coding, and border sequences in the test data sets.

We find that the elementary classifiers using homogeneous Markov models yield a consistently high AUC-PRI for all types of sequences. The classification of non-coding sequences especially profits from higher order Markov models, whereas this tendency is less pronounced for coding and border sequences. Coding and border sequences are also classified well by the elementary classifiers using the distribution of wavelet energies computed on the occurrence of A/T vs. G/C dinucleotides, computed on the base stacking energy, and computed on the melting temperature, using a scale of 64.

Interestingly, the elementary classifier using the number of CTG/CAG trinucleotides, which have been reported to be relevant for nucleosome formation (Wang et al., 1994; Lee et al., 2007; Gupta et al., 2008), is never selected for any of the component classifiers and for this reason is omitted from figure 4.31. Similarly, long poly-A/T tracts are known to be relevant for nucleosome depletion (Suter et al., 2000; Yuan et al., 2005; Peckham et al., 2007; Segal and Widom, 2009), but the elementary classifier using the length of poly-A/T tracts exhibits a generally low classification performance. In contrast, the elementary classifier using the entropy of k-mers for the reduced A/T vs G/C alphabet, which is one of the most simple numerical properties considered, classifies coding and border sequences surprisingly well.

The only wavelet energy for a scale of 3 that is actually selected is that for roll of complexed DNA. This is notable, because the mexican hat wavelet with a scale of 3 is generally capable of detecting ~ 10 bp periodicities, which have been widely found for nucleosome-bound sequences (Richmond and Davey, 2003; Segal et al., 2006; Liu et al., 2008; Field et al., 2008). We scrutinize the presence of these periodicities in nucleosome-bound sequences in section 4.3.4.5.

Turning to the second block of figure 4.31, which visualizes the number of cross validation iterations in which an elementary classifier is selected for one of the component classifiers, we find that three elementary classifiers are selected with exceptional total frequency. These are the elementary classifier using homogeneous Markov models of orders $d_1 = 1$ at the borders and $d_2 = 2$ in the center, the elementary classifier using the wavelet energy E_1 computed on the occurrence of A/T vs. G/C dinucleotides with a scale of 64, and the elementary classifier using the wavelet energy E_2 on the values of the geometrical property slide with scale 64. Considering the elementary classifiers selected for coding, non-coding, and border sequences separately, we observe that the elementary classifier using the homogeneous Markov models is mostly used for coding and non-coding sequences, the elementary classifier using the wavelet on A/T vs G/C dinucleotides is mainly selected for coding and border sequences, and the elementary classifier using slide is used most frequently for non-coding sequences. We find a similar pattern considering the elementary classifiers selected if we learn the component classifiers on the data of all 16 chromosomes, as indicated by the asterisks. It is also notable

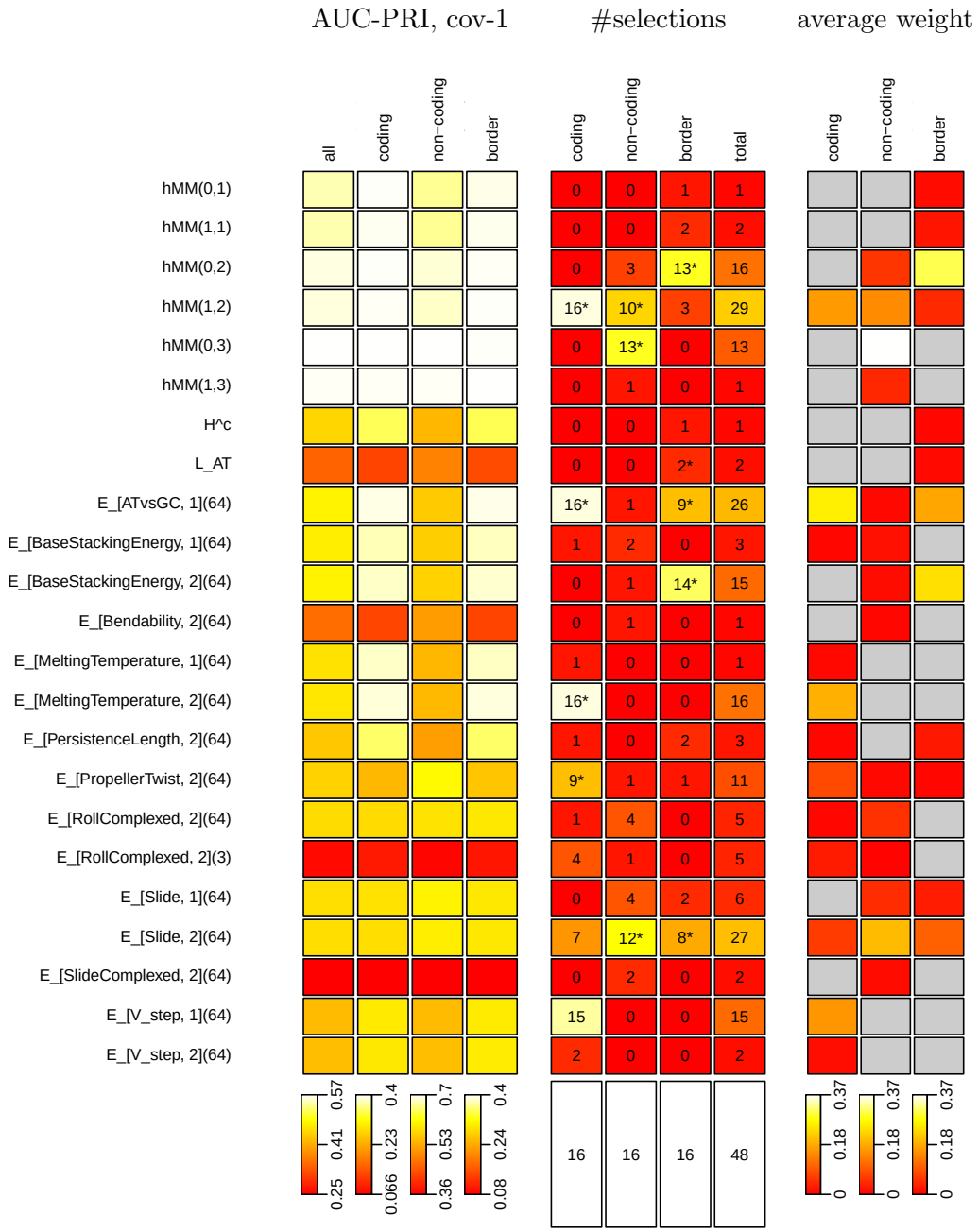


Figure 4.31.: Elementary classifiers selected for the component classifiers for coding, non-coding, and border sequences. In the left block, we visualize the classification performance as measured by AUC-PRI for coverage 1 obtained by a component classifier using only the elementary classifier specified by the row name. The columns of this block visualize AUC-PRI considering all data, only coding sequences, only non-coding sequences, and only border sequences. In the second block, we present the number of iterations of cross validation in which the corresponding elementary classifier is selected for the component classifiers of coding, non-coding, and border sequences, and in a fourth column we sum these numbers over all three types of sequences. At the bottom of the four columns, we give the maximum number that can be achieved in the corresponding column. We mark by an asterisk those elementary classifiers that are selected if we learn the component classifier on the data of all 16 chromosomes. In a third block, we visualize the average weight assigned to elementary classifiers in all 16 iterations of cross validation. We represent by a gray rectangle those elementary classifiers that are never selected for a given type.

that the wavelet energy using slide is frequently selected for the component classifiers although it yields a mediocre classification performance. One might speculate that, although slide alone discriminates nucleosome-bound sequences from linkers worse than many of the other elementary classifiers, it contributes additional information to the classification task that is not fully captured by other selected elementary classifiers.

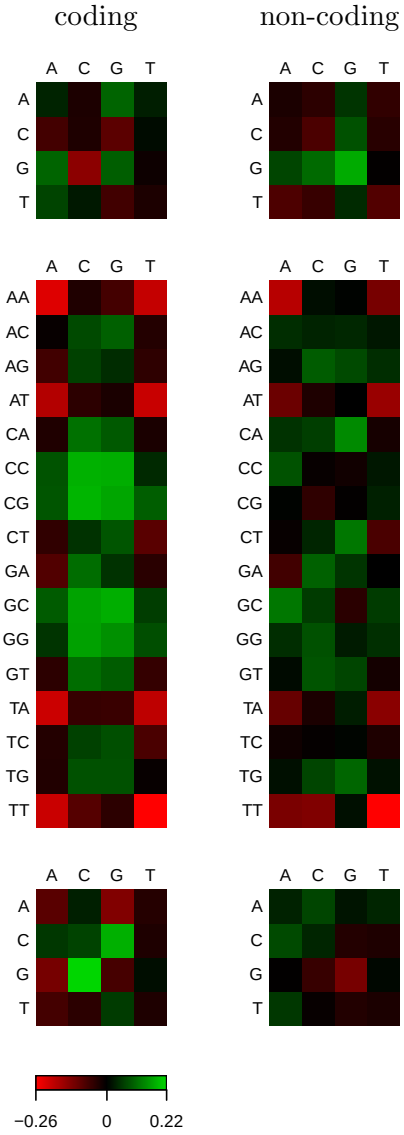
Besides the elementary classifiers mentioned above, some additional elementary classifiers are selected specifically for coding, non-coding, or border sequences. For coding sequences, these are the wavelet energy E_2 computed for melting temperature using a scale of 64, the wavelet energy E_1 computed on the geometrical property V_{step}, and, with lower frequency, the wavelet energy E_2 computed on propeller twist. Except V_{step}, these are also selected when learning the component classifier for coding sequences on the data of all chromosomes. The elementary classifier using homogeneous Markov models of order $d_1 = 0$ and $d_2 = 3$ is specifically selected for non-coding sequences, whereas order $d_2 = 2$ is preferred for border sequences. The latter selection could be interpreted as a combination of the order $d_1 = 0$ at the borders selected for non-coding sequences and the order $d_2 = 2$ at the center selected for coding sequences, since border sequences are expected to share properties of coding and non-coding sequences. The wavelet energy E_2 for base stacking energy with scale 64 is also selected specifically for border sequences. Interestingly, base stacking energy is not selected for coding sequences, although the achieved classification performance on coding and border sequences is comparable.

The average weight of the elementary classifiers within the voting of the component classifiers is visualized in the right block of figure 4.31. Generally, it shows a similar pattern as observed for the number of selections. Notable exceptions are the elementary classifiers using homogeneous Markov models and the wavelet energies for melting temperature and V_{step} for the coding sequences. Although these elementary classifiers are selected in almost all iterations of cross validation they obtain a low weight compared to that of the wavelet energy for A/T vs. G/C dinucleotides. Strikingly, the elementary classifier using homogeneous Markov models of order $d_1 = 0$ and $d_2 = 3$ gains an exceptionally high weight for non-coding sequences.

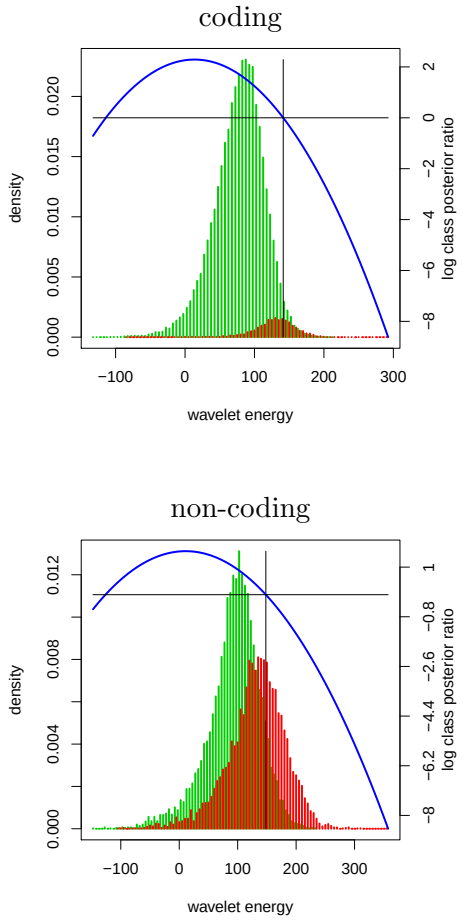
In the following, we scrutinize two of the elementary classifiers for features of nucleosome-bound sequences and linkers, namely the elementary classifier using homogeneous Markov models of order $d_1 = 1$ and $d_2 = 2$, and the elementary classifier using the wavelet energy E_1 for A/T vs. G/C dinucleotides with scale 64.

We present an illustration of the classifier using homogeneous Markov models in figure 4.32(a). For the graphical representation, we compute the stationary dinucleotide distribution at the borders and the stationary trinucleotide distribution at the center for the model representing nucleosome-bound sequences and for the model representing linkers. We then compute the log ratio of the two stationary distributions for nucleosome-bound sequences and linkers for each di- or trinucleotide independently. In figure 4.32(a), positive log ratios, i.e. di- or trinucleotides to which the model for nucleosome-bound sequences assign a higher probability than that for linkers, are represented by green squares, whereas negative log ratios are printed in red. The left column of figure 4.32(a) corresponds to the models learned on coding sequences, and the right column corresponds to the models learned on non-coding sequences.

Considering the stationary distributions of the first order Markov models at the borders of



(a) Log ratios of the stationary distributions of homogeneous Markov models of orders $d_1 = 1$ and $d_2 = 2$ for coding (left) and non-coding (right) sequences. Green colors represent positive log ratios, i.e. di- or trinucleotides to which the stationary distribution for nucleosome-bound sequences assigns higher probabilities than the stationary distribution for linkers, whereas red colors represent negative log ratios. The blocks at the top and at the bottom represent the first order Markov models at the borders, while the block in the middle represents the second order Markov model at the center.



(b) Histogram of wavelet energies E_1 for A/T vs. G/C dinucleotides using scale 64 for coding (top) and non-coding (bottom) sequences. The histogram for nucleosome-bound sequences is plotted in green, whereas the histogram for linkers is plotted in red. The blue line illustrates the log ratio of the class posteriors for nucleosome-bound and linker sequences. Regions where this line lies above 0 are classified as nucleosome-bound sequences using a classification threshold of 0.

Figure 4.32.: Graphical representation of elementary classifiers using homogeneous Markov models and wavelet energies of A/T vs. G/C dinucleotides.

coding sequences, we find opposing preferences for dinucleotides at the two borders. While GC and CG dinucleotides in nucleosome-bound sequences are preferred at the border visualized at the bottom, the same dinucleotides are preferred for linker sequences at the other border. In contrast, we find a slight preference for GA, AG, and GG dinucleotides for nucleosome-bound sequences at the border displayed at the top, whereas these dinucleotides obtain a higher probability in linkers regarding the border presented at the bottom. The preference pattern of GG dinucleotides can also be found for non-coding sequences, for which we do not observe additional strong preferences.

Turning to the stationary distributions of trinucleotides at the center regions displayed in the middle blocks, we find as a common pattern of coding and non-coding sequences a strong preference of AAA and TTT trinucleotides in linker sequences, and a lower preference for the other trinucleotides comprising only A and T nucleotides. This observation is in accordance with previous findings, that poly-A and poly-T tracts are strong indicators of nucleosome depletion (Suter et al., 2000; Yuan et al., 2005; Peckham et al., 2007; Segal and Widom, 2009). This may also explain why the elementary classifiers that model poly-A/T tracts explicitly, i.e. N_{AT} and L_{AT} are seldom selected for the component classifiers, since the preferences modelled by homogenous Markov models may be sufficient for discriminating nucleosome-bound from linker sequences, and homogenous Markov models are often among those elementary classifiers selected first. Interestingly, we also find a preference for CTG and CAG trinucleotides in nucleosome-bound sequences for the non-coding regions, while this preference is less pronounced in coding sequences. Similar to the poly-A/T tracts, this may be one of the reasons why the elementary classifiers that model only the occurrence of these trinucleotides are never selected for any of the component classifiers.

For coding sequences, we find additional strong preferences for trinucleotides comprising only C and G in nucleosome-bound sequences, which can not be observed for non-coding sequences. This may be an indication that coding nucleosome-bound sequences either exhibit a generally higher G/C-content than coding linkers, or comprise a considerably high number of these very trinucleotides. We also find a slight preference for other trinucleotides consisting of two G/C nucleotides and one A/T nucleotide in nucleosome-bound sequences, which might support the former interpretation.

As a second classifier, we consider the elementary classifier using the distribution of wavelet energies E_1 for A/T vs. G/C dinucleotides with a scale of 64 in figure 4.32(b). For coding and non-coding sequences, we plot in each case two histograms, one for the wavelet energies for nucleosome-bound sequences (green) and one for the wavelet energies of linker sequences (red). In addition, we draw the value of the log class posterior ratio as a blue line into the same plot, and we indicate the class border with respect to the wavelet energies for a classification threshold of 0 by a black line.

From the histograms, we conclude that modelling the wavelet energies by a Gaussian density is appropriate. We also observe a considerably larger number of nucleosome-bound sequences than linker sequences in coding regions, whereas both classes occur with approximately the same frequency in non-coding regions. This observation is in accordance with previous findings that coding sequences are often bound in nucleosomes with short linkers between them, while the (non-coding) promoter regions are often depleted for nucleosome to allow the binding of

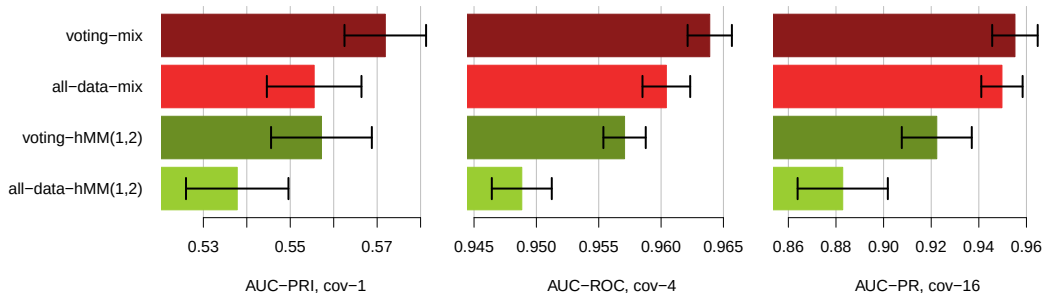


Figure 4.33.: Classification performance as measured by AUC-PRI for coverage 1 (left), AUC-ROC for coverage 4 (center), and AUC-PR for coverage 16 (right) of a weighted voting of component classifiers for coding, non-coding, and border sequences (voting mix) compared to a single component classifier learned on the merged data of all three types of sequences (all-data-mix). In addition, we consider a weighted voting of component classifiers comprising only the elementary classifier using homogeneous Markov models of orders $j = 1$ and $i = 2$ learned on coding, non-coding, and border sequences (voting-hMM(1,2)) and learned on the merged data (all-data-hMM(1,2)).

transcription factors (Yuan et al., 2005; Peckham et al., 2007; Lee et al., 2007). The short linker lengths in coding regions may also result in an artificially low number of linker sequences, as these are filtered for a minimum length of 50 bp beforehand (see section 4.3.3).

4.3.4.3. Influence of differentiating coding and non-coding sequences

As another aspect of voting-mix, we assess the influence of the differentiation into coding, non-coding, and border sequences. To this end, we learn a single component classifier including the selection of elementary classifiers on merged training data sets comprising all three types of sequences in each iteration of the cross validation. We evaluate the classification performance of the joint component classifier (*all-data-mix*) using AUC-PRI as performance measure for a coverage of 1, AUC-ROC for coverage 4, and AUC-PR for coverage 16, and we compare the achieved classification performance to that of voting-mix. These results are presented in figure 4.33. We additionally include component classifiers into the analysis that comprise only the elementary classifier using homogeneous Markov models of orders $d_1 = 1$ and $d_2 = 2$, which turned out to be a frequently selected and reasonably performing elementary classifier in the previous section. We denote the corresponding classifier using a weighted voting by *voting-hMM(1,2)*, and we denote the classifier learned on the merged training data by *all-data-hMM(1,2)*.

Considering the two component classifiers learned using greedy selection of elementary classifiers, we find that the differentiation into coding, non-coding, and border sequences utilized by voting-mix yields a significant improvement over all-data-mix regarding AUC-PRI for coverage 1 and AUC-ROC for coverage 4, whereas the improvement is not significant regarding AUC-PR for coverage 16. This might be an indication that strong nucleosome positioning signals, which are responsible for the high coverage, are less prone to superimposition by G/C-content or coding potential. Hence, these are modelled well by all-data-mix, whereas the weaker signals of lower coverages profit from the differentiation of voting-mix. The improvement gained by voting-hMM(1,2) over all-data-hMM(1,2) is greater than observed for the more complex

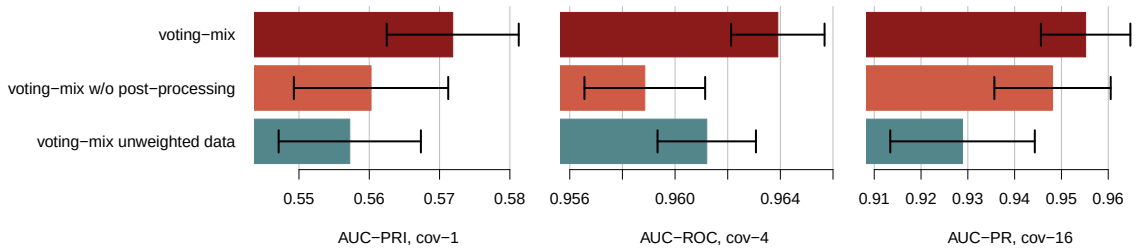


Figure 4.34.: Classification performance as measured by AUC-PRI for coverage 1 (left), AUC-ROC for coverage 4 (center), and AUC-PR for coverage 16 (right) of voting-mix learned on input sequences weighted by the probabilities inferred from coverages and using the post-processing (voting-mix), without post-processing (voting-mix w/o post-processing), and voting mix learned on input sequences exclusively assigned to one of the two classes (voting-mix unweighted data).

component classifiers. In this case, voting-HMM(1,2) yields a significantly higher AUC-PRI for coverage 1, a significantly higher AUC-ROC for a coverage of 4, and a significantly higher AUC-PR for coverage 16 than all-data-hMM(1,2). One reason for this observation might be that the less complex component classifiers especially profit from the increased degrees of freedom induced by the differentiation into coding, non-coding, and border sequences, whereas the component classifiers of all-data-mix are capable of compensating for the loss of differentiation by a greater internal variability. Interestingly, voting-hMM(1,2) using fairly simple component classifiers already performs better than the approach of (Field et al., 2008) regarding all three performance measures (cf. figure 4.29).

4.3.4.4. Influence of weighting and post-processing

We also investigate the contribution of the weighting of data by the probabilities reflecting coverage (see section 4.3.2.7) and of the post-processing step that incorporates preferred linker lengths into the final prediction. To this end, we compare in figure 4.34 the classification performance as measured by AUC-PRI for coverage 1, AUC-ROC for coverage 4, and AUC-PR for coverage 16 of voting-mix to that of voting-mix without the post-processing step, i.e. directly using the class posteriors $P\left(\text{nuc} \,|\, \boldsymbol{x}, \boldsymbol{\beta}\right)$ for the prediction, and to that of voting-mix learned on the input sequences that are either exclusively assigned to the class of nucleosome-bound sequences or to the class of linkers.

Considering the contribution of post-processing to classification performance, we find that utilizing preferred linker lengths is especially beneficial for the lower coverages, as we observe significant differences of AUC-PRI for a coverage of 1 and AUC-ROC for a coverage of 4 regarding voting-mix with and without post-processing. In contrast, the classification performance for a coverage of 16 as measured by AUC-PR does not profit significantly from post-processing. We expect the positioning of nucleosomes to be more rigid for those nucleosome-bound sequences exhibiting a high coverage than for those with a low coverage. Hence, the exact positioning of more loosely positioned nucleosomes may depend on the formation of other nucleosomes in the vicinity and preferred linker lengths *in vivo*. We might speculate that this is the reason why the positioning of nucleosome-bound sequences with a low coverage is also more exact *in silico* when utilizing these dependencies.

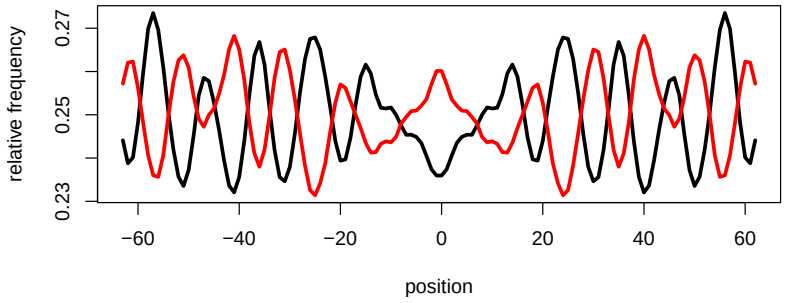
Turning to the contribution of mapping coverage to probabilities and using these probabilities as weights on the input sequences, we find a significant improvement of classification performance with respect to the measures considered for all three levels of coverage compared to the training on unweighted data. This improvement might be expected for the higher coverages, that are under-represented in the training data (see table 4.4), which is partly compensated for by higher weights. The improvement observed for a coverage of 1 is less foreseeable as, for the same reason, we could expect a better adaption of the component classifiers to low coverages if all input sequences obtain the same weight for learning the component classifiers. However, if we assume that the same properties of DNA sequences are responsible for the positioning of strong and weak formation of nucleosomes, these should be more prevalent in sequences exhibiting a higher coverage, which could lead to an improvement of the classification of loosely bound sequences as well.

4.3.4.5. Periodicities

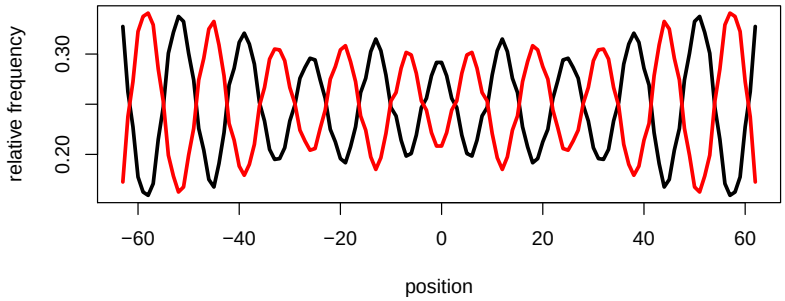
Periodicities of ~ 10 bp regarding different properties of DNA like the geometrical property tip (Richmond and Davey, 2003) or the occurrence of A/T and G/C dinucleotides (Segal et al., 2006; Liu et al., 2008; Field et al., 2008) are widely accepted as being major determinants of nucleosome positioning. We present the periodic pattern of A/T and G/C dinucleotides observed for the data of (Field et al., 2008) in figure 4.35(a). For the generation of the figure, we follow the protocol of (Field et al., 2008): we first select only those reads with a length of 146 to 148 bp. Sequences of even length are included twice with a weight of 0.5 shifting the original sequence 1 bp in each direction. All sequences are considered once in the original orientation and once as their reverse complement. From this set of sequences, we count position-wise the number of occurrences of A/T and G/C dinucleotides and we smooth the obtained relative frequencies by computing the average relative frequencies over three neighboring positions. Finally, the relative frequencies are normalized such that the mean relative frequency of A/T dinucleotides and the mean relative frequency of G/C dinucleotides is equal to 0.25, respectively.

Surprisingly, the greedy selection of elementary classifiers for voting-mix almost never chooses elementary classifiers than can capture such periodicities, i.e. the combination of an inhomogeneous Markov model of order 1 and a homogeneous Markov model of order 4 or the elementary classifiers using wavelet energies for a scale of 3. However, although voting-mix does not exploit periodicities, it yields a superior classification performance compared to the approach of (Field et al., 2008).

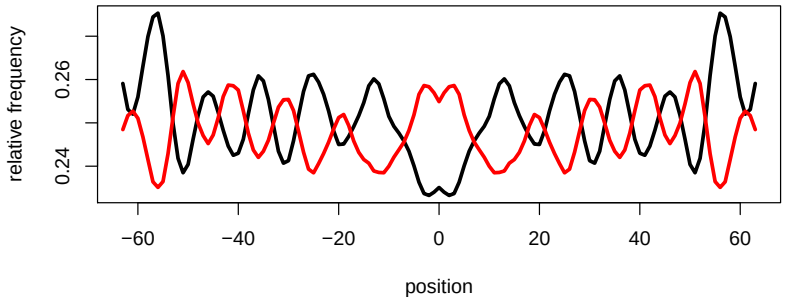
These findings raise the question, how specific for nucleosome-bound sequences the observed periodicities are, and if these could even be an artifact resulting from the digestions by MNase. In figure 4.36, we present a sequence logo of restriction sites of MNase determined from the ends of the mapped reads of (Field et al., 2008). The PWM visualized by the sequence logo is generated by a strand model (see e.g. section 4.3.2.2) learned by the generative ML principle using expectation-maximization. The corresponding DNA sequences are cut by MNase between positions 3 and 4 of the sequence logo. From the sequence logo we observe a mild preference for A/T dinucleotides at the restriction site.



(a) Reads of length 146 − 148 bp from (Field et al., 2008).



(b) Artificially generated sequences with dinucleotide correlations.



(c) Reads of length 167 − 169 bp from (Field et al., 2008).

Figure 4.35.: Periodic patterns of A/T (black line) and G/C (red line) dinucleotides as observed in the data of (Field et al., 2008) considering (a) reads of length 146−148 bp, (c) reads of length 167 − 169 bp, and (b) as observed in artificially generated data using only correlations of dinucleotides and the restriction preference of MNase.

In combination with a strong correlation pattern of dinucleotides, this preference could generally be responsible for periodic patterns as observed in figure 4.35(a). We demonstrate the validity of the latter proposition by simulations. To this end, we randomly generate sequences by drawing dinucleotides from a homogeneous Markov model of order 3 with strong correlations between the dinucleotides, such that we obtain three consecutive dinucleotides of the same type, i.e. either A/T or G/C dinucleotides, with high probability, and afterwards switch the type of dinucleotides with a similarly high probability. Additionally, we ensure that the relative frequencies at the borders of the generated sequences are in accordance with the preference for A/T dinucleotides at the restriction site of MNase. We present the periodic patterns found for these artificial data in figure 4.35(b). Obviously, strongly periodic sequences can be

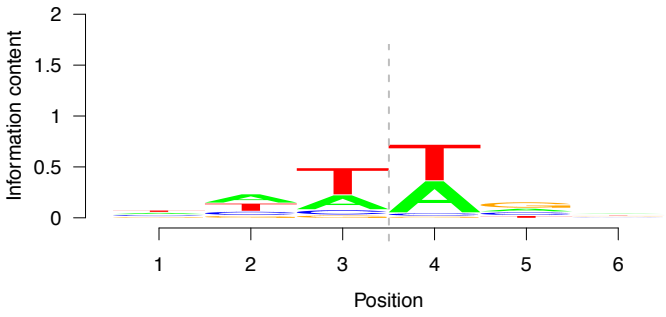


Figure 4.36.: Sequence logo of a strand model learned on the restriction sites of MNase. MNase cute the sequence between positions 3 and 4 as indicated by the dashed grey line.

generated by a homogeneous Markov model using only correlations between dinucleotides, if we additionally fix the relative frequencies of dinucleotides at the borders, although these occur with a period of ~ 13 bp in this example.

In order to check if the periodicities present in figure 4.35(a) are due to correlations or due to specific properties of nucleosome-bound sequences, we conduct a simple test. If the periodicities would be the result of correlations combined with the restriction preference of MNase, we should observe similar patterns for other lengths of reads. If, on the other hand, these periodicities are related to the positioning of nucleosomes, they should be less articulate for longer reads, since nucleosome positions can appear shifted within the longer reads and for this reason are not perfectly aligned. Hence, we extract reads of length 167 to 169 from the data of (Field et al., 2008) and apply the same protocol to these as to the shorter reads before. The results of this analysis are depicted in figure 4.35(c). We find that a considerable part of the periodic pattern we observed in figure 4.35(a) is lost at the center region between positions -50 and $+50$ for reads of length $167 - 169$ bp. This might be an indication that ~ 10 bp periodicities are relevant for nucleosome positioning. However, we do not observe an overall extinction of the periodic signal. Since voting-mix yields a superior classification performance compared to the approach of (Field et al., 2008), we might speculate that periodic signals mostly control local nucleosome positioning, whereas the general tendency of a sequence to be bound in a nucleosome is mostly controlled by other properties of DNA.

4.3.4.6. Evaluation of predictions

Finally, we examine the predictions of voting-mix and the approach of (Field et al., 2008) in their genomic contexts. On the one hand, we consider the five genomic regions that are also presented in (Field et al., 2008), and on the other hand, we choose five additional regions where the predictions of the two approaches differ considerably. For both approaches, the prediction of regions covered by nucleosomes depend on the classification threshold. We choose this threshold such that both approaches yield a sensitivity of 0.95 for a coverage of 4, i.e. correctly recover 95% of the nucleosome-bound sequences covered by at least 4 reads. In figures 4.37 and 4.38, we plot the model scores of (Field et al., 2008) as a solid blue line, and we plot the probabilities of nucleosome formation of voting-mix as a solid red line. The scores of voting-mix appear smoother than those of the approach of (Field et al., 2008) in the figures due to

the post-processing, which is not the case for the original scores of voting-mix (not shown). For the approach of (Field et al., 2008), we additionally include the scores of the Genomica file available at `http://genie.weizmann.ac.il/pubs/field08/field08_genomes.html` as a dashed blue line, since these were also used in the figures of (Field et al., 2008). We indicate regions that are covered by nucleosomes according to the prediction using the previously chosen thresholds as straight blue and red lines, respectively. If one of the approaches predicts a nucleosome-bound sequence centered around some position ℓ, we assume that a stretch of DNA of length 147 is bound in this nucleosome and, hence, positions $\ell - 73$ to $\ell + 73$ are considered as covered. As a reference, we also include the positions of the mapped reads into the figure, and we add yellow arrows representing position and orientation of genes within the genomic regions considered.

For positions 299000 to 300500 on chromosome 2, we do not find notable differences between the predictions of the two approaches. Both predict a nucleosome-free region in the potential promoter region of the gene *RPL4A*, while the remainder of these positions is predicted to be covered by nucleosomes. Another region within the gene *YBR030W* that is not covered by nucleosomes according to the mapped reads obtains scores below the threshold for both approaches. However, the corresponding segments are not long enough to open a nucleosome-free region.

In contrast, we do find differences between the predictions for positions 1107500 to 1109000 on chromosome 4. Voting mix predicts two short nucleosome-free regions in the vicinity of positions 1107500 and 1108500, respectively. The latter, very short nucleosome-free region is supported by the mapped reads and is similarly discovered by the approach of (Field et al., 2008), whereas the former appears to be incorrect. Considering the mapped reads in this region, we might conclude that the nucleosomes around position 1107500 are only loosely positioned, which could lead to a weaker signal of nucleosome positioning in the sequence and, hence, to the false prediction by voting-mix. A weaker signal in this region is also detected by the approach of (Field et al., 2008), but most of the corresponding scores are still above the chosen threshold.

Turning to positions 341800 to 343300 of chromosome 5, we find a large number of mapped reads within the putative promoter region of the genes *MET6* and *IES5*. These nucleosome-covered positions and a short nucleosome-free region in the vicinity of the TSS of *MET6* are correctly discovered by both approaches. In contrast, a broad nucleosome-free region in the putative promoter region of the neighboring gene *IES5* is only predicted by voting-mix. Since nucleosome-bound and nucleosome-free regions in promoters can be used to exclude false-positive predictions of transcription factor binding sites, we consider differences in such regions especially important.

The last genomic regions presented in figure 4.37 are positions 126800 to 128300 of chromosome 9. In this region, we find a fairly long intergenic region between the two genes *AYR1* and *SIM1*. Voting-mix predicts two nucleosome-free regions around position 127500 and approximately at position 127900, which are both supported by the mapped reads. In contrast, the approach of (Field et al., 2008) detects only a small portion of the former nucleosome-free region. However, position 127500 is possibly located upstream of the promoter region of *SIM1* and this difference

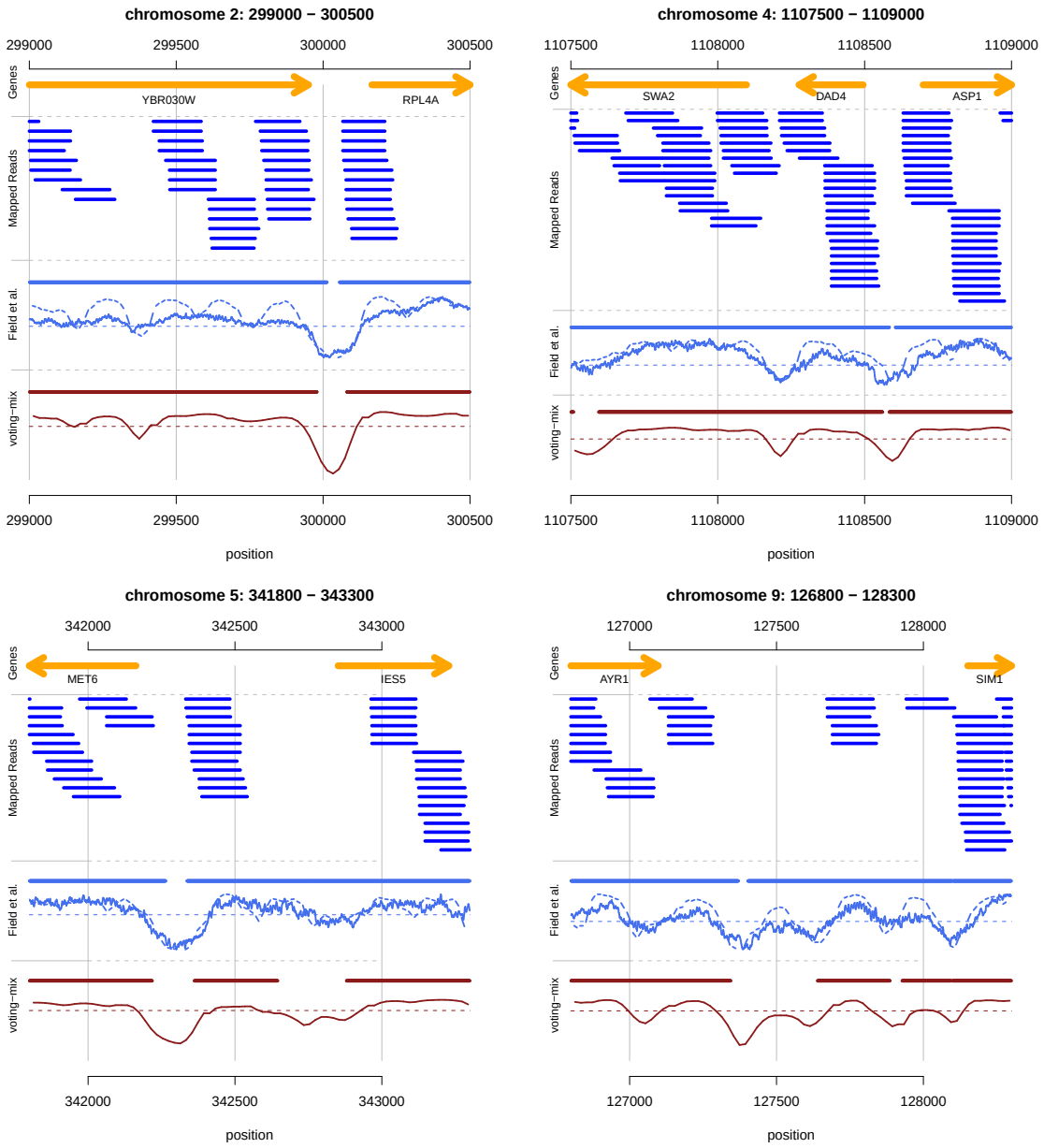


Figure 4.37.: Predictions of voting-mix and the approach of (Field et al., 2008) in their genomic contexts. In a first block we plot genes as yellow lines, where the arrowheads indicate the orientation of the gene. In the next block, we plot the reads of nucleosome bound sequences mapped onto the genome by (Field et al., 2008). In the last two blocks, we illustrate the predictions of the approach of (Field et al., 2008) (blue) and voting-mix (red). In each case, we plot a straight line over regions that are predicted to be covered by nucleosomes, and we also plot the scores that are considered for the prediction. The additional dashed blue curve corresponds to the scores of the Genomica file provided by (Field et al., 2008) (see also section 4.3.4.1). The straight dashed red and blue lines indicate the thresholds that are used for the prediction of the two approaches.

might be less relevant for excluding false positive predictions of transcription factor binding sites.

In figure 4.38, we present the predictions for six additional genomic regions. Positions 264800 to 266300 of chromosome 10 comprise the potential promoter regions of a head-to-head configu-

ration of two genes, namely *DPB11* and *SIP4*. Comparing the location of the mapped reads to the predictions of the two approaches, we find that both fail to reliably identify nucleosome-covered and nucleosome-free regions. Regions that are predicted to be nucleosome-free are covered by several of the mapped reads, and the short nucleosome-free region before the TSS of *DPB11* is only partly recovered by voting-mix. However, since the scores of the two approaches are not articulately contradicting, we might speculate that nucleosome positioning in this region can not be fully explained by signals detected from sequence, but also depends on other, epi-genetic properties like methylation patterns.

Considering positions 359000 to 360500 of chromosome 10, we again find largely consistent predictions of the two approaches. The only exception is a short nucleosome-free region approximately at position 360000, which is only discovered by voting-mix, and which is also supported by the mapped reads. In the potential promoter region of the head-to-head genes *GYP6* and *YJL043W*, we find a large number of consistently positioned reads, which are also reflected by the scores of both approaches.

Positions 409800 to 411300 of chromosome 12 comprise a putative promoter of the gene *PDC5*, which is widely covered by nucleosomes according to the mapped reads. Nonetheless, both approaches predict nucleosome-free regions in this promoter. The nucleosome-free region predicted by the approach of (Field et al., 2008) around position 410550 is covered by six mapped reads and, hence, most probably incorrect. In contrast, the nucleosome-free region according to voting-mix is only covered by a few, loosely positioned reads, which again might lead to a weaker signal of nucleosome positioning.

Turning to the predictions for positions 666800 to 668300 of chromosome 13, we find another nucleosome-free region that is only predicted by voting-mix. In this case, we can not clearly distinguish from the mapped reads, if this prediction is correct or not. The remainder of this region is highly covered by mapped reads. This is reflected by the scores of both approaches, which consistently stay above the chosen threshold.

As another example, we examine positions 830500 to 832000 of the same chromosome. Here, voting-mix predicts a broad nucleosome-free region spanning a large fraction of the potential promoter of the gene *CAT8*. This nucleosome-free region is not discovered by the approach of (Field et al., 2008), which strongly predicts a nucleosome located around position 831500. If we use the predicted nucleosome-covered and nucleosome-free regions for excluding of false positive transcription factor binding sites, the prediction of voting-mix would clearly give a more accurate result in this case.

As a last genomic region, we consider positions 17500 to 19000 of chromosome 14. For this region, we find notable differences between the two approaches as well. Voting-mix predicts a broad nucleosome-free region around position 17750, which is clearly not supported by the mapped reads. In contrast, the approach of (Field et al., 2008) correctly predicts these positions as nucleosome-covered.

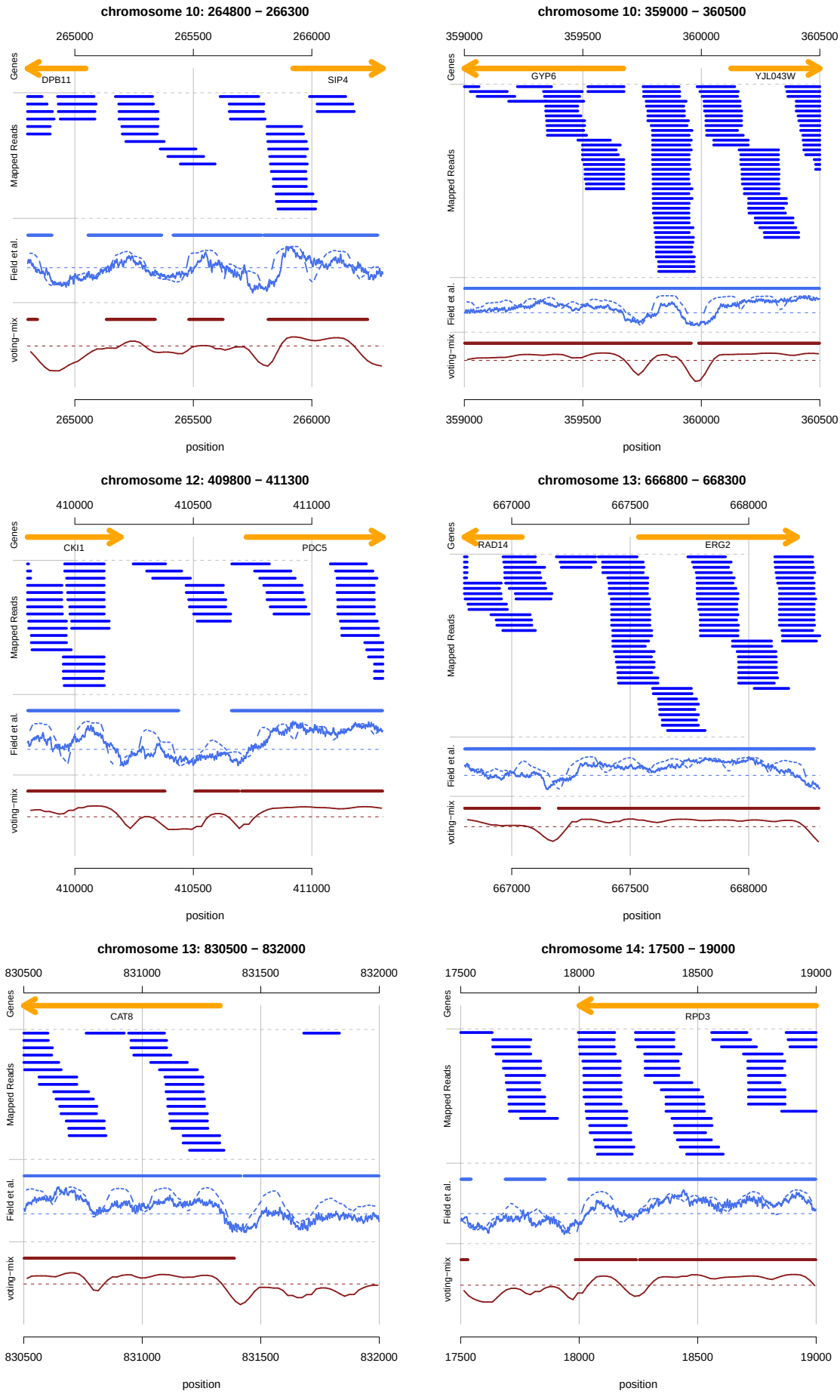


Figure 4.38.: Predictions of voting-mix and the approach of (Field et al., 2008) in their genomic contexts (cf. figure 4.37).

4.3.5. Conclusions

In this section, we present a novel approach for discriminating nucleosome-bound sequences from linkers that differentiates between coding and non-coding sequences and automatically selects elementary classifiers representing different aspects of nucleosome formation. These elementary classifiers are combined in component classifiers for coding, non-coding, and border sequences, which in turn are combined in a weighted voting to yield the final probability of nucleosome formation. We assess the classification performance of this approach by means of AUC-ROC, AUC-PR, and AUC-PRI, and we find that it consistently outperforms the current state-of-the-art approach of (Field et al., 2008). Scrutinizing the selected elementary classifiers, we find several known features of nucleosome-bound sequences and linkers represented, e.g. a preference for poly-A/T in linkers, and a preference for CAG/CTG and general G/C-rich trinucleotides in nucleosome-bound sequences. However, periodicities, which are considered important features of nucleosome-bound sequences, are not covered by the selected elementary classifiers. Against the background of superior classification performance, we may speculate that these periodicities are relevant for local nucleosome positioning, but less important for the general potential of a sequence to be bound in a nucleosome. Considering the predictions of the novel approach and the approach of (Field et al., 2008) in their genomic context, we find notable differences that become especially relevant when using predicted nucleosome-covered regions to eliminate non-functional predictions of transcription factor binding sites.

4.4. Recognition of donor splice sites

The computational prediction of splice sites has been in the focus of bioinformatics research for more than two decades. The two main sites of interest are donor splice sites at the 5' end and acceptor splice sites at the 3' end of introns. Acceptor splice sites have an AG at positions -2 and -1 relative to the 3' end of the intron, whereas donor splice sites exhibit a canonical consensus GT or a non-canonical GC at positions 1 and 2 of the intron.

4.4.1. Background

A first statistical approach for the prediction of donor and acceptor splice sites is proposed by Staden (1984), who uses position weight matrices (PWMs) learned by the generative maximum likelihood (ML) principle for modelling both types of sites. PWM models entail the assumption that the nucleotides at each position occur statistically independently. Zhang and Marr (1993) extend the PWM model to dinucleotides and, hence, to dependencies between directly adjacent positions. The proposed weight array matrix (WAM) model is applied to splice donor sites of *Schizosaccharomyces pombe* (see also section 4.1.1). WAM models, which are equivalent to first order Markov models, are also used by Salzberg (1997) for the recognition of eukaryotic donor and acceptor splice sites. Zhao et al. (2005) predict donor splice sites from SpliceDB (Burset et al., 2001) with permuted variable length Markov models (PVLMMs) which extend Markov models to non-adjacent dependencies as well as context-specific orders. PVLMMs are described in more detail in section 4.1.1.

The maximal dependence decomposition (MDD) algorithm (Burge, 1998) combines a decision tree with a number of PWM models at its leaves to represent the heterogeneity of donor splice sites. Each inner node of the decision tree represents a binary decision: If the nucleotide at position ℓ in a sequence $\boldsymbol{x}$ is equal to the consensus, we proceed to the *consensus child* and to the *non-consensus child* otherwise. When we finally reach a leaf of the decision tree, we score the sequence according to the PWM model at this leaf and the probability of the path from the root to that leaf. The structure of the decision tree is learned by a greedy algorithm. Burge (1998) employs the MDD algorithm for the prediction of human donor splice sites and also proposes a gene finder (Burge and Karlin, 1997), which employs the decision tree of MDD as one of its components. The decision tree model is also one of the bases of the *maximum supervised posterior decomposition* (MSPD) algorithm presented in this work.

Yeo and Burge (2004) propose maximum entropy models (MEM) for the prediction of human donor splice sites. MEMs correspond to Markov random fields (MRFs) learned by the generative ML principle. Yeo and Burge (2004) find that, among the considered MEMs, the model comprising all two-point dependencies between positions yields the best accuracy for the prediction of human splice donor sites.

With focus on improved learning principles, Keilwagen et al. (2007) learn Markov models of different orders by the discriminative maximum conditional likelihood (MCL) principle. They show that discriminatively learned Markov models can achieve an improved classification accuracy as compared to (Yeo and Burge, 2004) on data sets of human splice donor and acceptor sites also considered by Yeo and Burge (2004).

4.4.2. Maximum supervised posterior decomposition

In this work, we propose *maximum supervised posterior decomposition* (MSPD) for the computational prediction of donor splice sites. MSPD combines the decision tree model of the MDD algorithm (Burge and Karlin, 1997) with the discriminative maximum supervised posterior (MSP) principle. Again, we assume a training data set of sequences $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and associated class labels $\boldsymbol{c} = (c_1, \ldots, c_N)$.

Burge (1998) defines a binary decision tree having PWM models (see section 3.3.1) at its leaves. Each inner node of the decision tree holds a *consensus nucleotide* $K \in \Sigma$ and a *split position* $\ell \in 1, \ldots, L$, where L denotes the length of the donor splice sites. The consensus nucleotide K is defined as that nucleotide which occurs most frequently at position ℓ. We denote the set of non-consensus nucleotides by $\overline{K} = \Sigma \backslash \{K\}$. Each inner node has exactly two children: one *consensus child* and one *non-consensus* child. For an input sequence $\boldsymbol{x}$, each inner node represents a binary decision: If the nucleotide x_ℓ at the split position ℓ is equal to the consensus nucleotide at the current inner node, i.e. $x_\ell = K$, this sequence is modelled in the subtree below the consensus child. If this nucleotide is included in the set of non-consensus nucleotides, i.e. $x_\ell \in \overline{K}$, it is modelled in the subtree below the non-consensus child. Starting at the root, we can apply these binary decisions in a recursive manner, until we reach a leaf b. Thus the decision tree defines a partitioning of the data, where each partition is modelled by an independent PWM model.

Figure 4.39 illustrates the partitioning of the data according to the binary decisions at the inner nodes. In this example, the first split at the root is conducted for a "T" at position +6. The consensus child of the root is already a leaf, which is responsible for all sequences with a "T" at position +6 regardless of the nucleotides at other positions. The non-consensus child, however, is another inner node, which partitions for an "A" at position -2. Hence, all sequences that contain no "T" at position +6 are partitioned again according to this split.

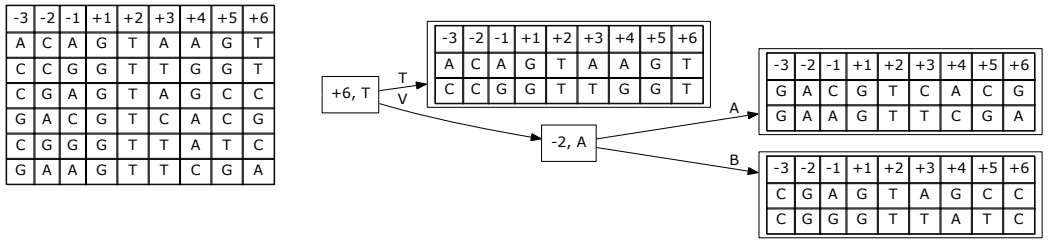


Figure 4.39.: Decision tree with two inner nodes and three leaves. The sequences of the left table are partitioned according to the decisions represented by the inner nodes. The resulting partitions are itemized in the leaves.

The path from the root and consequently the considered leaf depends on the sequence. Additionally, we allow different structures of the decision trees in different classes and we denote the structure of the decision tree in class c by τ_c. To reflect these dependencies, we denote the leaf that we reach for a concrete sequence $\boldsymbol{x}$ in the decision tree of class c by $b(\tau_c, \boldsymbol{x})$.

We define the *leaf probability* $P(b|\boldsymbol{\xi})$ as the probability of reaching leaf b, or equivalently, the probability of traversing the path from the root to that leaf. With these prerequisites, we define the likelihood of sequence $\boldsymbol{x}$ and class c given the parameters $\boldsymbol{\xi}$ of the decision tree and the PWM models at its leaves as

$$P\left(\boldsymbol{x}, c \,|\, \tau_c, \boldsymbol{\xi}\right) \;=\; P(c|\boldsymbol{\xi}) \cdot P(b(\tau_c, \boldsymbol{x})|c, \boldsymbol{\xi}) \cdot P\left(\boldsymbol{x} \,|\, c, b(\tau_c, \boldsymbol{x}), \boldsymbol{\xi}\right), \tag{4.69}$$

where $P(c|\boldsymbol{\xi})$ denotes the a-priori probability of class c, $P(b(\tau_c, \boldsymbol{x})|c, \boldsymbol{\xi})$ denotes the leaf probability of the leaf $b(\tau_c, \boldsymbol{x})$ for sequence $\boldsymbol{x}$ and class c, and $P\left(\boldsymbol{x} \,|\, c, b(\tau_c, \boldsymbol{x}), \boldsymbol{\xi}\right)$ corresponds to the likelihood defined by the PWM model at leaf $b(\tau_c, \boldsymbol{x})$.

MSPD comprises two tasks: Learning the structures τ_c of the decision trees in the classes $c \in \mathcal{C}$, and learning the parameters $\boldsymbol{\beta}$ of these decision trees. We learn the structures of the decision trees by a greedy algorithm which requires optimized parameters in each step. Hence, we start the presentation of our algorithm with learning the parameters $\boldsymbol{\xi}$ for fixed structures τ_c.

4.4.2.1. Parameter estimation

We learn the parameters $\boldsymbol{\xi}$ by the discriminative MSP principle as defined in section 3.2.2, i.e.

$$\boldsymbol{\xi}^*_{\mathrm{MSP}} \;=\; \operatorname*{argmax}_{\boldsymbol{\xi}} \left[\prod_{n=1}^{N} \frac{P\left(\boldsymbol{x}_n, c_n \,|\, \tau_{c_n}, \boldsymbol{\xi}\right)}{\sum_{\tilde{c}} P\left(\boldsymbol{x}_n, \tilde{c} \,|\, \tau_{\tilde{c}}, \boldsymbol{\xi}\right)} \right] q\left(\boldsymbol{\xi} \,|\, \boldsymbol{\alpha}\right). \tag{4.70}$$

Like for the other models considered in this work, MSP estimation for MSPD must be carried out numerically. To this end, we parameterize the model in analogy to the parameterization of Markov models described in section 3.3.1. The likelihoods $P\left(\boldsymbol{x} \,|\, c, b(\tau_c, \boldsymbol{x}), \boldsymbol{\xi}\right)$ of the PWM models at the leaves are parameterized as

$$P\left(\boldsymbol{x} \,|\, c, b(\tau_c, \boldsymbol{x}), \boldsymbol{\xi}\right) \;=\; \frac{1}{Z_{b(\tau_c,\boldsymbol{x})|c}(\boldsymbol{\xi})} \exp\left(\sum_{\ell=1}^{L} \xi_{\ell, x_\ell | b(\tau_c,\boldsymbol{x}), c} \right), \tag{4.71}$$

where the normalization constant for leaf b is defined as

$$Z_{b|c}(\boldsymbol{\xi}) \;=\; \sum_{\boldsymbol{x} \in \Sigma^L} \delta_{b, b(\tau_c,\boldsymbol{x})} \exp\left(\sum_{\ell=1}^{L} \xi_{\ell, x_\ell | b, c} \right) \tag{4.72}$$

and corresponds to the normalization constant defined in equation (3.37) in section 3.3.1. We further decompose the leaf probabilities into probabilities of traversing the edges on the path from the root to that leaf. Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_d)$, denote the path from the root π_1 to leaf $b = \pi_d$, and let $P(\pi_i|\pi_{i-1})$ denote the probability of going to child π_i after having visited the parent node π_{i-1} on that path. The probability of leaf b is then defined as

$$P(b|c, \boldsymbol{\xi}) \;=\; P(\pi_1|c, \boldsymbol{\xi}) \prod_{i=2}^{d} P(\pi_i|\pi_{i-1}, c, \boldsymbol{\xi}), \tag{4.73}$$

where $\pi_d = b$ and the probability of starting at the root $P(\pi_1|c, \boldsymbol{\xi})$ is always 1 and can, hence, be omitted.

We parameterize the probabilities $P(\pi_i|\pi_{i-1}, c, \boldsymbol{\xi})$, i.e. the probability of visiting a child π_i from its direct parent π_{i-1}, as

$$P(\pi_i|\pi_{i-1}, c, \boldsymbol{\xi}) = \frac{\exp\left(\xi_{\pi_i|c}\right) Z_{\pi_i|c}(\boldsymbol{\xi})}{Z_{\pi_{i-1}|c}(\boldsymbol{\xi})}, \quad \text{where} \quad Z_{\pi_{i-1}|c}(\boldsymbol{\xi}) = \sum_{\tilde{\pi}_i} \exp\left(\xi_{\tilde{\pi}_i|c}\right) Z_{\tilde{\pi}_i|c}(\boldsymbol{\xi}), \quad (4.74)$$

and $\tilde{\pi}_i$ denotes the consensus and non-consensus children of node π_{i-1}. If π_i is the leaf of the current path, i.e. $\pi_i = \pi_d$, the normalization constants $Z_{\tilde{\pi}_i|c}(\boldsymbol{\xi})$ are equal to the normalization constants $Z_{b|c}(\boldsymbol{\xi})$ of equation (4.72) of the corresponding leaves.

We insert these definitions into equation (4.73) to obtain the leaf probabilities in terms of $\boldsymbol{\xi}$ parameters

$$\begin{aligned} P(b|c, \boldsymbol{\xi}) &= \prod_{i=2}^{d} \frac{\exp\left(\xi_{\pi_i|c}\right) Z_{\pi_i|c}(\boldsymbol{\xi})}{Z_{\pi_{i-1}|c}(\boldsymbol{\xi})} && (4.75) \\ &= \frac{1}{Z_{\pi_1|c}(\boldsymbol{\xi})} \left[\prod_{i=2}^{d} \exp\left(\xi_{\pi_i|c}\right)\right] Z_{\pi_d|c}(\boldsymbol{\xi}) && (4.76) \end{aligned}$$

and find that all of the normalization except the normalization constants $Z_{\pi_1|c}(\boldsymbol{\xi})$ of the root and $Z_{\pi_d|c}(\boldsymbol{\xi}) = Z_{b|c}(\boldsymbol{\xi})$ of the leaf b cancel.

The normalization constant of the root node, i.e. $Z_{\pi_1|c}(\boldsymbol{\xi})$, is also the normalization constant of the tree τ_c of class c, which we define as

$$Z_c(\boldsymbol{\xi}) := Z_{\pi_1|c}(\boldsymbol{\xi}). \quad (4.77)$$

With these prerequisites, we can finally define the parameterization of the class probabilities $P(c|\boldsymbol{\xi})$ as

$$P(c|\boldsymbol{\xi}) = \frac{\exp\left(\xi_c\right) Z_c(\boldsymbol{\xi})}{Z(\boldsymbol{\xi})}, \quad \text{where} \quad Z(\boldsymbol{\xi}) = \sum_{\tilde{c} \in \mathcal{C}} \exp\left(\xi_{\tilde{c}}\right) Z_{\tilde{c}}(\boldsymbol{\xi}). \quad (4.78)$$

We insert these definitions into the likelihood of equation (4.69) and obtain

$$\begin{aligned} P\left(\boldsymbol{x}, c \,|\, \tau_c, \boldsymbol{\xi}\right) &= \frac{\exp\left(\xi_c\right) Z_c(\boldsymbol{\xi})}{Z(\boldsymbol{\xi})} \cdot \frac{1}{Z_{\pi_1|c}(\boldsymbol{\xi})} \left[\prod_{i=2}^{d} \exp\left(\xi_{\pi_i|c}\right)\right] \cdot \\ &\qquad Z_{\pi_d|c}(\boldsymbol{\xi}) \cdot \frac{1}{Z_{b(\tau_c, \boldsymbol{x})|c}(\boldsymbol{\xi})} \exp\left(\sum_{\ell=1}^{L} \xi_{\ell, x_\ell | b(\tau_c, \boldsymbol{x}), c}\right). \quad (4.79) \end{aligned}$$

As $Z_{\pi_1|c}(\boldsymbol{\xi}) = Z_c(\boldsymbol{\xi})$ and π_d denotes the leaf $b(\tau_c, \boldsymbol{x})$, we can further simplify the definition of the likelihood, yielding

$$P\left(\boldsymbol{x}, c \,|\, \tau_c, \boldsymbol{\xi}\right) = \frac{1}{Z(\boldsymbol{\xi})} \exp\left(\xi_c\right) \cdot \left[\prod_{i=2}^{d} \exp\left(\xi_{\pi_i|c}\right)\right] \cdot \exp\left(\sum_{\ell=1}^{L} \xi_{\ell, x_\ell | \pi_d, c}\right). \quad (4.80)$$

As for the Markov models, the normalization constants $Z(\boldsymbol{\xi})$ cancel as well, if we define the class posterior $P(c \mid \boldsymbol{x}, \boldsymbol{\xi})$:

$$P(c \mid \boldsymbol{x}, \boldsymbol{\tau}, \boldsymbol{\xi}) = \frac{\exp\left(\xi_c + \sum_{i=2}^{d} \xi_{\pi_i|c} + \sum_{\ell=1}^{L} \xi_{\ell,x_\ell|\pi_d,c}\right)}{\sum_{\tilde{c}\in\mathcal{C}} \exp\left(\xi_{\tilde{c}} + \sum_{i=2}^{\tilde{d}} \xi_{\tilde{\pi}_i|\tilde{c}} + \sum_{\ell=1}^{L} \xi_{\ell,x_\ell|\tilde{\pi}_{\tilde{d}},\tilde{c}}\right)}, \tag{4.81}$$

where $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_K), K = |\mathcal{C}|$. Although we waive to denote it explicitly, the paths in the decision trees and, consequently, d, $\tilde{d}$, π_i, and $\tilde{\pi}_i$ depend on the leaf $b(\tau_c, \boldsymbol{x}_n)$ chosen for sequence $\boldsymbol{x}_n$ in the decision tree τ_c.

Wettig et al. (2003) prove that the log conditional likelihood of Markov models in this parameterization (see section 3.3.1) is a concave function of the parameters $\boldsymbol{\xi}$. We can easily see that the same can be proven for the MSPD decision trees in the parameterization defined above. To this end, we compare equation (3.30) (p. 19) to equation (4.81) and find that both exhibit the same kind of functional dependence on the parameters $\boldsymbol{\xi}$. Hence, the result of (Wettig et al., 2003) can be directly transferred to the decision tree models of MSPD and we obtain optimal parameters $\boldsymbol{\xi}^*_{\mathrm{MCL}}$ using the MCL principle regardless of the initialization. A more general proof that also includes log concavity of the prior defined in the following is given in appendix A.1.

We want to use the MSP principle for estimating the optimal parameters $\boldsymbol{\xi}^*_{\mathrm{MSP}}$. To this end, we need to define a prior $q(\boldsymbol{\xi} \mid \boldsymbol{\alpha})$ with hyper-parameters $\boldsymbol{\alpha}$ on the parameters $\boldsymbol{\xi}$. In section 3.4.2, we defined a transformed product-Dirichlet prior for Markov models in $\boldsymbol{\xi}$ parameterization. This prior can be used almost directly for the parameters of the PWM models at the leaves, because these are just Markov models of order 0, and we use the same parameterization as defined in section 3.3.1. However, the splitting into consensus and non-consensus branch entails that we can observe only the consensus nucleotide K in the consensus branch, whereas K can never be observed in the non-consensus branch. We deal with this specialty by i) excluding the split position from the prior in the consensus branch and ii) reducing the alphabet – and consequently the number of parameters – to $\bar{K}$ in the non-consensus branch. The first can be justified by the insight that the consensus nucleotide appears deterministically in the consensus branch and, hence, no uncertainty in the parameter estimation needs to be modelled. The probability of K is fixed to 1, while the probabilities of the other symbols $x \in \bar{K}$ are fixed to 0. As another consequence of the split, the consensus nucleotide cannot appear in the non-consensus branch. Hence, the reduction of the alphabet at the split position in the non-consensus branch is reasonable as well.

We define the prior for the probabilities of the outgoing edges of node π_{i-1} as a transformed Beta prior, where the Beta distribution is the specialization of the Dirichlet distribution for one free parameter. Let π_i denote the consensus child of π_{i-1} and let π'_i denote the non-consensus

child of π_{i-1}. Then the prior $q\left(\xi_{\pi_i|c}, \xi_{\pi'_i|c} \,\middle|\, \alpha_{\pi_i|c}, \alpha_{\pi'_i|c}\right)$ is defined as

$$q\left(\xi_{\pi_i|c}, \xi_{\pi'_i|c} \,\middle|\, \alpha_{\pi_i|c}, \alpha_{\pi'{}_i|c}\right) := \Gamma(\alpha_{\pi_i|c} + \alpha_{\pi'_i|c}) \frac{1}{\Gamma(\alpha_{\pi_i|c})} \left(\frac{\exp\left(\xi_{\pi_i|c}\right) Z_{\pi_i|c}(\boldsymbol{\xi})}{Z_{\pi_{i-1}|c}(\boldsymbol{\xi})}\right)^{\alpha_{\pi_i|c}} \cdot \frac{1}{\Gamma(\alpha_{\pi'_i|c})} \left(\frac{\exp\left(\xi_{\pi'_i|c}\right) Z_{\pi'_i|c}(\boldsymbol{\xi})}{Z_{\pi_{i-1}|c}(\boldsymbol{\xi})}\right)^{\alpha_{\pi'_i|c}}. \tag{4.82}$$

As the parameterization of the class probabilities (see equation (4.78)) is the same as in the case of Markov models (see section 3.3.1, equation (3.31)), we can again apply the Dirichlet prior of section 3.4.2 to the parameters ξ_c.

We choose the hyper-parameters $\boldsymbol{\alpha}$ according to the assumption of uniform pseudo data in analogy to Markov models (see section 3.4.2). The hyper-parameter α_c is equal to the equivalent sample size of class c. Again, we assume uniformly distributed pseudo data. This assumption implies that before the first split each nucleotide appears $\frac{\alpha_c}{|\Sigma|}$ times at each position in the set of pseudo data. Accordingly, we choose the hyper-parameter of the consensus child π_2 of the root π_1 as $\alpha_{\pi_2|c} := \frac{\alpha_c}{|\Sigma|}$ and the hyper-parameter of the non-consensus child π'_2, which is responsible for the remaining $|\Sigma| - 1$ nucleotides, as $\alpha_{\pi'_2|c} := \frac{\alpha_c(|\Sigma|-1)}{|\Sigma|}$. In case of a DNA-alphabet, this amounts to $\alpha_{\pi_2|c} = \frac{1}{4}\alpha_c$ and $\alpha_{\pi'_2|c} = \frac{3}{4}\alpha_c$. With the same reasoning, we generally define the hyper-parameters for the probabilities of the consensus child π_i and the non-consensus child π'_i given their common parent π_{i-1} as

$$\alpha_{\pi_i|c} := \frac{\alpha_{\pi_{i-1}|c}}{|\Sigma|} \qquad \text{and} \qquad \alpha_{\pi'_i|c} := \frac{\alpha_{\pi_{i-1}|c}\,(|\Sigma| - 1)}{|\Sigma|}. \tag{4.83}$$

Finally, the equivalent sample size of the PWM at a leaf $b = \pi_d$ is equal to $\alpha_{\pi_d|c}$ and consequently (see section 3.4.2) the hyper-parameters of the parameters $\xi_{\ell,a|b,c}$ for symbol a at position ℓ of this PWM amount to $\alpha_{\ell,a|b,c} = \frac{1}{|\Sigma_\ell|}\alpha_{\pi_d|c}$, where Σ_ℓ denotes the *local* alphabet, which may be reduced due to a split at position ℓ in the predecessors of π_d in the decision tree. We use $\alpha_c = 256$ throughout the analyses presented in section 4.4.5. The complete product-Dirichlet prior is then the product of the Dirichlet prior for the class probabilities, the Beta prior for the probabilities of consensus and non-consensus edges, and the product-Dirichlet prior of the parameters of the PWMs at the leaves.

With this choice of hyper-parameters, most of the normalization constants (cf. equation (4.82)) cancel in the same manner as for the likelihood (equation (4.76)) and we obtain a simplified definition of the prior

$$q\left(\boldsymbol{\xi} \,|\, \boldsymbol{\alpha}\right) = \frac{c}{Z(\boldsymbol{\xi})^{\alpha_\cdot}} \exp\left(\sum_{c \in \mathcal{C}} \left(\alpha_c\, \xi_c + \sum_{\pi \in \tau_c} \alpha_{\pi|c}\, \xi_{\pi|c} + \sum_{b \in \tau_c} \sum_{\ell} \sum_{a} \alpha_{\ell,a|b,c}\, \xi_{\ell,a|b,c}\right)\right), \tag{4.84}$$

where c is a constant comprising Γ-terms that only depend on the hyper-parameters, $\pi \in \tau_c$ denotes the nodes in τ_c except the root, and $b \in \tau_c$ denotes the leaves of τ_c. As we prove in appendix A.1, this prior is a log concave function of the parameters $\boldsymbol{\xi}$. Since conditional likelihood for decision tree models in this parameterization is also log concave, the supervised posterior is a log concave function of the parameters as well.

4.4.2.2. Structure learning

The greedy algorithm for learning the structures $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_K), K = |\mathcal{C}|$ of the decision trees is outlined in figure 4.40. We initialize all decision trees in all classes with a single leaf, which corresponds to a single PWM model for the complete data. We then consider in each class $c \in \mathcal{C}$ and at each leaf b in the decision tree τ_c each *admissible* split position ℓ in this leaf. We call a split position admissible, if it has not already been used as a split position by one of the predecessors of b in τ_c. Temporarily, we replace leaf b by an inner node with split position ℓ having two children, one for the consensus nucleotide K and one for the non-consensus nucleotides $\overline{K}$. We denote this temporary structure by $\tilde{\tau}_c$ and replace the original tree τ_c in $\boldsymbol{\tau}$ by $\tilde{\tau}_c$. We then compute the supervised posterior $P(\boldsymbol{c}|\boldsymbol{X}, \tilde{\boldsymbol{\tau}}, \tilde{\boldsymbol{\xi}}^*)q(\tilde{\boldsymbol{\xi}}^*|\boldsymbol{\alpha})$ using the optimal parameters $\tilde{\boldsymbol{\xi}}^*$ for the provisional structure. After testing all admissible splits, we persistently conduct the split that yields the maximum supervised posterior.

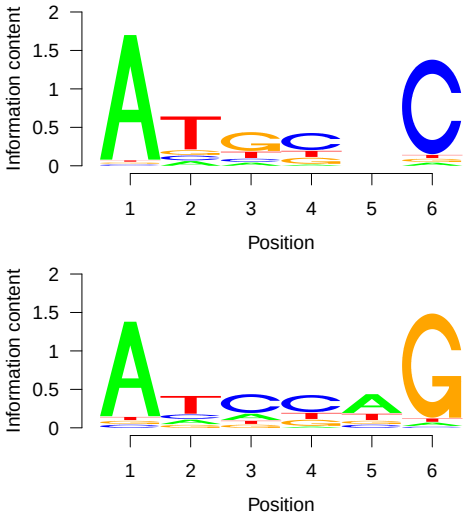
foreach $c \in \mathcal{C}$ **do**
    $\tau_c :=$ PWM
done
$\boldsymbol{\tau} := (\tau_1, \ldots, \tau_K)$
do
    $\boldsymbol{\xi}^*_{old} = \underset{\boldsymbol{\xi}}{\operatorname{argmax}} P(\boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\tau}, \boldsymbol{\xi})\, q(\boldsymbol{\xi} \mid \boldsymbol{\alpha})$
    $\mathrm{SP} := P(\boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\tau}, \boldsymbol{\xi}^*_{old})\, q(\boldsymbol{\xi}^*_{old} \mid \boldsymbol{\alpha})$
    $\boldsymbol{\tau}' := \boldsymbol{\tau}$
    $\mathrm{SP}' := -\infty$
    foreach $c \in \mathcal{C}$ **do**
        foreach leaf b in τ_c **do**
            foreach admissible split position ℓ in b **do**
                $\tilde{\tau}_c := \tau_c$
                split $\tilde{\tau}_c$ in leaf b at position ℓ
                $\tilde{\boldsymbol{\tau}} := (\tau_1, \ldots, \tau_{c-1}, \tilde{\tau}_c, \tau_{c+1}, \ldots, \tau_K)$
                $\tilde{\boldsymbol{\xi}}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmax}} P(\boldsymbol{c} \mid \boldsymbol{X}, \tilde{\boldsymbol{\tau}}, \boldsymbol{\xi})\, q(\boldsymbol{\xi} \mid \boldsymbol{\alpha})$
                $\widetilde{\mathrm{SP}} := P\left(\boldsymbol{c} \,\middle|\, \boldsymbol{X}, \tilde{\boldsymbol{\tau}}, \tilde{\boldsymbol{\xi}}^*\right) q\left(\tilde{\boldsymbol{\xi}}^* \,\middle|\, \boldsymbol{\alpha}\right)$
                if $\widetilde{\mathrm{SP}} > \mathrm{SP}'$ **then**
                    $\mathrm{SP}' := \widetilde{\mathrm{SP}}$
                    $\boldsymbol{\tau}' := \tilde{\boldsymbol{\tau}}$
                fi
            od
        od
    od
    if $\mathrm{SP}' > \mathrm{SP}$ **then**
        $\mathrm{SP} := \mathrm{SP}'$
        $\boldsymbol{\tau} := \boldsymbol{\tau}'$
    fi
while split possible
$\boldsymbol{\xi}^*_{\text{final}} = \underset{\boldsymbol{\xi}}{\operatorname{argmax}} P(\boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\tau}, \boldsymbol{\xi})\, q(\boldsymbol{\xi} \mid \boldsymbol{\alpha})$

Figure 4.40.: Pseudo code of the greedy algorithm for learning the tree structures of MSPD.

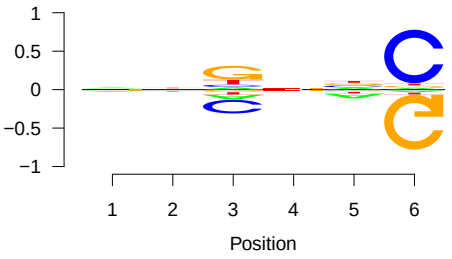
If the maximum supervised posterior SP$'$ reached in the current iteration is larger than the previous SP, we retain the corresponding structures $\boldsymbol{\tau}'$. We repeat the iterations until no admissible split can be found in any of the trees τ_c. Hence, at the end of the algorithm, $\boldsymbol{\tau}$ holds the structures that yield the largest supervised posterior among all structures that are tested by this greedy procedure. We justify the selection of the optimal structure by means of the supervised posterior in section 4.4.5.1. Finally, we obtain the optimal parameters $\boldsymbol{\xi}^*_{\text{final}}$ with respect to the supervised posterior given the chosen structures $\boldsymbol{\tau}$.

4.4.3. Discriminant sequence logos

Sequence logos (Schneider and Stephens, 1990) are a popular visualization of the probability distributions of a PWM model. In a sequence logo, the probabilities of nucleotides are displayed as relative heights of the corresponding letters. The letters at each position are stacked and ordered according to the probabilities, i.e. the most probably occurring nucleotide is displayed topmost, while the least probably occurring nucleotide is placed at the bottom of the stack. The size of the stack is scaled according to the deviation from the uniform distribution. This deviation is measured in terms of the so-called *information content* $2 - H(X_\ell)$, where $H(X_\ell)$ denotes the entropy of the distribution of the random variable X_ℓ emitting the nucleotides in units of bits. Examples of sequence logos are given in figure 4.41(a)



(a) Sequence logos



(b) Discriminant sequence logo

Figure 4.41.: Sequence logos of the probability distributions of two PWM models (a) and the discriminant sequence logo for plotting the first distribution against the second (b).

Here, we propose an alternative sequence logo that aids the viewer in the perception of differences between the distributions represented by two PWM models. Let $\boldsymbol{p}_\ell = (p_1, \ldots, p_{|\Sigma|})$ denote the probability distribution at position ℓ in the first PWM and let $\boldsymbol{q}_\ell = (q_1, \ldots, q_{|\Sigma|})$ denote corresponding probability distribution in the second PWM, where in the case of DNA sequences $\Sigma = \{A, C, G, T\}$ and $|\Sigma| = 4$. We aim at a joint representation of the $\boldsymbol{p}_\ell$ and $\boldsymbol{q}_\ell$ that highlights positions ℓ with relevant differences between $\boldsymbol{p}_\ell$ and $\boldsymbol{q}_\ell$. To this end, we

plot the stacked letters representing the $\boldsymbol{p}_\ell$ on the positive scale and those representing the $\boldsymbol{q}_\ell$ on the negative scale. In analogy to the original sequence logo, we order the stacked letters according to the probabilities such that the nucleotides with the highest probability are placed at the very top and the very bottom, respectively, whereas the nucleotides occurring with the lowest probability are located at the axis. We scale the stacked letters by the *Jensen-Shannon divergence* (Lin, 2002), which is defined as

$$D_{\mathrm{JS}}(\boldsymbol{p}_\ell, \boldsymbol{q}_\ell) = \frac{1}{2}\left[D_{\mathrm{KL}}(\boldsymbol{p}_\ell, \boldsymbol{m}_\ell) + D_{\mathrm{KL}}(\boldsymbol{q}_\ell, \boldsymbol{m}_\ell)\right], \tag{4.85}$$

where

$$\boldsymbol{m}_\ell = \frac{1}{2}\left[\boldsymbol{p}_\ell + \boldsymbol{q}_\ell\right] \tag{4.86}$$

and D_{KL} denotes the Kullback-Leibler divergence (Kullback and Leibler, 1951). In contrast to the Kullback-Leibler divergence, the Jensen-Shannon divergence is limited to $[0, 1]$ (Lin, 2002) and, hence, better suited as a scaling factor of discriminant sequence logos. Jensen-Shannon divergence has also been successfully applied to the segmentation of DNA sequences (Grosse et al., 2002).

An example of a discriminant sequence logo is given in figure 4.41(b). This discriminant sequence logo is an alternative representation of the two sequence logos presented in figure 4.41(a). The example illustrates the utility of discriminant sequence logos for locating differences between PWM models. In the two sequence logos of figure 4.41(a) the most important positions appear to be position 1 and 6, whereas the remaining positions do not exhibit a major deviation from the uniform distribution. With stress on the differences, the discriminant sequence logo reveals that the largest deviations between the two PWM models can be found at positions 3 and 6. While the deviation at position 6 (C vs. G) is easily observed from the sequence logos as well, the deviation at position 3 is more subtle and can likely be overlooked. Hence, we use discriminant sequence logos in the following, whenever we are interested in the differences between distributions.

4.4.4. Donor splice sites

We use two sources of data to evaluate the MSPD algorithm. First, we use the data set of (Yeo and Burge, 2004), which contains $12,623$ human canonical donor splice sites and $269,155$ decoy sites, i.e. sequences that exhibit the canonical GT at positions +1 and +2 but are no functional donor splice sites. This data set is already partitioned into training and test data sets, and we adopt this partitioning. Second, we use data sets compiled in (Sonnenburg et al., 2007) for five species, namely *Arabidopsis thaliana* (thale cress), *Caenorhabditis elegans* (nematode), *Drosophila melanogaster* (fruit fly), *Danio rerio* (zebra fish), and *Homo sapiens* (human). Besides canonical donor splice sites, these data also contain non-canonical sites with GC at positions +1 and +2, respectively. The sizes of the data sets are listed in table 4.5. Sonnenburg et al. (2007) also suggest partitionings into five parts for each of the five data sets, which we use in a 5-fold cross validation.

Table 4.5.: Number of donor and decoy sites in the six data sets.

Data set	Yeo & Burge	*A. thaliana*	*C. elegans*	*D. melanogaster*	*D. rerio*	*H. sapiens*
donor sites	12,623	76,659	64,844	29,788	143,495	160,601
decoy sites	269,155	3,311,934	2,846,598	4,126,777	33,175,785	76,335,126

Sonnenburg et al. (2007) originally provide sequences of length 398 bp, which they cut down to length 141 bp for the analysis. However, most of the algorithms considered in this dissertation do not work on sequences of such length for reasons of computation time: Estimating the parameters of the MEM model is limited to shorter sequences due to the computation of the partition function, which sums over all $|\Sigma|^L$ possible sequences of length L. Finding the optimal permutation of a PVLMM is an NP-hard problem and, hence, limited to short sequences as well. Finally, the greedy algorithm for learning the decision tree structures of MSPD tests all admissible split positions in each iteration. Hence, we cut the sequences to a length of 9 bp where 3 positions are located at the end of the exon, which is the same choice of positions as for the data set of (Yeo and Burge, 2004), and covers the positions bound by U1 and U6 during splicing (see section 2.3).

4.4.5. Results & Discussion

In this section, we evaluate MSPD on the six data sets introduced in the previous section. We first investigate if the supervised posterior is a suitable measure for selecting tree structures. We then compare the classification performance of MSPD to five other algorithms for donor splice site prediction. Finally, we scrutinize the tree structures learned for features that are specific for donor splice sites, and we use MSPD as an exploratory method for finding differences between donor splice sites of different organisms.

4.4.5.1. Supervised posterior for structure selection

For investigating if the supervised posterior is suited for selecting tree structures, we start the MSPD algorithm for learning the structures on the first training data set for *D. melanogaster*. In each iteration, i.e. after conducting a persistent split, we evaluate the performance of the resulting classifier on the corresponding test data set. We repeat this procedure for all five training data sets and corresponding test data sets for *D. melanogaster*.

Figure 4.42 shows the plot of the performance measures AUC-ROC and AUC-PR (see section 3.5.1) against the supervised posterior (SP). The lines start with a combination of two PWMs in the lower left corner and continue to the largest trees in the upper right corner. Besides one major dip we find a good correlation of AUC-ROC and SP for the first partitioning as well as for the averaged results. The Pearson correlation coefficient between the values of AUC-ROC and SP amounts to 0.890 on the first partitioning and 0.854 on the averaged results. We observe an even stronger correlation between AUC-PR and SP, where we find correlation coefficients of 0.983 and 0.96, respectively. We compare these values to the correlations between AUC-ROC and AUC-PR, which may serve as a base line of the range of correlations that can be expected in this scenario. These correlation coefficients are 0.921 and

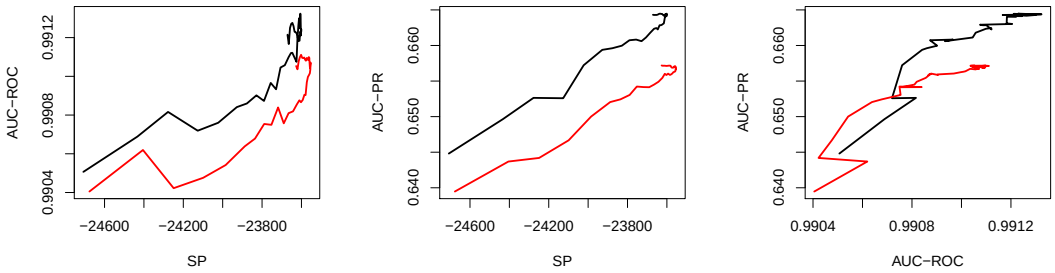


Figure 4.42.: Relation between the value of the supervised posterior on the training data and classification performance on the test data for *D. melanogaster*. The left figure shows a plot of the performance as measured by AUC-ROC against SP for the first partitioning (black line) and averaged over all five partitionings (red line). The figure in the middle illustrates the same analysis for AUC-PR vs. SP. As an indication of the relevance of the correlation we repeat the analysis for AUC-PR vs. AUC-ROC in the right figure.

0.897, respectively, indicating a relevant correlation between the supervised posterior and the two performance measures. Although the best trees with respect to AUC-ROC or AUC-PR are not exactly equal to those yielding the largest supervised posterior, we judge the supervised posterior a reasonable measure for the selection of tree structures.

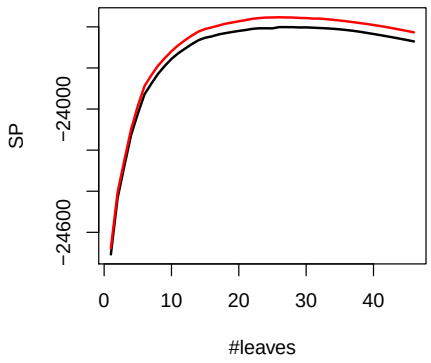


Figure 4.43.: Plot of SP against the number of leaves for the first partitioning (black line) and averaged over all five partitionings (red line) for *D. melanogaster*.

We plot SP against the total number of leaves across the trees of both classes in figure 4.43. We find that most of the improvement in SP is gained approximately during the first 10 splits. Trees with more than 15 leaves yield only slightly increasing SPs.

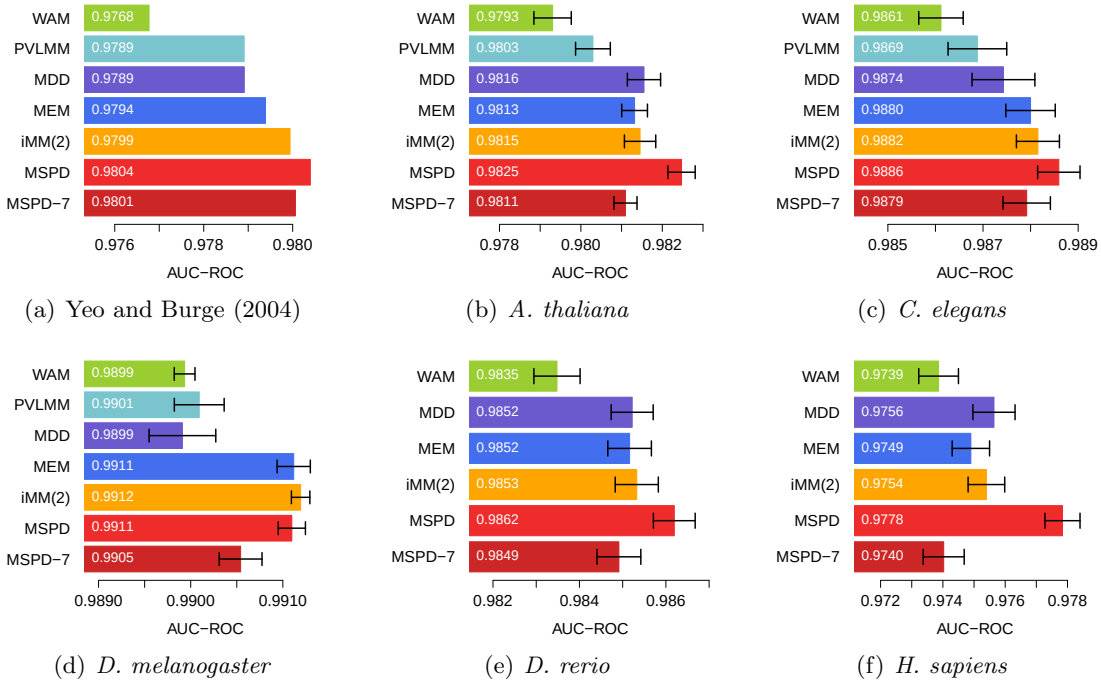
Because the optimal trees are too large for a reasonable interpretation, we restrict the total number of leaves across the trees of both classes to 7 in section 4.4.5.3. To obtain an assessment of the performance lost due to this restriction, we include these classifiers into the comparison presented in the next section. We also observe that the curve of SP against the number of leaves is fairly smooth. Hence, we stop the iterations of structure learning, if SP does not increase for more than five iterations in order to save computation time.

4.4.5.2. Comparison of classification performance

We compare the classification performance of MSPD to five other popular approaches for the prediction of donor splice sites. The classifiers considered are (i) a combination of a WAM model (Zhang and Marr, 1993) for the donor splice sites and a WAM model for the decoy sites (abbreviated by WAM), (ii) a combination of a PVLMM of initial order 3 and a variable length Markov model (VLMM) of initial order 2 as proposed by Zhao et al. (2005) (PVLMM), (iii) a combination of two MDD decision trees (Burge, 1998) using identical significance levels (MDD), (iv) a combination of two MEM models using all two-point dependencies, which was the best model in (Yeo and Burge, 2004) (MEM), and (v) a combination of two inhomogeneous Markov models of order 2 learned by the discriminative MSP principle (iMM(2)). We use an equivalent sample size of 256 in both classes for MSPD and iMM(2) throughout the analyses. We also test the combination of an MDD decision tree for modelling donor splice sites and a PWM model for the decoy sites, which achieves a consistently lower classification performance than the combination of two MDD decision trees and is hence omitted in the final comparison. In addition to iMM(2), we also evaluate the performance of MSP-trained Markov models of orders 1, 3, and 4 and find that iMM(1) performs considerably worse than iMM(2), while the results iMM(3) and iMM(4) differ only slightly in both directions from the results of iMM(2). Keilwagen et al. (2007) originally propose to learn these inhomogeneous Markov models by the non-Bayesian MCL principle, which also results in a reduced classification performance. For these reasons, we include only the MSP-trained iMM(2) into the comparison presented in the following. For the largest two data sets, namely those of *D. rerio* and *H. sapiens*, we could not conduct experiments for PVLMM, because the current version of the program is unable to handle data sets of this size.

We use AUC-ROC and AUC-PR as measures of classification performance, because AUC-ROC is a common general measure of classification performance, whereas AUC-PR is better suited in cases of unbalanced class abundances (Davis and Goadrich, 2006) (cf. table 4.5). These two measures are also selected in (Sonnenburg et al., 2007). Additionally, we include FPR for a fixed Sn of 95%, i.e. the point on the ROC curve that measures the rate of erroneously classified decoy sites if we correctly recover 95% of the true donor splice sites. For the data set of (Yeo and Burge, 2004) we perform a single training on the dedicated training data set and test the resulting classifiers on the independent test data set. For the remaining data sets, we adopt the partitioning proposed by Sonnenburg et al. (2007) in a 5-fold cross validation (see section 3.5.2).

Figure 4.44 presents the classification performance of the five approaches and MSPD on the data sets of (Yeo and Burge, 2004) and the five organisms from (Sonnenburg et al., 2007) considering AUC-ROC. We additionally include MSPD with exactly 7 leaves across all decision trees into the analysis, since we examine trees of this size in the next section, where we search for specific patterns of donor splice sites. We conduct a 5-fold cross validation for the five data sets of (Sonnenburg et al., 2007) and, hence, obtain standard errors together with the values of AUC-ROC. We can use these standard errors to assess the significance of the differences in classification performance. We consider a difference significant, if it exceeds two-fold the standard error. In the barplots of figure 4.44 et seqq. the two-fold standard errors are indicated as error bars.



(a) Yeo and Burge (2004) (b) *A. thaliana* (c) *C. elegans*

(d) *D. melanogaster* (e) *D. rerio* (f) *H. sapiens*

Figure 4.44.: AUC-ROC achieved by the combination of two WAM models (WAM), a PVLMM of order 3 and a VLMM of order 2 (PVLMM), two MDD decision trees (MDD), two MEM models (MEM), two inhomogeneous Markov models trained by MCL (iMM(2)), two MSPD decision trees (MSPD), and MSPD limited to exactly 7 leaves (MSPD-7). The error bars in figure b) through f) indicate two-fold the standard error observed in the cross validation experiment.

For five out of the six data sets, MSPD yields the largest AUC-ROC of all considered approaches. The only exception is the data set of *D. melanogaster* for which we observe a comparable AUC-ROC for MEM, iMM(2), and MSPD. MSPD yields an AUC-ROC of 0.9804 compared to 0.9794 for MEM and 0.9799 for iMM(2) on the data set of (Yeo and Burge, 2004), 0.9825 compared to 0.9813 and 0.9815, respectively, for *A. thaliana*, 0.9886 compared to 0.9880 and 0.9882 for *C. elegans*, 0.9911 compared to 0.9911 and 0.9912 for *D. melanogaster*, 0.9862 compared to 0.9852 for MEM and 0.9853 for iMM(2) on the *D. rerio* data set, and an AUC-ROC of 0.9778 for *H. sapiens*, where MEM yields 0.9749 and iMM(2) achieves an AUC-ROC of 0.9754. The improvement of MSPD over the other approaches is significant for *A. thaliana*, *D. rerio*, and *H. sapiens*, whereas it is not significant for *C. elegans* compared to iMM(2).

MEM significantly outperforms MDD only for *D. melanogaster* and performs even worse in case of *H. sapiens*. Similarly, the improvement of iMM(2) over MEM with respect to AUC-ROC is significant for none of the five data sets. Against this background, we might reason that the improvement gained by the combination of the decision tree model and discriminative learning by MSP is relevant compared to earlier improvements.

PVLMM achieves an AUC-ROC that is comparable to that of MDD on the tested data sets with exception of the *A. thaliana* data set, where it performs significantly worse. MSPD limited to exactly 7 leaves across both decision trees performs significantly worse than MEM

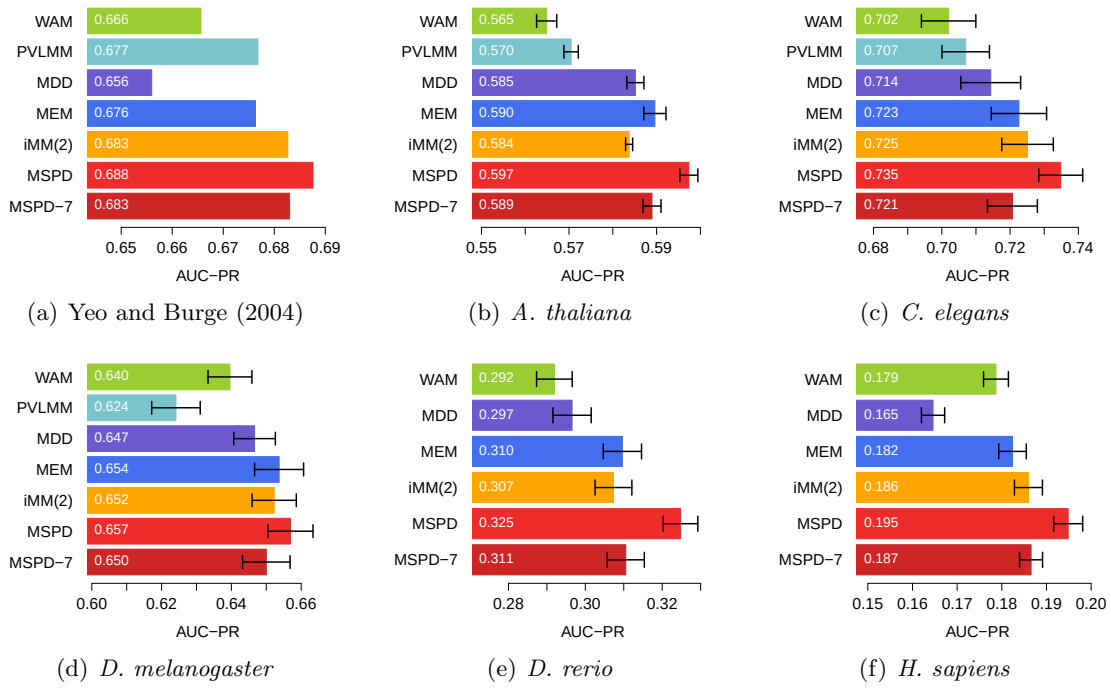
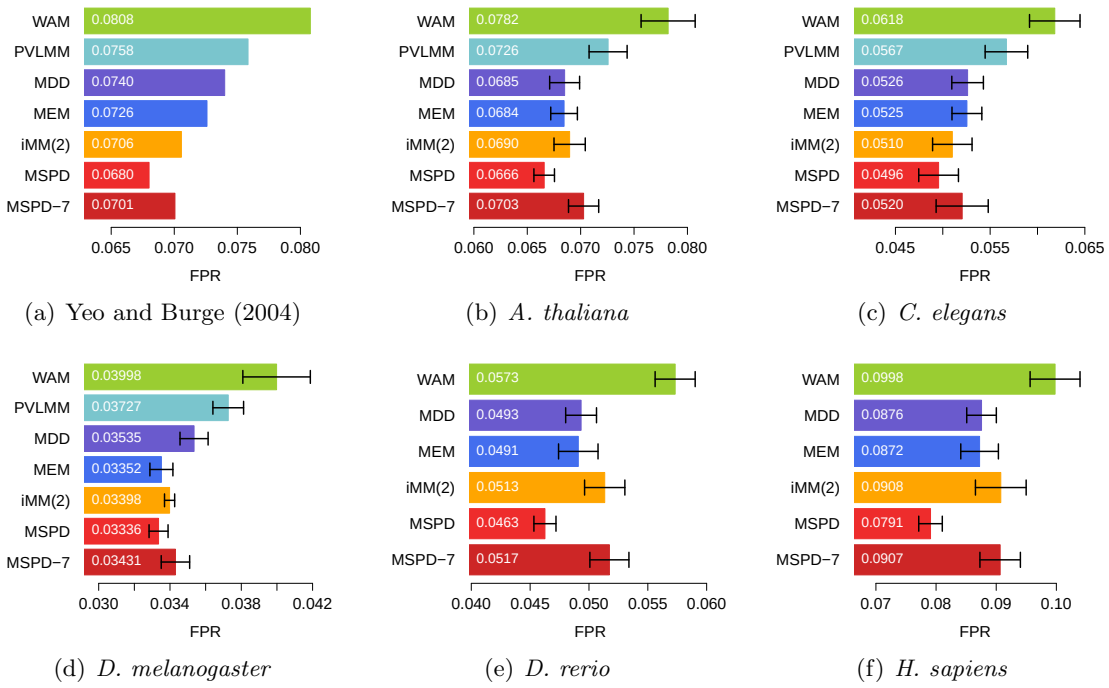


Figure 4.45.: AUC-PR achieved by the seven considered classifiers. The error bars in figure b) through f) indicate two-fold the standard error observed in the cross validation experiment.

only in two of the five data sets of (Sonnenburg et al., 2007), while it yields an even improved AUC-ROC for the data set of (Yeo and Burge, 2004). It therefore appears justifiable to limit the examination of the learned decision trees to trees of this size in the next section.

Turning to AUC-PR as measure of performance in figure 4.45, we find a similar picture. MSPD performs best compared to the other five approaches for all six data sets. Again, the difference between MEM, iMM(2), and MSPD is not significant for *D. melanogaster*, whereas it is significant for *A. thaliana*, *D. rerio*, *H. sapiens*, and *C. elegans*. For the latter data set we could not observe a significant improvement of MSPD over iMM(2) with respect to AUC-ROC. We compare the values of AUC-PR gained by MSPD to the best of the previous approaches for the six data sets and find a AUC-PR of 0.688 compared to 0.683 for iMM(2) on the data set of (Yeo and Burge, 2004), 0.597 compared to 0.590 for MEM on the *A. thaliana* data set, 0.735 compared to 0.725 for iMM(2) on the *C. elegans* data set, 0.657 compared to 0.654 (MEM) for *D. melanogaster*, 0.325 compared to 0.310 for MEM on the *D. rerio* data set, and an AUC-PR of 0.195 compared to 0.186 for iMM(2) on the data set of *H. sapiens*.

Using AUC-PR, the difference between MDD and MEM becomes more articulate than it was the case for AUC-ROC. MEM yields a significantly larger AUC-PR for three of the five data sets, and the improvement is almost significant for the remaining two data sets. Notably, the combination of two WAM models, published in 1993, outperforms MDD for two of the six data sets ((Yeo and Burge, 2004) and *H. sapiens*) and yields a larger AUC-PR than PVLMM on the *D. melanogaster* data set. Generally, the performance of PVLMM is fairly unsteady: its AUC-PR is in the same range as that of MEM and considerably above that of MDD on the data set of (Yeo and Burge, 2004), whereas it performs worst of all studied approaches on



(a) Yeo and Burge (2004) (b) *A. thaliana* (c) *C. elegans*

(d) *D. melanogaster* (e) *D. rerio* (f) *H. sapiens*

Figure 4.46.: FPR achieved by the seven considered classifiers. The error bars in figure b) through f) indicate two-fold the standard error observed in the cross validation experiment. In this case, lower values mean superior classification performance.

the *D. melanogaster* data and yields an AUC-PR between those of WAM and MDD for *A. thaliana* and *C. elegans*. The improvements of iMM(2) over MEM as compared by AUC-PR is, again, not significant, and iMM(2) performs even significantly worse than MEM for *A. thaliana*. Considering AUC-PR, MSPD-7 performs comparably well as MEM for all six data sets.

Comparing the absolute differences between the different approaches in AUC-PR to those in AUC-ROC supports the theoretical consideration that, for unbalanced data sets, AUC-PR is better suited for a comparison of classifiers than AUC-ROC. For AUC-PR the differences in separability between the data sets stemming from different organisms become more obvious as well. One reason for this observation might be that (Sonnenburg et al., 2007) include sites with a non-canonical GC at positions +1 and +2 into the sets of both, the donor splice site and the decoy sites. However, the ratio of canonical and non-canonical sites in the set of decoys is not controlled to be the same as in the set of donors. Depending on the fraction of non-canonical sites in the considered organism, the presence of a C at position +2 could thus identify potential decoys to a different degree. Another possible explanation is the mere size of the data sets for *D. rerio* and *H. sapiens*. We consider sequences of length 9 over an alphabet of size 4 resulting in only $262,144$ possible sequences. However, the decoy data sets for these organisms contain $33,175,785$ and $76,335,126$ sequences, respectively, rendering a relevant overlap between donor sites and decoy sites more probable than for the other organisms. Finally, the differences between the organisms might also occur due to a different abundance of alternative splicing, which might result in less rigid donor splice sites.

We consider FPR for a fixed Sn of 95% as a third performance measure. Since we aim at keeping the rate of false positives as low as possible, lower values correspond to better classification performance for FPR. In general, the results with respect to FPR are similar to those for AUC-ROC and AUC-PR: MSPD achieves the best (lowest) FPR for all six data sets, where the improvement over the best existing approach is significant for three (*A. thaliana*, *D. rerio*, and *H. sapiens*) of the five data sets that are analyzed in a cross validation experiment, whereas it is not significant in case of *D. melanogaster* and *C. elegans*. MSPD yields an FPR of 0.068 for the data set of (Yeo and Burge, 2004) compared to the next larger FPR of 0.0706 for iMM(2), 0.0666 compared to 0.0684 (MEM) for *A. thaliana*, 0.0496 compared to 0.0510 for iMM(2) on the *C. elegans* data set, 0.0334 compared to 0.0335 (MEM) for *D. melanogaster*, an FPR of 0.0463 compared to 0.0491 for MEM on the *D. rerio* data set, and, finally, an FPR of 0.079 compared to 0.087 achieved by MEM on the data set stemming from *H. sapiens*. Like for AUC-ROC, we observe a significant improvement of MEM over MDD only for *D. melanogaster* considering FPR. Again, iMM(2) does not gain a significant improvement over MEM, and performs even significantly worse on the *D. rerio* data set. Considering FPR, PVLMM consistently achieves an FPR between those of WAM and MDD.

Summarizing the results for the three performance measures, we find that MSPD yields a considerably improved performance compared to existing approaches. The magnitude of improvement on the studied data sets is larger than those gained by MEM (Yeo and Burge, 2004) or iMM(2) (Keilwagen et al., 2007) over the original MDD algorithm (Burge and Karlin, 1997; Burge, 1998). In the next section we aim at elucidating which properties of MSPD might be responsible for its superior classification performance.

4.4.5.3. MSPD decision trees

We start the examination of decision tree structures with exactly 7 leaves learned by MSPD with the *H. sapiens* data set. As a first step, we compare the tree structures obtained for the different partitionings as depicted in figure 4.47 to assess the stability of these structures. The thickness of the borders of the inner nodes indicates the contribution of the corresponding splits to the improvement of SP achieved by the final decision tree compared to a combination of two PWM models, i.e. trees of size one without any split.

We find that the general structure of the decision trees is identical over the five partitionings. The only difference we find is the split position used in the non-consensus branch of split position +5 in the foreground tree. In three out of five cases we observe an additional split on the consensus T at position +2, whereas the algorithm introduces a split on the consensus G at position −1 in the remaining two cases. Considering the thickness of the borders of these alternative nodes, we observe that these splits are responsible for the smallest improvement of SP of all splits in the foreground tree. Against the background of these rather small differences, we restrict the analyses to the tree learned on the first partitioning for all considered data sets. However, we refer to this difference when scrutinizing the foreground tree of the *H. sapiens* data set in the following.

We consider the structure of the decision trees learned by MSPD and examine patterns in the occurrence of nucleotides that are discovered by the splits chosen. To this end, we visualize

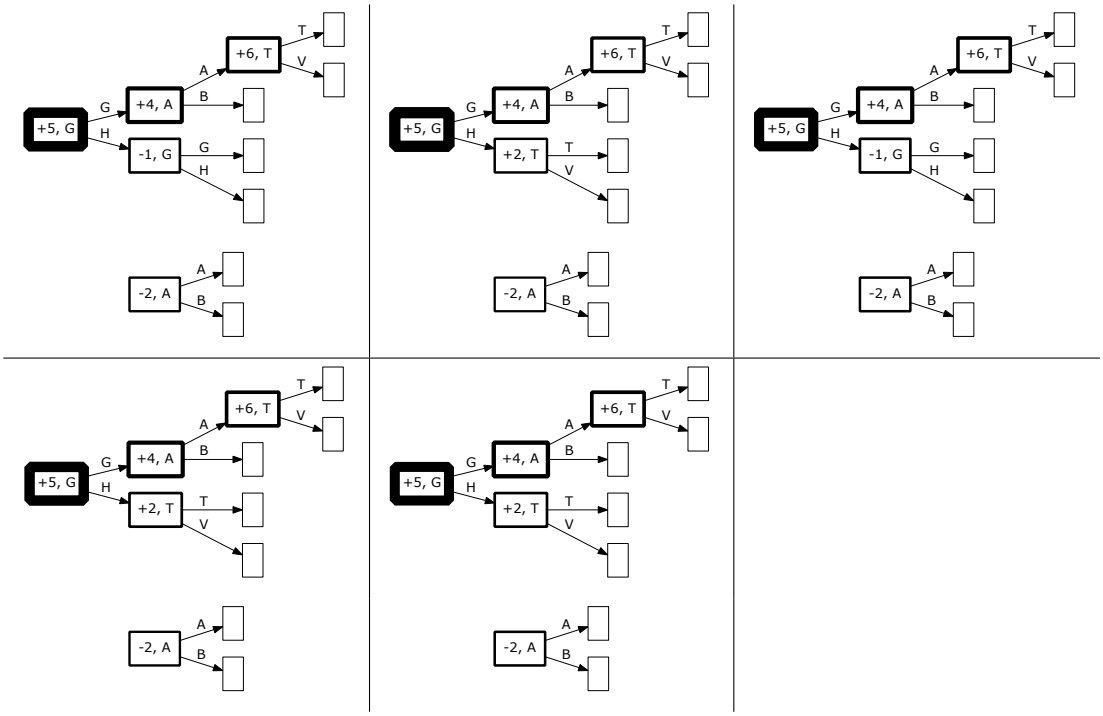
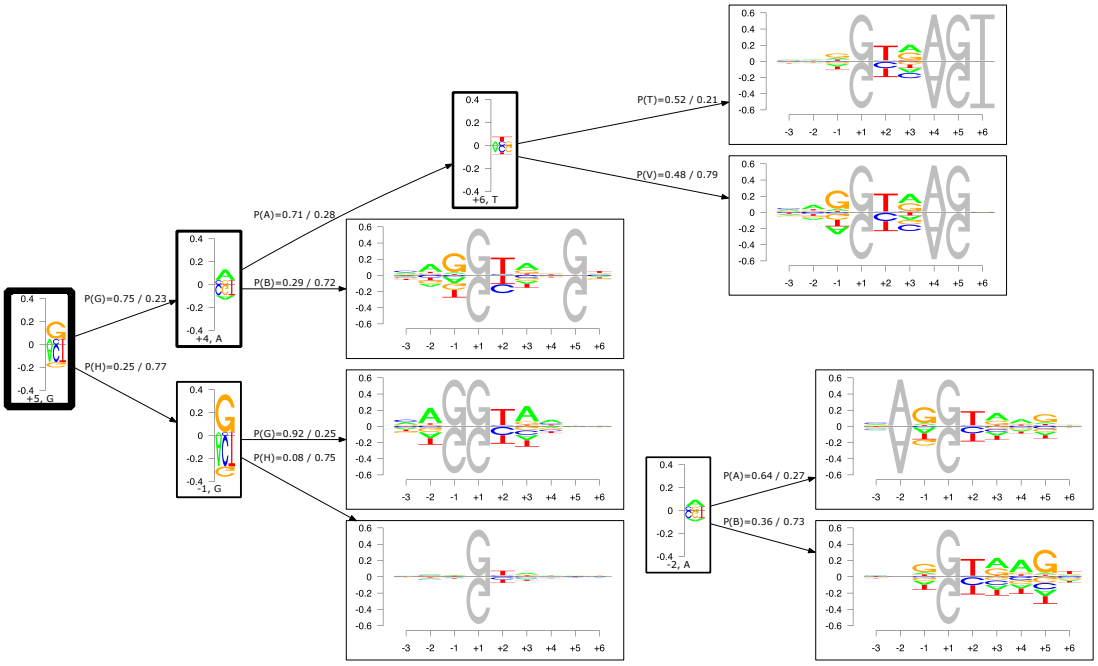


Figure 4.47.: Decision tree structures with 7 leaves across both classes as obtained for the five partitionings of the *H. sapiens* data set. The thickness of the borders of the inner nodes indicates the contribution of the corresponding split to the increase in SP achieved by the full tree compared to two PWM models.

the structures of the decision trees in the foreground and background on the *H. sapiens* data set in figure 4.48. While the tree structures are those obtained by MSPD, we populate both structures with the sequences present in the data and determine the (conditional) relative frequencies of nucleotides in the partitions determined by the decision trees. For each of the trees, we conduct this procedure for the donor splice sites and the decoy sites and use the resulting relative frequencies to plot discriminant sequence logos (see section 4.4.3). We visualize the relative frequencies of consensus and non-consensus nucleotides at the split position by discriminant sequence logos as well, but adapt these to the alphabet of size two, i.e. K and $\overline{K}$, where the non-consensus nucleotides are displayed next to each other with the same height. Since discriminant sequence logos scale nucleotides according to the divergence of the relative frequencies observed in splice sites and decoy sites, this visualization helps to identify specific properties of donor splice sites that are present in the data.

We use the frequencies calculated from the data instead of discriminatively learned parameters, because the discriminatively learned parameters of both trees are influenced by the foreground *and* the background data set. Hence, the contributions of the sequences of the two classes to the parameter values can not be distinguished. However, we can compare the discriminant sequence logos to the corresponding sequence logos for the discriminatively learned parameters and investigate which characteristics of the data lead to the parameters learned.

Considering the root of the foreground tree, we find that a G at position +5 occurs with high frequency in the donor splice sites but with lower frequency in the decoy sites. The importance



(a) Foreground tree (b) Background tree

Figure 4.48.: Foreground (a) and background (b) tree learned on the *H. sapiens* data set. The thickness of the borders around the inner nodes indicates the contribution of the corresponding split to the increase in SP. The one-position discriminant sequence logos depicted in the inner nodes visualize the relative frequencies of the consensus nucleotide and the non-consensus nucleotides in the data at this position. The latter are symbolized by the three non-consensus nucleotides displayed next to each other. The split position and corresponding consensus are given below the logo. The discriminant sequence logos in the leaves represent the corresponding conditional relative frequencies found in the data. Nucleotides that are determined either by the selection of data (G at position +1) or the splits at predecessors in the tree are printed in gray with full height.

of G at position +5 is confirmed by the fact that the split at the root node of the foreground tree gains the largest increase in SP of all splits considered. Turning to the non-consensus branch of the root, we observe a split for a G at position −1, which occurs in this partition of the donor sites with even higher frequency than the G at position +5, whereas the frequencies in the decoy sites remain virtually unchanged. The non-consensus leaf of this inner node is visited by donor sites with almost negligible frequency. These observations lead to two statements: i) at least one of the positions +5 and −1 must be G for a functional donor site and ii) a G at position −1 may compensate for a lack of G at position +5. Considering the consensus leaf of the node "-1, G" we find an increased frequency of the consensus A at position +3 and at position −2 compared to the remaining leaves. This also indicates that a compensation of a non-G nucleotide at position +5 is possible by a strong binding to positions −1 and −2 on the exon side and position +3 on the intron side. These findings are in accordance with those of (Burge and Karlin, 1997), who find a general compensatory effect between the exon and the intron side of donor splice sites, a strong compensatory relation between positions +5 and −1, and a positive dependency of position −2 on position −1. The strong demand for a

G at position −1 and +5 is also found by Carmel et al. (2004), who investigate dependency structures within donor splice sites based on comparative studies between human and mouse. Carmel et al. (2004) also note the general dependency between exon and intron side and a possible compensatory effect between positions −2 and +5. In contrast to previous findings, we observe that position +4 is of minor relevance if the G at position +5 is missing. We even find that all positions downstream of +3 are of minor relevance if the G at position +5 is absent as represented by the two leaves in the non-consensus branch of the root.

The compensatory effects between position −1 and −2, and +5 and +6 can possibly be attributed to the simultaneous binding of U6 to these positions at the beginning of the splicing process (cf. section 2.3).

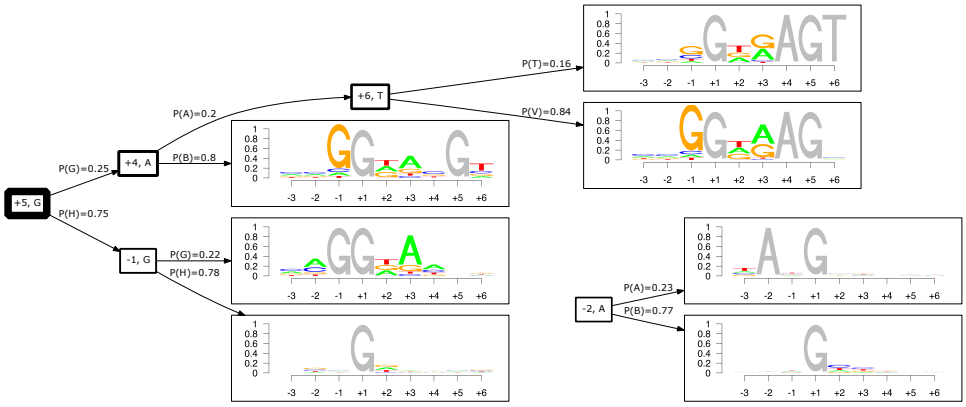
Considering the consensus branch of the root, i.e. the sequences with a G at position +5, we find two further splits for positions +4 and +6. In those sequences that exhibit the consensus at positions +4 through +6 represented in the leaf in the upper right, we observe a reduced relevance of the exon side indicating that a strong binding of U6 on the intron side is sufficient for a functional donor splice site. This is again in accordance with the findings of (Burge and Karlin, 1997) and (Carmel et al., 2004), who both find similar dependencies between positions +4, +6, and +5. Notably, −1 becomes relevant as soon as either +6 or +4 is not equal to the consensus as can be observed from the discriminant sequence logos in the non-consensus leaves of nodes "+4, A" and "+6, T". In case of lacking consensus at position +4 we also find that position +3 is less relevant than in the two leaves in the consensus branch under node "+4, A".

A compensatory effect between positions −2 and +5 is also detectable in the background tree. Here, the root node splits for an A at position −2 on the exon side. On the one hand, we observe an increased relevance of position −1 in the consensus case and a reduced divergence between donor and decoy sites on the intron side. On the other hand, a lack of A at position −2 is compensated by a better accordance to the consensus on the intron side.

Surprisingly, the differences between canonical donor splice sites, which have a T at positions +2, and non-canonical donor splice sites, which have a C this position, seem to be of minor relevance, since these are neither used for a split nor do we observe widespread differences at position +2 in the discriminant sequence logos at different leaves. One difference we do observe is a relatively high abundance of non-canonical splice sites in the non-consensus leaf of "-1, G" as indicated by the low divergence between donor and decoy sites at position +2 compared to the consensus leaf. Stated differently, given a non-consensus nucleotide at position +5 we observe a positive correlation between the canonical T at position +2 and a G at position −1. This might be one reason why we also found a split at position +2 instead of −1 in the trees depicted in figure 4.47. However, this could also be a random effect due to the low frequency of observing G neither at position +5 nor at position −1.

In the following, we want to evaluate how the findings on the data – guided by the discriminatively learned structure of the decision trees – are also represented by the discriminatively learned parameters. In figure 4.49, we depict the foreground and background tree learned on the *H. sapiens* data set. In contrast to figure 4.48, we do not plot discriminant sequence logos, but sequence logos that illustrate the discriminatively learned parameters. A first observation

is a probability close to $\frac{1}{4}$ for all consensus edges and, as a consequence, a probability close to $\frac{3}{4}$ for all non-consensus edges. The closer a split is to the root the lower is the deviation from this distribution. Most likely, this is an effect of the choice of hyper-parameters (see section 4.4.2.1), which penalize a deviation from the $\frac{1}{4}$ / $\frac{3}{4}$ pattern close to the root stronger than in he depth of the tree. Additionally, all parameters in the foreground and background tree are optimized conjointly and, hence, these parameters may also be influenced e.g. by the a-priori probabilities of the classes. Unfortunately, this also complicates the interpretation of the consensus and non-consensus probabilities learned by MSPD.



(a) Foreground tree (b) Background tree

Figure 4.49.: Foreground (a) and background (b) tree learned on the *H. sapiens* data set. In contrast to figure 4.48, the sequence logos in the leaves visualize the parameters of the models learned by MSPD. Again, nucleotides determined by the selection of data or splits at predecessors in the tree are printed in gray.

In the foreground tree, the parameters in the two leaves below "-1,G" in the non-consensus branch of the root support our previous findings. Especially the importance of position +3 if we observe G only at position −1 but not at position +5 is emphasized by the discriminatively learned parameters. In contrast, position −2 appears to be relevant, but less than could be expected from the frequencies visualized in figure 4.48. Turning to the consensus branch of the root, we find again that the exon side looses importance in case of the consensus at positions +4 through +6. The compensatory effect of position −1 if either position +4 or position +6 lacks the consensus, however, becomes even more evident from the discriminatively learned probabilities. The same holds true for position +3, which is more relevant if an A is present at position +4 than in the non-consensus case.

Considering the background tree, we find the compensatory effect between position −2 and the intron side only slightly supported. In the consensus branch of the root, we find deviations from the uniform distribution for positions −3 and −1, whereas for the non-consensus branch, we find similar deviations for positions +3 and +4. However, the parameters do not reflect the preference for G at position +5 if position −2 is not equal to the consensus A. One explanation might be that this relation is already sufficiently modelled in the foreground tree.

With regard to the differences between canonical and non-canonical donor splice sites, we find

that, in the foreground tree, the parameters at position +2 assign almost equal probabilities to A, G, and T, and a probability close to 0 to C. Stated differently, the discriminative MSP principle preferentially learn that position +2 must not be C in the foreground class. A slight deviation from this observation can be observed only for the non-consensus leaf under "-1, G", which has been discussed earlier. Such effects, where the model of the foreground class essentially represents an inverted property of the sequences in the background class, may also interfere with learning the probabilities of consensus and non-consensus at the split positions.

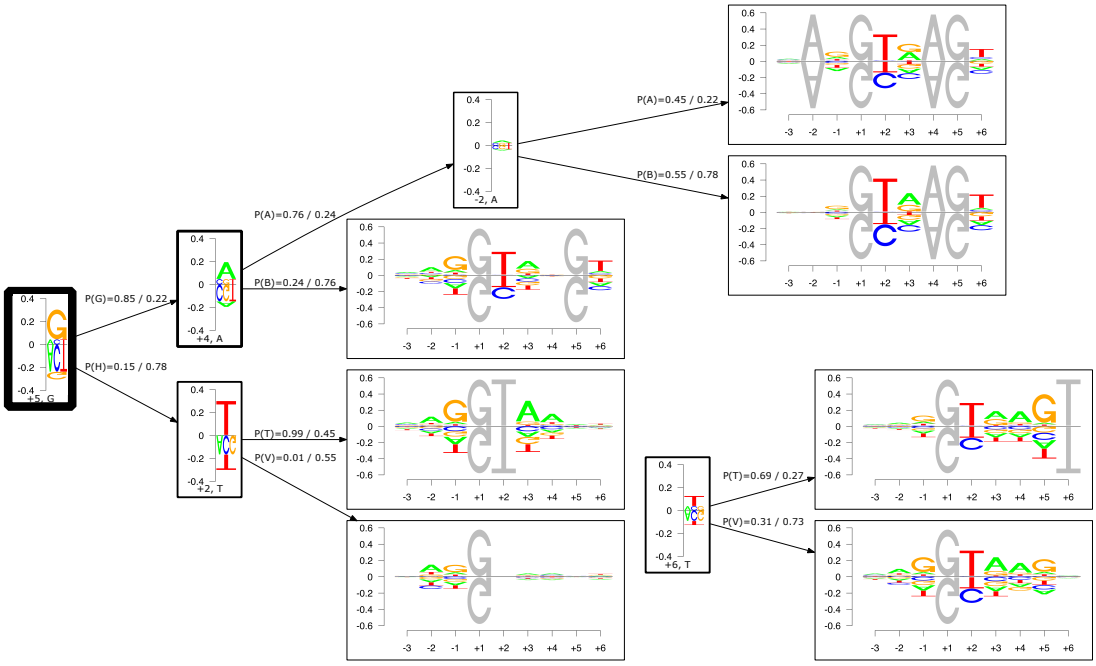
We conclude from these findings that most of the observation from figure 4.48 are supported by the discriminatively learned parameters of figure 4.49. However, besides their discriminative power, the parameters learned by MSPD are less suited for exploring specific properties of donor splice sites than the discriminative structure combined with discriminant sequence logos reflecting the differences between donor and decoy sites present in the data. Hence, we only use the latter approach in the following.

We verify our findings on the decision trees learned from the training data sets of (Yeo and Burge, 2004) (data not shown). Besides the statements regarding non-canonical donor splice sites, which are not included in these data, we find all properties stated for the *H. sapiens* data set of (Sonnenburg et al., 2007) confirmed.

Most of the properties of donor splice sites that we derive from the *H. sapiens* trees are also valid for the other organisms. All exhibit a strong preference for G at position +5 and for all organisms this split is located at the root of the foreground tree with the greatest contribution to the final SP. We can also confirm a general compensatory effect between the exon and the intron site of the donor splice site, which is especially pronounced between positions +5 and −1. We find consistently among the different organisms that a strong intron side renders the exon side less relevant, although position −1 appears to be of greater general importance than −2 and −3. We also find confirmed that a lack of +5 is a "blocker" on the intron side, i.e. the relevance of positions +4 and +6 is greatly decreased in this case. The differentiation between canonical and non-canonical donor splice sites remains of minor relevance across the studied organisms.

As a specific example we consider the decision tree structures learned by MSPD on the *D. melanogaster* data set, which are depicted in figure 4.50. Comparing the foreground and background tree to those learned on the *H. sapiens* data set, we find similar structures and a large overlap between the split positions for these two data sets. However, the split on position −1 in the non-consensus branch of the root of the foreground tree is replaced by a split on position +2, which also appeared in some of the trees for *H. sapiens* (see figure 4.47). Additionally, position −2 in the background tree and position +6 in the foreground tree are exchanged.

We also find notable differences. On the one hand, we observe in the root node of the foreground tree that the consensus G at position +5 is more conserved in *D. melanogaster* than in *H. sapiens*. On the other hand, we observe a C at position +2 of the decoy sites with a higher frequency than T in all leaves of both trees, except for the non-consensus branch of the root of the foreground tree. The high abundance of C at position +2 of the decoy sites might



(a) Foreground tree (b) Background tree

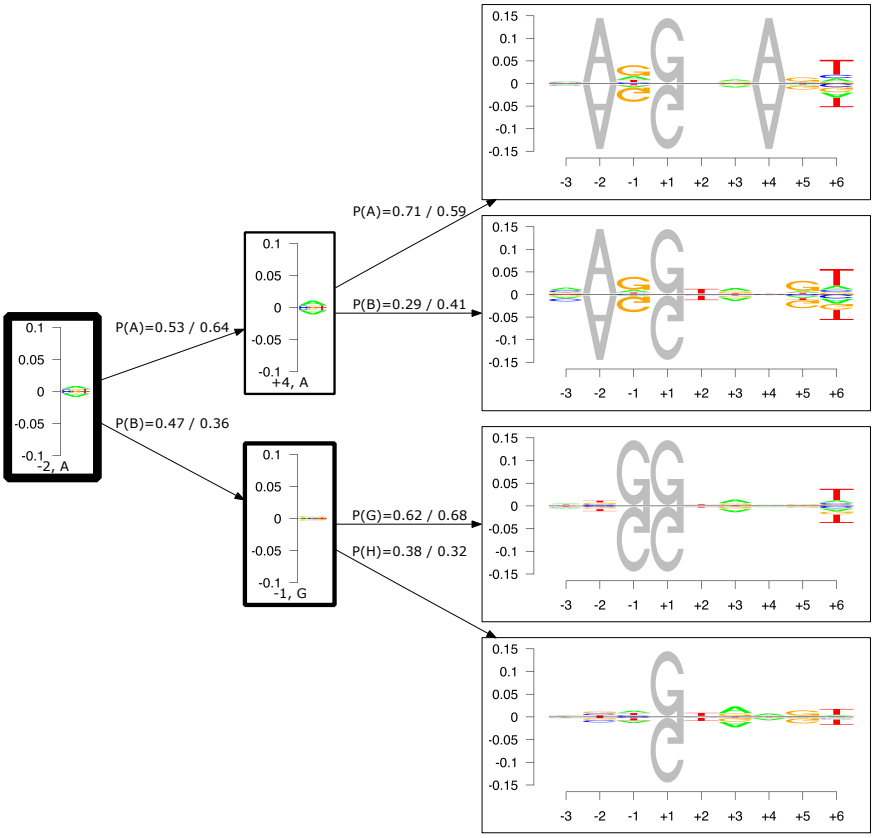
Figure 4.50.: Foreground (a) and background (b) tree learned on the *D. melanogaster* data set. For a description of the visualization see figure 4.48.

be one of the reasons of the good performance of all approaches on this data set compared to the other organisms as presented in the previous section.

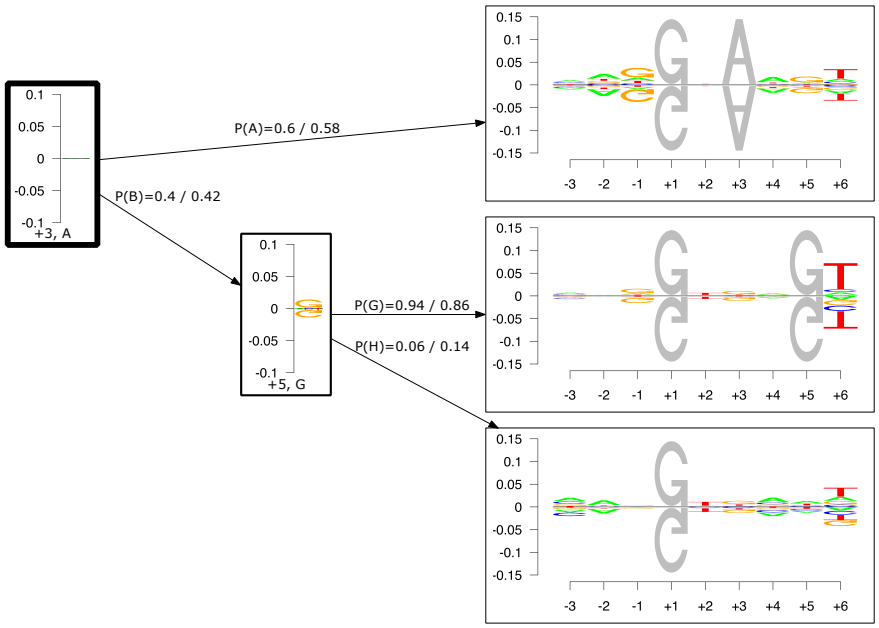
We also find some additional properties we did not observe for the *H. sapiens* data set. Considering the background tree, we find a strong positive correlation between a T at position +6 and a G at position +5. This is in accordance with the findings of Carmel et al. (2004) for *H. sapiens*. In the background tree we also find a compensatory effect of position −1 if position +6 does not exhibit the consensus T. The general relevance of positions −1 and +6 in *D. melanogaster* has also been found by Lo et al. (1994). Although both properties are not noticeable in the trees learned for *H. sapiens*, they do not contradict any of the former results.

In the following, we want to use MSPD as an exploratory tool to find differences between the donor splice sites of the studied organisms. To this end, we start the MSPD training using the donor splice sites of one organism as foreground and of another organism as background data set. Since the splicing machinery, especially the snRNA U6 that is one of the snRNAs binding to donor sites, is evolutionary conserved (Brow and Guthrie, 1988), the differences between the donor splice sites of the studied organisms are expected to be less articulate than the differences between donor splice sites and decoy sites. Hence, we upscale the discriminant sequence logos in the illustrations by a factor of 4 in the following.

As a first example, we consider the differences between the donor splice sites of *D. melanogaster* and *H. sapiens*. The structures of the learned decision trees are depicted in figure 4.51. In the



(a) *D. melanogaster*



(b) *H. sapiens*

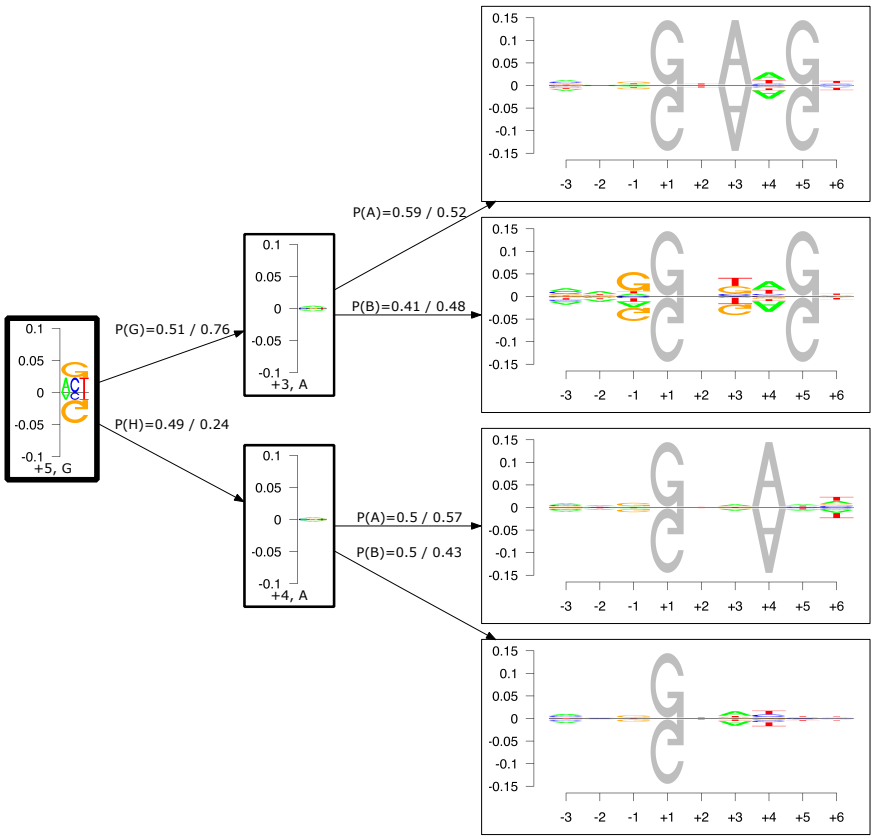
Figure 4.51.: Decision trees learned for the discrimination of splice donor sites of *D. melanogaster* and *H. sapiens*.

two leaves of the consensus branch of the root of the tree for *D. melanogaster*, we find that the consensus G at position −1 occurs more frequently for *H. sapiens* than for *D. melanogaster*. In this branch, in the consensus leaf below "+4,A", we find the consensus G at position −1 with a relative frequency of 0.61 for *D. melanogaster* and a relative frequency of 0.82 for *H. sapiens*. In the non-consensus leaf of "+4,A", we find the consensus with a relative frequency of 0.78 for *D. melanogaster* and 0.94 for *H. sapiens*. For the consensus *T* at position +6, we find the opposite, as the consensus occurs with relative frequencies of 0.65 and 0.67, respectively, for *D. melanogaster* and relative frequencies of 0.41 and 0.41, respectively, for *H. sapiens*. Since these differences are present in the consensus branch of "-2,A", one might speculate that the positive feedback between the adjacent positions −1 and −2 is more relevant for a functional donor splice site in *H. sapiens* than in *D. melanogaster*, whereas dependencies between position −2 on the exon side and position +6 at the intron side are more decisive in *D. melanogaster* than in *H. sapiens*.

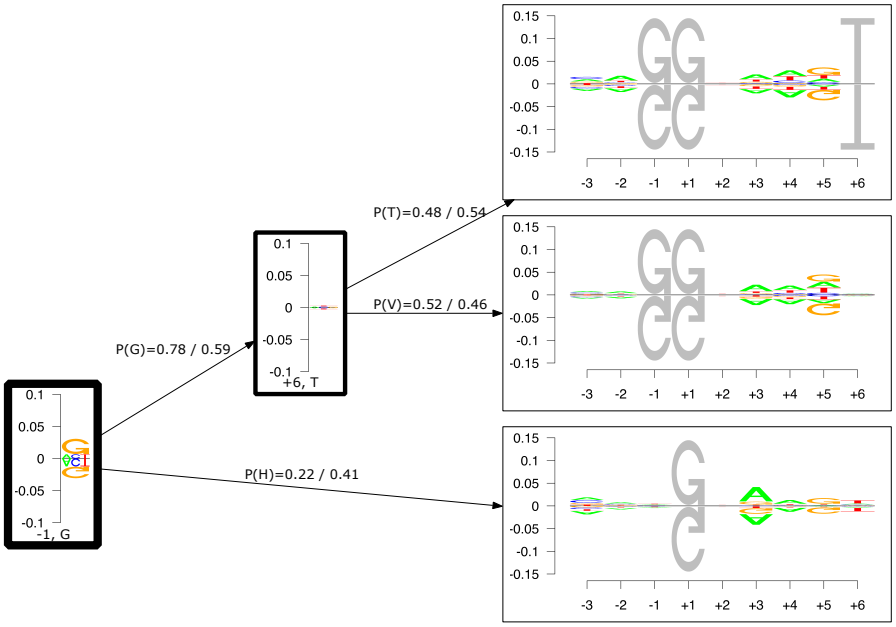
We observe a similar pattern in the tree for *H. sapiens*, where the presence of a G at position +5, represented by the consensus leaf of the inner node "+5,G", is more closely linked to the occurrence of a T at position +6 for *D. melanogaster* than for *H. sapiens*. In the consensus leaf of "+5,G", we find the consensus T at position +6 with a relative frequency of 0.82 for *D. melanogaster* and a relative frequency of 0.55 for *H. sapiens*, while in the non-consensus leaf, we find relative frequencies of 0.49 and 0.28, respectively. Combining these two results, we could also speculate that position +6 is generally more relevant in *D. melanogaster* and that this is also reflected by stronger relations to positions +5 as well as −2.

We examine the differences between the donor splice sites of *A. thaliana* and *C. elegans* as another example. The resulting decision tree structures together with the discriminant sequence logos are presented in figure 4.52. The most prominent difference is already perceivable from the roots of the foreground and background tree. In the root of the foreground tree, we find the consensus G at position +5 more prevalently in the donor splice sites of *C. elegans* than in those of *A. thaliana* resulting in relative frequencies of 0.76 and 0.51, respectively.

In the root of the background tree, we observe a more strongly conserved G at position −1 for *A. thaliana* than for *C. elegans* obtaining relative frequencies of 0.78 and 0.59, respectively. We also discover the latter difference in the foreground tree in the non-consensus leaf of node "+3, A", where we find the consensus G at position −1 with a relative frequency of 0.80 for *A. thaliana* and a relative frequency of 0.56 for *C. elegans*. This indicates that a compensation of a lack of A at position +3 by a G at position −1 occurs more frequently in *A. thaliana* than in *C. elegans*. On the other hand, a compensation by A at position +4 appears with a slight preference in *C. elegans*, as we find it with a relative frequency of 0.74 as opposed to 0.58 for *A. thaliana* in the non-consensus leaf of "+3, A".



(a) *A. thaliana*



(b) *C. elegans*

Figure 4.52.: Decision trees learned for the discrimination of splice donor sites of *A. thaliana* and *C. elegans*.

4.4.6. Conclusions

In this section, we propose maximum supervised posterior decomposition (MSPD), which learns structure and parameters of decision tree models by the discriminative MSP principle. We apply MSPD to the problem of predicting canonical and non-canonical donor splice sites. We find that the resulting classifiers yield an improved classification performance compared to popular models for the prediction of donor splice sites, namely the WAM model, PVLMMs, the original MDD, discriminatively learned iMMs, and the MEM model. However, the classification performance of all approaches studied is considerably inferior to that achieved by Sonnenburg et al. (2007) on sequences of length 141 bp. Hence, it might be worthwhile to enclose MSPD decision trees into a model that can effectively be learned for sequences of this length, e.g. a Markov model learned by the MSP principle.

Scrutinizing some of the decision trees learned, we confirm known properties of donor splice sites like a compensatory effect between exon and intron side of donor splice sites, especially between positions $+5$ and -1, or general positive correlations between positions $+4$, $+6$, and $+5$. However, we also find interesting new properties. For instance, we observe that position $+5$ acts as a "blocker" on the neighboring positions $+4$ and $+6$, i.e. if the consensus at $+5$ is lacking, these position become considerably less relevant. Surprisingly, the differentiation between canonical and non-canonical sites appears to be almost irrelevant in our analyses. Using MSPD as an exploratory tool for discovering variations between different organisms, we find notable differences in the relevance of the consensus Gs at positions $+5$ and -1, and in the strength of compensatory effects between intron and exon side.

4.5. Prediction of microRNA targets

MicroRNAs (miRNAs) are short ($\sim$ 22 nt) non-coding RNAs that bind to partially complementary sites on mRNA target sequences and induce cleavage of the miRNA-mRNA duplex or repress translation of the bound mRNA (Brennecke et al., 2005; Fu et al., 2007; Ghosh et al., 2007). Cleavage or repression is mediated by the RNA-induced silencing complex (RISC) bound to the miRNA. In plants, miRNAs exhibit an almost perfect complementarity to their target site (Rhoades et al., 2002; Reinhart et al., 2002), often bind to protein-coding regions of mRNA, and – due to the high complementarity – often cause degradation of the bound mRNA (Enright et al., 2003). In contrast, miRNAs in animals require a high complementarity only at the 5' end of the miRNA, often termed the *seed* region (Brennecke et al., 2005), and preferentially bind to the 3' untranslated region (UTR) of the mRNA. Due to the imperfect complementarity, animal miRNAs predominantly repress translation instead of inducing degradation of the mRNA (Enright et al., 2003).

4.5.1. Background

As one of the first attempts to predict targets of given miRNAs computationally, Rhoades et al. (2002) consider 16 miRNAs of *Arabidopsis thaliana*. For each of these miRNAs, they search for complementary sites in mRNAs with at most 3 mismatches and no gaps allowed. For 14 of the miRNAs, Rhoades et al. (2002) find potential target sites, which are predominantly located within genes that code for transcription factors involved in development.

Lewis et al. (2003) propose an algorithm for the prediction of targets of vertebrate miRNAs called TargetScan. TargetScan requires a perfect complementarity between positions 2 and 8 at the 5'-end of the miRNA and a potential target. Such potential target sites are elongated up to the first mismatch, but allowing for G:U wobble basepairs. Using RNAfold (Hofacker et al., 1994), the binding between the 3' portion of the miRNA and the 5' region on the mRNA next to the seed region is optimized and the resulting free energy is computed. Predictions are verified using orthologous UTR sequences from other organisms. Lewis et al. (2005) propose a refined version called TargetScanS, which demands a shorter region of the target to be complementary to nucleotides $2-7$ of the miRNA. In turn, it either requires position 8 to match as well – resulting in the original seed region of TargetScan – or an A directly downstream of the seed region, which may bind to the prevalent nucleotide U at position 1 of miRNAs.

In contrast to TargetScan, miRanda (Enright et al., 2003) does not require perfect complementarity at the seed region. Instead, it uses an algorithm similar to Smith-Waterman sequence alignment for the detection of potential target sites with similarity scores of $+5$ for G:C and A:U basepairs, $+2$ for G:U wobble basepairs, and -3 for mismatches. The alignment scores for the first 11 positions of the alignment are weighted by a factor of 2 to account for the importance of the seed region. Additional rules for potential target sites are that no mismatches occur at positions 2 to 4, less than 5 mismatches occur between positions 3 and 12, at least one mismatch is present from position 9 to $L-9$, where L is the length of the alignment of miRNA and target site, and less than 2 mismatches are observed for the last 5 positions.

Potential target sites fulfilling these requirements are additionally filtered for a minimum similarity score of the alignment and for minimum free energy as computed by the Vienna RNA folding package (Wuchty et al., 1998).

Stark et al. (2003) align the first 8 positions of the miRNA to potential target sites using HMMer (Eddy, 1996) with a scoring matrix that allows for G:U mismatches. The resulting hits are elongated such that potential targets are 5 bp longer than the corresponding miRNA and the free energy of the miRNA-mRNA duplex is computed using Mfold (Mathews et al., 1999). After a normalization step that facilitates the comparison of the free energies for different lengths of miRNAs, potential targets are ranked according to the normalized free energy.

PicTar (Krek et al., 2005) searches for perfectly complementary seed regions of 7 nt starting from position 1 or 2 of the miRNA. Mismatches in the seed region are allowed, if these do not increase the free energy of binding. Additionally, Krek et al. (2005) apply a filter with respect to the free energy of the complete miRNA-mRNA duplex, where the applied threshold is more stringent for targets with imperfect basepairing in the seed region. The resulting target sites of miRNAs are then combined in an HMM-like approach to predict coordinated target sites of a fixed set of miRNAs and, consequently, target genes that are putatively regulated by this set of miRNAs.

Rehmsmeier et al. (2004) propose RNAhybrid, which extends RNA secondary structure prediction to two RNA sequences, namely the miRNA and the target site. In contrast to other approaches for RNA secondary structure prediction, RNAhybrid forbids intramolecular interactions. RNAhybrid computes the minimum free energies of putative miRNA-mRNA duplexes, normalizes the computed energies to the length of the considered sequences, and models these by an extreme value distribution to assess the significance of achieved energies. The probability of multiple potential target sites within a common target UTR is modeled by a Poisson distribution to compute p- and E-values. Finally, putative targets are selected with respect to the computed E-values.

Similar to PicTar, DIANA-microT (Maragkakis et al., 2009) prefers perfect complementarity of 7 to 9 nt starting from position 1 or 2 of the miRNA. However, if the considered target site shows good complementarity to the 3' end of the miRNA, the length of this seed region may be reduced to 6 nt and single G:U wobble basepairs are allowed. DIANA-microT uses orthologous UTRs from up to 27 organisms for assessing the conservation of target sites. Finally, the score of a potential UTR target sequence is computed as the average of all potential target sites, which are weighted by the strength of binding and level of conservation relative to a set of "mock" miRNA sequences.

Selbach et al. (2008) measure the effect of miRNA levels on protein concentrations in wet lab experiments and find that single species of miRNA may repress the translation of hundreds of different proteins. However, often the level of down-regulation is fairly mild and Selbach et al. (2008) seldom observe more than 4-fold differences in protein concentration. Selbach et al. (2008) use these results to assess the predictions of target genes of a number of approaches, including TargetScanS, PicTar, miRanda, and DIANA-microT. They find that only for TargetScanS, PicTar, and DIANA-microT more than 50% of the predicted target UTRs are

supported by experimental evidence. In another study, Boross et al. (2009) study the overlap between the predictions of four different sources of predictions, namely the data base miRBase (Griffiths-Jones et al., 2006, 2008), PITA (Kertesz et al., 2007), PicTar, and TargetScan. They observe that less than 1% of the union of these predictions is supported by all four sources, and only 12.4% are predicted by at least two. These two studies reveal that the prediction accuracy of current approaches is still far from perfect. One reason for this shortcoming is that the number of experimentally verified target sites is still very low – for instance, less than 100 experimentally verified targets are reported in TarBase (Papadopoulos et al., 2009).

In contrast to previous approaches, we propose an approach for predicting target sites of given miRNAs that is capable of learning rules of miRNA-target site binding from data sets comprising pairs of target sites and associated miRNAs. This approach employs an extension of profile hidden Markov models (HMMs) (Krogh et al., 1994), which we call *conditional profile HMM* (CoProHMM), and learns the parameters of the CoProHMM by the discriminative MSP principle (see section 3.2.2, p. 13). As a proof of concept, we learn the CoProHMMs on the predictions of existing approaches. Due to the limited number of experimentally verified target sites, we augment a training data set of verified target sites by predictions of existing approaches, and we use the CoProHMM learned from these data for predicting target sites.

4.5.2. Model

CoProHMM consists of three types of states, namely match states, insert states, and delete states. We model the binding of miRNA and mRNA in the match states of the CoProHMM. For each nucleotide in the miRNA, we define one match state, which emits the nucleotide in the mRNA with a probability that depends on the observation in the miRNA. If an mRNA and the associated miRNA are perfectly complementary, we anticipate that only match states are visited for the emission of the complete sequence of the mRNA. Otherwise, delete states allow for skipping nucleotides of the miRNA and, hence, the insertion of gaps into the mRNA sequence. On the other hand, insert states allow for including additional nucleotides into the sequence of the mRNA corresponding to gaps in the miRNA. We expect that insert and delete states are visited, if this improves the alignment of mRNA and miRNA in subsequent match states.

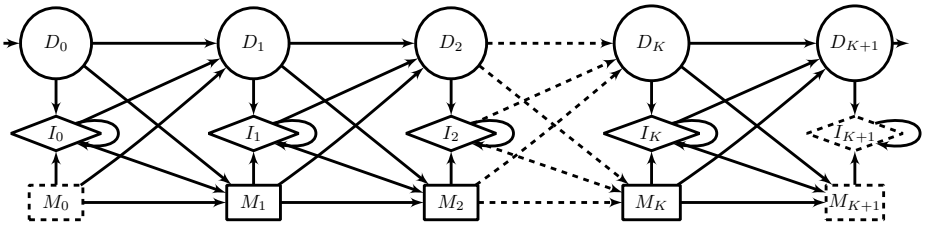


Figure 4.53.: Plan9 architecture of the proposed CoProHMMs. Circles represent silent delete states that do not emit symbols of the target site, diamonds represent insert states that emit symbols of the target site regardless of the sequence of the miRNA, and rectangles represent match states that emit symbols of the target site with probabilities conditional on symbols of the miRNA. Each admissible path starts at D_0 and ends at D_{K+1}. States with dashed borders are not visited in admissible paths.

We use a plan9 architecture (Krogh et al., 1994) for the proposed CoProHMMs, which is illustrated in figure 4.53. We represent silent delete states D_k by circles, insert states I_k by diamonds, and match states M_k by rectangles. Each insert state emits one symbol of a putative target site following a probability distribution that does not depend on the given miRNA but may be different for different insert states I_k. In contrast, the emission probabilities of the match states are defined as conditional probability distributions, where in match state M_k the probability of the symbol observed in the putative target site depends on the symbol observed in the miRNA at position k. In figure 4.53, edges represent transition probabilities that are not fixed to 0. From each node of column k, we can reach node I_k in the same column, and nodes M_{k+1} and D_{k+1} in the next column. Each admissible path in the CoProHMM starts at D_0 and ends at D_{K+1}. Hence, the states M_0, I_{K+1}, and M_{K+1} are never visited in admissible paths, and are only included to simplify recursive definitions in the following. Here, we use $K = 22$, since this is the length of a typical miRNA.

We define the emission probabilities at the insert and match states, and the transition probabilities in terms of real-valued parameters $\boldsymbol{\beta}$ to allow for an unconstrained optimization of the supervised posterior. We define the emission probability $P_{I_k}(a|\boldsymbol{\beta}_{I_k})$ of symbol a in a putative target site for insert state I_k given parameters $\boldsymbol{\beta}_{I_k}$ as

$$P_{I_k}(a|\boldsymbol{\beta}_{I_k}) = \frac{\exp(\beta_{a|I_k})}{\sum_{\tilde{a}\in\Sigma}\exp(\beta_{\tilde{a}|I_k})}, \tag{4.87}$$

where $\boldsymbol{\beta}_{I_k} = (\beta_{A|I_k}, \beta_{C|I_k}, \beta_{G|I_k}, \beta_{U|I_k})$.

In analogy, we define the conditional emission probability $P_{M_k}(a|\boldsymbol{r}, \boldsymbol{\beta}_{M_k})$ of symbol a in a putative target site for match state M_k given the sequence of the miRNA $\boldsymbol{r}$ and parameters $\boldsymbol{\beta}_{M_k}$ as

$$P_{M_k}(a|\boldsymbol{r}, \boldsymbol{\beta}_{M_k}) = P_{M_k}(a|r_k, \boldsymbol{\beta}_{M_k}) \tag{4.88}$$

$$= \frac{\exp(\beta_{a|r_k,M_k})}{\sum_{\tilde{a}\in\Sigma}\exp(\beta_{\tilde{a}|r_k,M_k})}, \tag{4.89}$$

where r_k denotes the k-th symbol of the miRNA $\boldsymbol{r}$ and $\boldsymbol{\beta}_{M_k} = (\beta_{A|A,M_k}, \beta_{C|A,M_k}, \ldots, \beta_{U|U,M_k},)$.

According to the plan9 architecture, we define the transition probability $P_T(V|S_k, \boldsymbol{\beta}_{T,S_k})$ of going to node V if we are currently at node S_k given parameters $\boldsymbol{\beta}_{T,S_k}$ as

$$P_T(V|S_k, \boldsymbol{\beta}_{T,S_k}) = \begin{cases} \frac{\exp(\beta_{V|S_k})}{\sum_{\tilde{V}\in\{I_k, M_{k+1}, D_{k+1}\}}\exp(\beta_{\tilde{V}|S_k})} & \text{if } V \in \{I_k, M_{k+1}, D_{k+1}\} \\ 0 & \text{otherwise} \end{cases}, \tag{4.90}$$

where $\boldsymbol{\beta}_{T,S_k} = (\beta_{I_k|S_k}, \beta_{M_{k+1}|S_k}, \beta_{D_{k+1}|S_k})$.

Generative learning of the parameters of an HMM is typically accomplished by the Baum-Welch algorithm, which is a special case of expectation maximization (EM). One step of this algorithm is the computation of *forward* variables $\mathcal{F}_{S_k}(\ell, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})$, which are defined as the probability of the first l symbols of the sequence $\boldsymbol{x}$ and visiting node S_k in time interval $s(\ell, \boldsymbol{x}|\boldsymbol{r})$, given parameters $\boldsymbol{\beta}$ and, in case of CoProHMMs, the sequence $\boldsymbol{r}$ of the miRNA,

i.e.

$$\mathcal{F}_{S_k}(\ell, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta}) = P\left(x_1, \ldots, x_\ell, S_k \in s(\ell, \boldsymbol{x}|\boldsymbol{r})|\boldsymbol{r}, \boldsymbol{\beta}\right). \tag{4.91}$$

A node S_k is visited in time interval $s(\ell, \boldsymbol{x}|\boldsymbol{r})$ if it is contained in a path from D_0 to D_{K+1}, and the symbols x_1 to x_ℓ have been emitted either by predecessors of S_k in the path or S_k itself, whereas $x_{\ell+1}$ is emitted by a successor of S_k in this path.

We use these forward variables for a recursive definition of the likelihood $P(\boldsymbol{x}|target, \boldsymbol{r}, \boldsymbol{\beta}_{target})$ of sequence $\boldsymbol{x}$ given the class of target sequences *target*, the sequence of the miRNA $\boldsymbol{r}$ and parameters $\boldsymbol{\beta}_{target}$. We define the likelihood as

$$P(\boldsymbol{x}|target, \boldsymbol{r}, \boldsymbol{\beta}_{target}) = \mathcal{F}_{D_{K+1}}(L, \boldsymbol{x}|\boldsymbol{r}\boldsymbol{\beta}_{target}), \tag{4.92}$$

which, following the definition of the forward variables, amounts to

$$P(\boldsymbol{x}|target, \boldsymbol{r}, \boldsymbol{\beta}_{target}) = P\left(x_1, \ldots, x_L, D_{K+1} \in s(L, \boldsymbol{x}|\boldsymbol{r})|\boldsymbol{r}, \boldsymbol{\beta}_{target}\right). \tag{4.93}$$

Using this definition, the likelihood $P(\boldsymbol{x}|target, \boldsymbol{r}, \boldsymbol{\beta}_{target})$ is not necessarily normalized over all possible sequences $\boldsymbol{x} \in \Sigma^L$ of given length L. However, we may extend the sequence of each putative target site by a sentinel symbol #, which is emitted by D_{K+1} with probability 1 and with probability 0 by all other states, and achieve a normalization over all sequences $\boldsymbol{x} \in \Sigma^L\#$. Since the length of putative target sites varies, we consider this laxity of minor influence.

For determining the forward variables of all states recursively, we must assure that the forward variables are evaluated according to the topological order of states. This can be achieved by considering states in the order of the columns of the plan9 architecture and, within column k, by computing the forward variables of delete and match states before computing the forward variable of the insert state. In analogy to original profile HMMs, we recursively define the forward variables of insert state I_k as

$$\begin{aligned}\mathcal{F}_{I_k}(\ell, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta}) = \big[&\mathcal{F}_{I_k}(\ell-1, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})\ P_T(I_k|I_k, \boldsymbol{\beta}_{T,I_k}) + \\ &\mathcal{F}_{D_k}(\ell-1, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})\ P_T(I_k|D_k, \boldsymbol{\beta}_{T,D_k}) + \\ &\mathcal{F}_{M_k}(\ell-1, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})\ P_T(I_k|M_k, \boldsymbol{\beta}_{T,M_k})\big]\ P_{I_k}(x_\ell|\boldsymbol{\beta}_{I_k}).\end{aligned} \tag{4.94}$$

Since symbol x_ℓ is emitted by state I_k, we consider the forward variables $\mathcal{F}_{S_k}(\ell-1, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})$ of all direct predecessors S_k of I_k in the plan9 architecture for position $\ell-1$, which are multiplied by the transition probability from S_k to I_k. The sum of these values is equal to the probability of reaching I_k from any of its direct predecessors after the emission of symbols x_1 to $x_{\ell-1}$, which is finally multiplied by the emission probability of x_ℓ at state I_k.

Similarly, we define the forward variables of match state M_k using its predecessors from the

previous column of the plan9 architecture (cf. figure 4.53) as

$$\begin{aligned}\mathcal{F}_{M_k}(\ell, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta}) = \big[&\mathcal{F}_{I_{k-1}}(\ell-1, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})\ P_T(M_k|I_{k-1}, \boldsymbol{\beta}_{T,I_{k-1}}) + \\ &\mathcal{F}_{D_{k-1}}(\ell-1, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})\ P_T(M_k|D_{k-1}, \boldsymbol{\beta}_{T,D_{k-1}}) + \\ &\mathcal{F}_{M_{k-1}}(\ell-1, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})\ P_T(M_k|M_{k-1}, \boldsymbol{\beta}_{T,M_{k-1}})\big]\ P_{M_k}(x_\ell|\boldsymbol{r}, \boldsymbol{\beta}_{M_k}). \end{aligned} \tag{4.95}$$

Finally, we define the forward variables of delete state D_k as

$$\begin{aligned}\mathcal{F}_{D_k}(\ell, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta}) = &\mathcal{F}_{I_{k-1}}(\ell, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})\ P_T(D_k|I_{k-1}, \boldsymbol{\beta}_{T,I_{k-1}}) + \\ &\mathcal{F}_{D_{k-1}}(\ell, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})\ P_T(D_k|D_{k-1}, \boldsymbol{\beta}_{T,D_{k-1}}) + \\ &\mathcal{F}_{M_{k-1}}(\ell, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta})\ P_T(D_k|M_{k-1}, \boldsymbol{\beta}_{T,M_{k-1}}), \end{aligned} \tag{4.96}$$

where the position ℓ does not increase, since the delete state does not emit a symbol of $\boldsymbol{x}$.

We initialize the forward variables as follows: We can observe D_0 only before the emission of the first symbol. Hence, we define

$$\mathcal{F}_{D_0}(\ell, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta}) = \begin{cases} 1 & \text{if } \ell = 0 \\ 0 & \text{otherwise.} \end{cases}. \tag{4.97}$$

We cannot reach M_0 in any admissible path and, consequently,

$$\forall \ell : \mathcal{F}_{M_0}(\ell, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta}) = 0. \tag{4.98}$$

Finally, the forward variables of all emitting states for $\ell = 0$ are defined as

$$\forall S_k \in \{I_1, M_1, I_2, \ldots, I_{K+1}, M_{K+1}\} : \mathcal{F}_{S_k}(0, \boldsymbol{x}|\boldsymbol{r}, \boldsymbol{\beta}) = 0. \tag{4.99}$$

In addition to the CoProHMM, which models the class of target sites, we use a homogeneous Markov model of order 1 (see section 3.3.1, p. 22) with parameters $\boldsymbol{\beta}_{hMM,bg}$ to model the class of background, i.e. non-target, sequences:

$$P(\boldsymbol{x}|bg, \boldsymbol{r}, \boldsymbol{\beta}_c) = P_{hMM(1)}(\boldsymbol{x}|\boldsymbol{\beta}_{hMM,bg}). \tag{4.100}$$

We define the class posterior using the class-conditional likelihoods $P(\boldsymbol{x}|target, \boldsymbol{r}, \boldsymbol{\beta}_{target})$ of equation (4.92) and $P(\boldsymbol{x}|bg, \boldsymbol{r}, \boldsymbol{\beta}_c)$ as

$$P\left(c \,|\, \boldsymbol{x}, \boldsymbol{r}, \boldsymbol{\beta}\right) = \frac{P(c|\boldsymbol{\beta})P(\boldsymbol{x}|c, \boldsymbol{r}, \boldsymbol{\beta}_c)}{\sum_{\tilde{c}} P(\tilde{c}|\boldsymbol{\beta})P(\boldsymbol{x}|\tilde{c}, \boldsymbol{r}, \boldsymbol{\beta}_{\tilde{c}})}, \tag{4.101}$$

where $P(c|\boldsymbol{\beta})$ denotes the a-priori probability of class c. We parameterize the a-priori class probabilities as

$$P(c|\boldsymbol{\beta}) = \frac{\exp(\beta_c)}{\sum_{\tilde{c}} \exp(\beta_{\tilde{c}})}, \tag{4.102}$$

where $\beta_c \in \mathbb{R}$.

As an additional requisite of the MSP principle, we define a prior on the parameters $\boldsymbol{\beta}$. For the homogeneous Markov model of class *bg*, we use a transformed product-Dirichlet prior as defined in section 3.4.2 (p. 30) with equivalent sample size (ESS) α_{bg} and expected length $L_E = 22$. We define another product-Dirichlet prior for the parameters of the emission and transition probabilities, which is – in analogy to Markov models of order 0 – the product of independent transformed Dirichlet priors for each set of transition parameters $\boldsymbol{\beta}_{T,S_k}$, and each set of emission parameters $\boldsymbol{\beta}_{I_k}$ and $\boldsymbol{\beta}_{M_k|b} = (\beta_{A|b,M_k}, \ldots, \beta_{U|b,M_K})$, $b \in \{A, C, G, U\}$ living on a common simplex. For each state S_k, we define a local ESS α_{target,S_k} depending on the ESS α_{target} of class *target*. We set the local ESS of D_0 and D_{K+1} to $\alpha_{target,D_0} = \alpha_{target,D_{K+1}} = \alpha_{target}$, since these two nodes are at the beginning and the end of every admissible path in the CoProHMM. We set the local ESS of the remaining delete states to $\alpha_{target,D_k} = \frac{\alpha_{target}}{10}$, we set the local ESS of the insert states to $\alpha_{target,I_k} = \frac{\alpha_{target}}{10}$, and we set the local ESS of the main states to $\alpha_{target,M_k} = \frac{8 \cdot \alpha_{target}}{10}$, which represents the a-priori assumption that the main states should be used more frequently than delete and insert states.

Using these local ESSs, we further define the hyper-parameters of the transformed Dirichlet priors for the transition parameters $\beta_{V|S_k}$ as $\alpha_{D_{k+1}|S_k} = \frac{\alpha_{target,S_k}}{10}$, $\alpha_{I_{k+1}|S_k} = \frac{\alpha_{target,S_k}}{10}$, and $\alpha_{M_{k+1}|S_k} = \frac{8 \cdot \alpha_{target,S_k}}{10}$. According to the assumption of uniform pseudo data, we set the hyper-parameters of the emission parameters $\beta_{a|I_k}$ of the insert states to $\alpha_{a|I_k} = \frac{\alpha_{target,I_k}}{4}$, $a \in \{A, C, G, U\}$, and we set the hyper-parameters of the emission parameters $\beta_{a|b,M_k}$ of the match states to $\alpha_{a|b,M_k} = \frac{\alpha_{M_{k+1}|S_k}}{16}$, $a, b \in \{A, C, G, U\}$.

Finally, we define a transformed Beta prior on the parameters $\boldsymbol{\beta}_c$ of the a-priori class probabilities using hyper-parameters α_{bg} and α_{target}. In the following experiments, we use $\alpha_{bg} = \alpha_{target} = 4$.

4.5.3. Data

We extract all human target sites and associated miRNAs predicted by TargetScan, RNAhybrid, and miRanda from miRNAMap[8] (Hsu et al., 2008). Additionally, we consider predicted target sites and associated miRNAs of DIANA-microT (Maragkakis et al., 2009), which are kindly provided by Manolis Maragkakis (personal communication). From each of these four data sets, we randomly sample 200,000 target sites and associated miRNAs as training data sets for the subsequent study. We refer to these data sets as *TargetScan data set*, *RNAhybrid data set*, *miRanda data set*, and *DIANA-microT data set*.

From miRecords[9] v. 1 (Xiao et al., 2009), we additionally obtain predicted and verified human target sites and associated miRNAs. In this case, we only consider target sites with experimental evidence, i.e. predicted target sites in UTRs of genes that are validated targets of the corresponding miRNA, and target sites that are directly validated, e.g. by mutation experiments. This data set contains 667 indirectly verified target site and 12 directly verified target sites. We refer to this data set as *miRecords data set*.

[8] `ftp://mirnamap.mbc.nctu.edu.tw/miRNAMap2/miRNA_Targets/Homo_sapiens/miRNA_targets_hsa.txt.tar.gz`

[9] `http://mirecords.biolead.org/download_data.php?v=1`

Additionally, we create a data set joining the data from the different sources. To this end, we randomly select 60,000 target sites from the TargetScan data set, 20,000 target sites from the RNAhybrid data set, 20,000 target sites from the miRanda data sets, 60,000 target sites of the DIANA-microT data set, and all target sites from the miRecords data set. We select a greater number of predictions for TargetScan and DIANA-microT than for RNAhybrid and miRanda, since these two approaches achieved the best accuracy in the study by Selbach et al. (2008). We refer to this data set as *joint data set*.

We generate a background data set, i.e. a data set of non-target sites, by drawing sub-sequences of length 30 from 3'-UTRs of human genes according to NCBI Genbank[10], human genome build 37.1. We choose UTRs to avoid a potential bias in base composition or other properties of DNA, which could perturb the subsequent prediction of target sites. We assign each of these sub-sequences a miRNA randomly selected from the mature human miRNAs listed at miRBase[11] (Griffiths-Jones et al., 2006, 2008). The resulting background data set comprises 100,000 non-target sites and associated miRNAs.

In order to represent our confidence in the predicted and verified target sites, we assign a weight to each pair of target site and miRNA. For the predictions of TargetScan, RNAhybrid, miRanda, and DIANA-microT, and for the background data set, we use weights of 1. For the miRecords data set, we use a weight of 50 for indirectly validated target sites, and a weight of 500 for directly validated target sites.

For learning the CoProHMMs by the discriminative MSP principle, we use the sequences of the miRNAs in 5'-3' orientation and the sequences of the target and non-target sites in 3'-5' orientation, since the employed CoProHMMs consider at most 22 nt of the miRNA and additional symbols in the miRNA at the 3' end are omitted.

4.5.4. Results & Discussion

We use a graphical representation to illustrate the CoProHMMs learned by the discriminative MSP principle. The representation of one column of a CoProHMM is depicted in figure 4.54. Like in figure 4.53, rectangles represent match states, diamonds represent insert states, and circles represent delete states. The thickness of outgoing edges represents the transition probabilities to the successors of a node, where thicker edges correspond to a higher transition probability.

We illustrate the emission probabilities of insert states by a row of colored boxes. The color of a box corresponds to the nucleotides, the saturation of each box represents the emission probability, and the brightness of the colors in a row represents the deviation of the corresponding probability distribution from a uniform distribution over the four nucleotides. In analogy, the conditional emission probabilities of match states are represented by a matrix comprising such colored rows, where each row corresponds to the probability distribution conditional on one of the nucleotides observed in the miRNA. From the match state in figure 4.54, we observe bright colors for complementary nucleotides in the target site and the miRNA, e.g. U in the

[10] `http://www.ncbi.nlm.nih.gov`
[11] `http://www.mirbase.org`

target site given an A in the miRNA is represented by the bright red box in the upper right of the matrix.

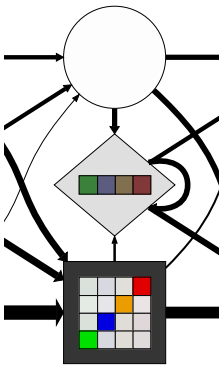


Figure 4.54.: One column of the graphical representation of a CoProHMM. The darkness of the background of nodes represents how frequently nodes are visited according to the forward variables. The thickness of edges represents transition probabilities. The colored row vectors within the diamonds representing insert states reflect emission probabilities: green corresponds to A, blue corresponds to C, orange corresponds to G, and red corresponds to U. The saturation of these colors represents the corresponding emission probabilities and the brightness represents the deviation from a uniform distribution. In colored matrices within the rectangles representing match states, each row corresponds to one nucleotide observed in the miRNA in the order A, C, G, U and the colors within this row represent the conditional emission probabilities given the nucleotide in the miRNA in analogy to the emission probabilities of the insert states.

We also compute the probability of visiting a state from the forward variables of all input sequences. These probabilities are visualized by the darkness of the background of each node. The darker the background of a node the higher the probability of visiting this node. In nodes representing delete states, this background fills the complete circle, while for insert and match states the background is partly covered by the colored row or matrix representing emission probabilities. In figure 4.54, the match state is visited with the highest probability, the insert state is visited with a fairly low probability, and the delete state is visited with a probability of almost 0.

We present the graphical representation of the CoProHMMs learned on the miRanda data set, the TargetScan data set, the miRecords data set, and the joint data set in figure 4.55. We learn CoProHMMs on the predictions of miRanda and TargetScan to investigate if CoProHMMs can adapt to the characteristics of these approaches. Considering the CoProHMM learned on the miRanda data set, which is depicted in figure 4.55(a), we find that from position 1 to 7 the corresponding match states are used with the highest probability, whereas insert and delete states obtain a low probability. Starting from position 8, these probabilities are gradually shifted to the insert and delete states, and between positions 13 and 19 the three types of states are visited with almost identical probabilities. From position 20 to 22, the probability of the match states gradually decreases and is shifted predominantly to the delete states. One reason for the latter observation might be that some of the targets predicted by miRanda are shorter than 22 nt, which necessitates visiting delete states to reach the final state D_{K+1}.

Turning to the emission probabilities of the insert states, we observe that the corresponding probability distributions are close to a uniform distribution at all positions as indicated by the

reduced brightness of the colored boxes in the graphical representation. From the conditional emission probabilities of the match states, we observe a general tendency to complementary base pairings between the target site and the miRNA. This tendency is especially pronounced for the match states at position 2 to 8 in the seed region, but can also be observed for the match state at position 1 and those at position 9 to 19 of the miRNA, whereas it declines over the remaining 3 positions. In the graphical representation, we also detect a slight preference for G:U wobble basepairs compared to the remaining non-complementary basepairs, as the red box in the last column of the third row and the orange box in the third column of the last row exhibit a slightly increased saturation.

If we relate these finding to the miRanda approach, we can attribute the majority of observations to characteristics built into miRanda: the preference for complementary basepairs across almost all match states and the slight preference for G:U wobble basepairs are most likely a result of the Smith-Waterman like alignment employed by miRanda. Additionally, miRanda assigns a weight of 2 to the first 11 positions of the alignment, which is also reflected by the increased probabilities of visiting match states in the seed region, although this preference already begins to decline at position 8 of the learned CoProHMM.

As a second example, we consider the CoProHMM learned on the TargetScan data set in figure 4.55(b). We observe a similar pattern of preferences regarding the probability of visiting states as for the miRanda data set. The main differences are an increased probability of visiting the insert and delete states up to position 1, a more abrupt shift of probability from the main states to predominantly the insert states starting from position 8, and an exceptionally high probability of visiting the insert state at position 22, most likely cycling several times in this state as indicated by the thickness of the recursive edge. The latter could again be explained by the length of the target sites predicted by TargetScan and by the preferential use of insert states at columns 8 to 18.

Notable differences between the CoProHMM learned on the TargetScan data set and that learned on the miRanda data set can be observed for the conditional emission probabilities at the match states. At position 2 to 8 if figure 4.55(b), we find complementary basepairs almost exclusively, while a slight preference for complementary basepairs is present at the bordering positions 1 and 9. In contrast, the remaining positions exhibit only very slight preferences for specific basepairs.

Again, these findings are closely related to the main characteristics built into TargetScan. The perfect complementarity at position 2 to 8 of the CoProHMM reflects the according requirement of TargetScan. We also observe a reduced preference for complementary basepairs at positions 1 and 9, which most likely can be attributed to the fact that initial perfect matches in the seed region may be elongated to either side in TargetScan. However, we do not observe a preference for G:U wobble basepairs at these two positions, although these are allowed in TargetScan.

We also learn CoProHMMs on the RNAhybrid and DIANA-microT data sets (not shown). For the RNAhybrid data set, we observe similar characteristics as for the miRanda data set. The major exception exist for the insert state at position 22 of the CoProHMM, which is more similar to that for the TargetScan data set, and for the conditional probability distributions

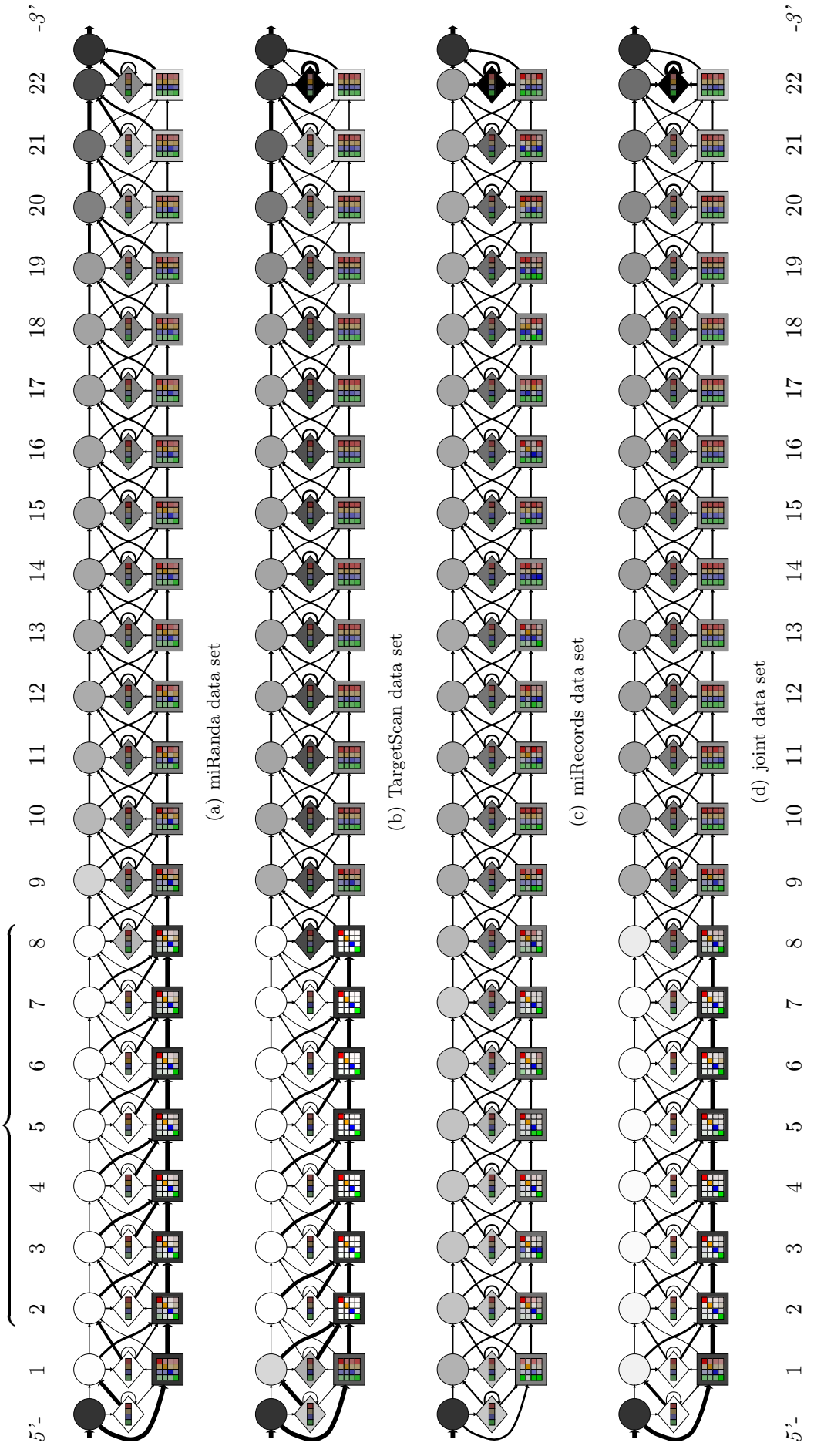


Figure 4.55.: CoProHMMs learned on the miRanda data set, TargetScan data set, miRecords data set, and joint data set.

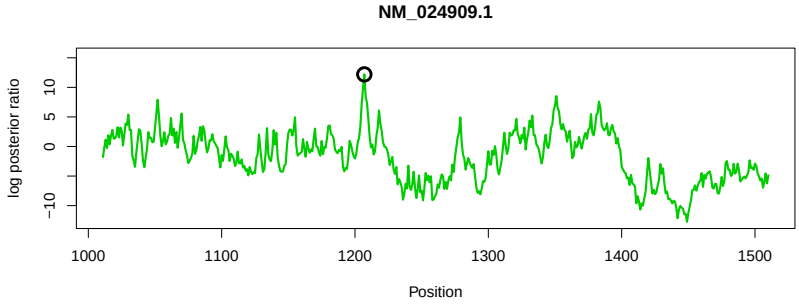
of the match states in the seed region, which exhibit a slightly reduced preference for complementary basepairs. The CoProHMM learned on the DIANA-microT data set is similar to that for the TargetScan data set. However, in this case the insert and delete states up to position 1 are visited with a probability of almost 0, and the preference for complementary basepairs already begins to decline at position 8, which is most likely an effect of 6 nt seed, which are also allowed in DIANA-microT.

In figure 4.55(c), we present the CoProHMM learned on the miRecords data set, which comprises predicted target sites from the UTRs of verified target genes as well as directly verified target sites. On this data set, we observe similar general tendencies as for the miRanda and TargetScan data sets. These are high probabilities for the match states in the seed region, a shift of probabilities to insert and delete states outside the seed region, and a strong preference for complementary basepairs in the match states of the seed region. However, we also observe variations of these rules e.g. at the conditional emission probabilities of the main states at position 3 and 5, and a rather erratic preference for specific nucleotides outside the seed region. These variations are likely due to the limited size of the miRecords data set, which comprises 679 target sites and associated miRNAs, resulting in over-fitting effects especially for the conditional probability distributions of the match states.

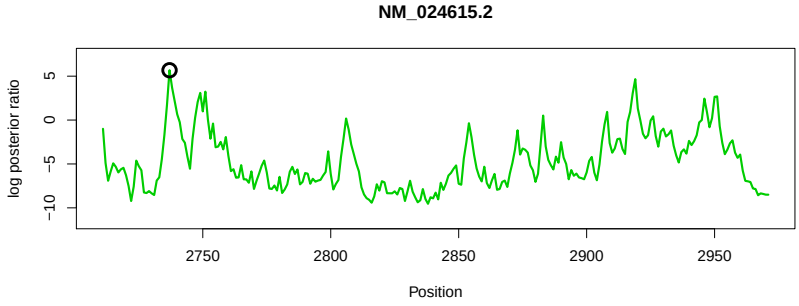
Finally, we consider the CoProHMM learned on the joint data set, which is presented in figure 4.55(d). As expected, this CoProHMM combines characteristics of the considered approaches. The seed region exhibits a high similarity to that of the CoProHMM learned on the TargetScan data set, although the preference for complementary basepairs is less stringent. In contrast, the shift of probabilities from match states to insert and delete states appears to be less abrupt than for the TargetScan data set and more similar to that of the CoProHMM learned on the miRanda data set. Additionally, the match states outside the seed region exhibit a preference for complementary basepairs which is stronger than can be observed for the TargetScan data set, but less articulate than for the miRanda data set.

We use the classifier comprising this CoProHMM and the corresponding homogeneous Markov model in the subsequent studies, when we predict putative target sites of given miRNAs. In the following, we consider three human miRNAs, namely hsa-miR23a, hsa-miR145, and hsa-miR196a, with the largest number of experimentally verified target sites according to miRecords. These miRNAs and all associated target sites are excluded from the joint data set to avoid an overlap of training and test data. Given each of these miRNAs, we compute the log class posterior ratios of the class posterior given the CoProHMM and the class posterior given the homogeneous Markov model for each overlapping sub-sequence of length 30 of 3'-UTRs of human genes. In the following, we illustrate the results for the UTRs of three genes, each containing of verified target sites of one of the miRNAs considered.

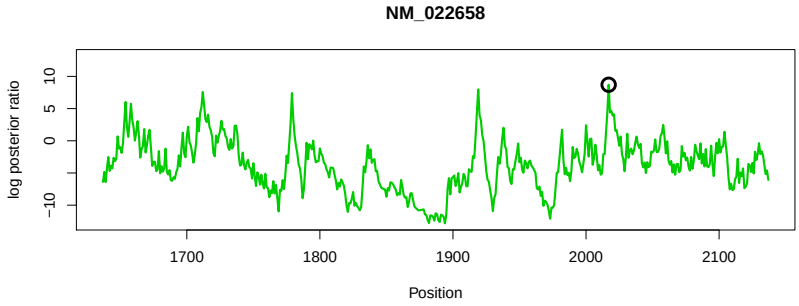
Figure 4.56(a) depicts the profile of log class posterior ratios for a 500 nt fragment of the UTR of gene NM_024901.1 given the miRNA hsa-miR23a. The position of the experimentally verified target site is marked by a black circle. We find that the sub-sequence at this position clearly achieves the largest log class posterior ratio of all considered positions. This indicates that the rules of miRNA-target site interaction inferred by CoProHMMs from the data are also suited to predict target sites of miRNAs that have not been part of the training data.



(a) hsa-miR23a



(b) hsa-miR145



(c) hsa-miR196a

Figure 4.56.: Profile of log class posterior ratios of the classifier learned on the joint data set using a CoProHMM and a homogeneous Markov model of order 1 for three UTRs given the miRNAs hsa-miR23a, hsa-miR145, and hsa-miR196a, respectively. The annotated positions of experimentally verified target sites are indicated by black circles.

The picture for the UTR of the human gene NM_02415.2, which is of length 291 bp, given the miRNA hsa-miR145 in figure 4.56(b) is similar. Again, we observe the greatest log class posterior ratio for the sub-sequence at the position of the experimentally verified target site. However, in this case the log class posterior ratio is only marginally greater than that of another sub-sequence approximately at position 2920. Considering a fragment of length 500 nt of the 3'-UTR of NM_022658 given the miRNA hsa-miR196a in figure 4.56(c), we find two additional distinct peaks at positions 1779 and 1919 of the log class posterior ratio besides that of the experimentally verified target site, which reach comparable values of the log class posterior ratio.

To investigate if these additional predictions are putative target sites of hsa-miR196a, we compute the alignments of these three sequences and the sequence of hsa-miR196a using the Viterbi algorithm for the CoProHMM. The resulting three alignments are presented in figure 4.57, where the first two alignments correspond to the two additionally predicted target sites, and the third alignment comprises the experimentally verified target site. We find that the first two alignments exhibit a perfect complementarity at the first 8 positions of the alignment. In both cases, we observe a considerable number of gaps at the 3' end of the miRNA, which are most likely due to the high probability of the self transition of the last insert state of the CoProHMM (cf. figure 4.55(d)). According to the annotation of miRecords, these two putative target sites are also predicted by miRanda and TargetScan. In the third alignment, 10 perfectly complementary positions are intercepted by two G:U wobble basepairs. In turn, the third alignment also exhibits multiple complementary basepairs outside the seed region, while only individual complementary basepairs can be observed for the regions outside the putative seed in the other two alignments. This alignment is in agreement with the annotation of the verified target sites in miRecords, and is also predicted by miRanda and RNAhybrid.

```
utr=NM_022658 end=1779 score=7.41
miRNA:  5' UAGGUAGUUUCAUGUUGUUGGG---------- 3'
           |||||||| |  |   :
target: 3' AUCCAUCAUA-CAAUUUCU---AAAUAAUAAUAA 5'


utr=NM_022658 end=1919 score=7.98
miRNA:  5' UAGGUAGUU-UCAUGUUGUUGGG------- 3'
           ||||||||  | | :|   : ||
target: 3' AUCCAUCACUAUUUUAUGCGACCUCCAAAG 5'


utr=NM_022658 end=2017 score=8.71
miRNA:  5' UAGGUAGUUUCAUGUUGUUGGG--------- 3'
           ||||:|||:||| |||||||||
target: 3' AUCCGUCAGAGU-CAACAACCCAAAAGAAUC 5'
```

Figure 4.57.: Alignment of putative target sites in the UTR of NM_022658 to the sequence of the miRNA hsa-miR196a according to the CoProHMM learned on the joint data set. Vertical lines indicate perfectly complementary basepairs, while colons indicate G:U wobble basepairs.

4.5.5. Conclusions

We propose conditional profile HMMs (CoProHMMs), which are an extension of profile HMMs, for modeling the target sites of given miRNAs. As a proof of concept we learn the parameters of CoProHMMs on the predictions of several existing algorithms and associated miRNAs, and we demonstrate that CoProHMMs adapt well to the characteristics of the considered approaches. This gives indication that CoProHMMs might also be capable of learning general rules of miRNA-target site interactions from verified target sites if these become available in sufficient quantities.

Additionally, we learn a CoProHMM on a data set incorporating predicted as well as experimentally verified target sites of given miRNAs. On an independent test data set, we show that the learned CoProHMM predicts verified target sites of miRNAs that have not been part of the training data. We scrutinize additionally predicted putative target sites and find that these might be functional target sites as well.

For the prediction of target *genes* exhibiting possibly multiple target sites of different miRNAs, the next step would be to enclose CoProHMMs in another model, e.g. a semi-hidden Markov model.

5. Implementation

For all experiments presented in this work, we use Jstacs (Grau et al., 2008), an open-source Java framework for the statistical analysis and classification of biological sequence data. Jstacs is a joint project of the groups "Pattern Recognition and Bioinformatics" and the "Bioinformatics" at the Institute of Computer Science of Martin Luther University Halle–Wittenberg and the "Research Group Data Inspection" at the Leibniz Institute of Plant Genetics and Crop Plant Research in Gatersleben. We created Jstacs with the goal of having ready-to-use implementations for frequently recurring tasks like reading and representing sequence data or assessing different classifiers, and providing a standardized foundation for the implementation of new statistical models or classifiers. Since Jstacs is open-source software, we make these facilities available to the scientific community. In the following, we give an introduction to the general structure of Jstacs, and we illustrate the utility of Jstacs for one specific example.

For sequence data, we use a numerical representation of the discrete symbols, which for instance allows for an efficient access to elements of arrays. This numerical representation is encapsulated in the class `Sequence`, which also holds the mapping from numerical values to the original symbols and provides methods for the access to single symbols, the excision of subsequence, or the determination of the reverse complementary sequence. In Jstacs, `Sequence`s are aggregated in `Sample`s representing data sets, which for instance provide methods for generating random partitionings of the original `Sample` required for holdout sampling. Jstacs also comprises an adaptor to the sequence representation of BioJava (Holland et al., 2008).

Statistical models that are to be learned by discriminative learning principles like MCL or MSP, must either implement the interface `NormalizableScoringFunction` or extend the abstract class `AbstractNormalizableScoringFunction`. `NormalizableScoringFunction` extends the interface `ScoringFunction` which is intended for more general scoring functions – as opposed to statistical models – that do not necessarily define a proper probability distribution over the input sequences. For defining a new statistical model, a user of Jstacs only needs to implement methods that compute the log likelihood of a sequence given the model and the corresponding gradient with respect to the parameters, and to compute the value and gradient of the prior. Additionally, Jstacs requires the specification of a method that returns an XML-representation of the learned model, which can later be used to restore the model for additional analyses.

Jstacs defines a hierarchy of inheritance for classifiers using different kinds of scoring functions or statistical models. The abstract class `AbstractClassifier` constitutes the root of this hierarchy and defines standardized methods for the training of a classifier, the classification of sequences, or the evaluation of several performance measures on a given test data set. The latter methods are implemented in the abstract sub-class `ScoreClassifier`, which comprises one `ScoringFunction` for each class considered. A specific implementation of a classifier learned by the discriminative MSP principle is available as the class `MSPClassifier` defined for

NormalizableScoringFunctions. The user may choose from different numerical optimization techniques including conjugate gradients and second order quasi-Newton methods supplied by Jstacs for learning the parameters of the a-priori class probabilities of this classifier and the parameters of the enclosed NormalizableScoringFunctions,

The assessment of different classifiers in cross validation or holdout experiments in Jstacs is established by the abstract class ClassifierAssessment. Standardized extensions of this abstract class for cross validation (KFoldCrossValidation) and stratified holdout sampling (RepeatedHoldOutExperiment) are defined on AbstractClassifiers, which makes different classifiers readily comparable. Additionally, Jstacs provides a generic implementation of all performance measures considered in this work in the class ScoreBasedPerformanceMeasure-Definitions, which is also employed by the class ScoreClassifier.

Since we designed Jstacs strictly object oriented and due to the different levels of inheritance, algorithms like the numerical optimization techniques or classifiers like the MSPClassifier are implemented on a high level of abstraction. This facilitates a great variability of the tasks that can be accomplished using Jstacs and allows for a modular design of novel approaches. For instance, we implement a ScoringFunction in Jstacs that represents Bayesian networks that are to be learned by one of the discriminative learning principles. Since inhomogeneous Markov models are a special case of Bayesian networks, this implementation can be used for all applications presented in this work that require inhomogeneous Markov models. By this means, we can use the identical implementation of inhomogeneous Markov models for the prediction of transcription factor binding sites, for the PWM model that represents the motif for de-novo motif discovery, for the elementary classifier using an inhomogeneous Markov model of order 1 for the prediction of nucleosome positioning, and for the PWM models at the leaves of the decision tree model.

Similarly, we define an abstract class representing parameter priors, which is inherited by the implementations of Gaussian, Laplace, and transformed product-Dirichlet priors. Since the MSPClassifier is defined on the abstract prior, we can easily plug in Gaussian, Laplace, and Dirichlet priors. As mentioned above, the implementation of stratified holdout sampling is defined on an abstract superclass of the MSPClassifier, which is also the superclass of classifiers that are to be learned generatively.

With the classes described in the previous two paragraphs, we have all methods at hand that we need for the first study presented in this work, namely the evaluation of classifiers learned by the MAP and MSP principle for the prediction of transcription factor binding sites. The source code of the main-method that accomplishes the experimental part of the complete study is given in appendix A.5 and illustrates how such a study can be conducted with marginal effort using Jstacs. In a similar manner, the models for all applications studied in this work are implemented using Jstacs. Other ScoringFunctions implemented for this work include the ZOOPS and MuMo models, and the position distributions employed for the de-novo discovery of cis-regulatory modules in section 4.2, the decision tree model used in section 4.4, the conditional profile HMM of section 4.5, and the Gaussian and gamma densities used in section 4.3. Since Jstacs is open-source software under GPL[1], all classes used in this work are either already publicly available or will be made public in a future release of Jstacs.

[1] GNU General Public License

6. Conclusions

In this work, we investigate the utility of the discriminative maximum supervised posterior (MSP) principle for statistical sequence analysis and classification in bioinformatics. While generative principles for learning the parameters of statistical models aim at an accurate representation of the distribution of the data, discriminative learning principles are tailored to an accurate classification of the data. The MSP principle optimizes the parameters of employed statistical models with respect to the supervised posterior, which is the product of the conditional likelihood and a prior on the parameters of the models. While conditional likelihood is a measure for the accuracy of classification on the training data, the prior helps to avoid over-fitting and allows to incorporate a-priori assumptions aside the training data. Hence, the MSP principle is a discriminative analogon of the generative maximum a-posteriori (MAP) principle, which amounts to the product of likelihood and prior.

We apply the MSP principle to the prediction of sequence signals that are related to biological processes influencing the product of a gene at different stages of gene expression. Applications considered in this work are the prediction of transcription factor binding sites, the discrimination of nucleosome-bound sequences and linkers, the prediction of donor splice sites, and the prediction of miRNA target sites. In case of prediction, the classification task is to distinguish functional sites from non-functional sites. We additionally consider the de-novo discovery of transcription factor binding sites and cis-regulatory modules, where the goal is to discover motifs and binding sites that are specific for the promoters of a target set of co-regulated genes. In the discriminative context, this translates to the task of finding that motif or set of motifs which is suited best to distinguish the promoters in the target set from the promoters of other, unrelated genes.

Considering the prediction of transcription factor binding sites, we learn the parameters of inhomogeneous Markov models of different orders by the MSP principle for ten data sets comprising binding sites of different transcription factors of mammals, *A. thaliana*, and *E. coli* and associated background data sets. Since Markov models in standard parameterization are not suited for numerical optimization, we use a parameterization of Markov models in terms of real-valued parameters, and we derive a prior on these parameters by transforming the conjugate product-Dirichlet prior to this parameterization. Hence, we can use the equivalent product-Dirichlet priors for the MSP and the MAP principle, which avoids a potential bias on the results due to the choice of different priors. We find that the discriminative MSP principle with product-Dirichlet prior significantly outperforms the generative MAP principle for the majority of the studied data sets. Since we use identical hyper-parameters for the product-Dirichlet prior on the parameters of the Markov models for the MSP and the MAP principle, we may conclude that in this case the improvement of classification performance can be attributed to the discriminative learning principle alone. In a subsequent study, we

investigate how the MSP and MAP principle are affected by the size of training data. We find, that the MSP principle again outperforms the MAP principle in the majority of cases yielding an improved classification performance even for small data sets.

The results for the prediction of transcription factor binding sites indicate that the discriminative MSP principle can be of value for sequence classification as a general concept. Hence, we employ the MSP principle for other problems related to sequence classification as well. However, we do not focus on the direct comparison of learning principles in the following studies, but we broaden benchmarks and comparisons to other state-of-the-art approaches that have been proposed for these specific problems.

Since in real-world problems, we often neither know the exact location of binding sites nor the binding motif of the transcription factor of interest, the de-novo discovery of transcription factor binding sites or cis-regulatory modules is highly relevant for the elucidation of transcriptional regulation. For the de-novo discovery of cis-regulatory modules, we learn the parameters of an extended ZOOPS by the discriminative MSP principle. Here, we extend the ZOOPS model to two motifs, which may occur coordinately in promoters, representing cis-regulatory modules comprising binding sites of two different transcription factors. We include a Gaussian position distribution of the binding sites into the model, since the binding sites of most transcription factors are known to occur non-uniformly distributed in the promoters. Again, we apply the transformed product-Dirichlet prior to the parameters of the sequence models. We derive a parameterization of the bivariate Gaussian distribution using unconstrained parameters, and we transform the conjugate normal-Wishart prior accordingly. Additionally, we develop a heuristic to automatically adapt the length of the motif and to compensate for phase shifts.

As a first benchmark study, we compare this approach – called MuMFi – considering a single motif to several other approach for de-novo motif discovery, namely MEME, Gibbs Sampler, A-GLAM, Weeder, Improbizer, DME, and DEME. We find that MuMFi is the only approach that can successfully discover the correct motif in 18 benchmark data sets. If we do not specify the correct motif length in advance, MuMFi still discovers all motifs with high accuracy, whereas the other approaches perform considerably worse. Weeder is the only of the approaches studied that discovers 5 of the 9 motifs, while other approaches like Improbizer or A-GLAM yield a higher accuracy for single data sets, but recover a smaller number of motifs correctly. These results demonstrate that the combination of discriminative learning of the parameters, incorporating a model for the position distribution of binding sites, and an automatic adaption of the motif length are highly beneficial for the de-novo discovery of single motifs.

In a second benchmark study, we compare MuMFi for cis-regulatory modules comprising binding sites of two motifs to other approaches that are specifically designed for the de-novo discovery of cis-regulatory modules, namely CisModule, CoBind, and MoAn. We find that MuMFi performs comparable or even better than the existing approaches and, again, the performance of MuMFi can be attributed to the combination of the MSP principle and the explicit modeling of the position distribution.

Finally, we apply MuMFi to promoters of auxin responsive genes, and find a motif that can be interpreted as a refined and elongated variant of the canonical auxin response element. We

also find that occurrences of this motif exhibit a clear positional preference centered around a mean value approximately 130 bp upstream of the transcription start site.

Approaches for the computational prediction of transcription factor binding sites typically suffer from a large number of false-positives, i.e. non-functional predicted binding sites. Since DNA bound in nucleosomes is virtually inaccessible to transcription factors due to steric hindrance, an accurate prediction of nucleosome positioning, especially nucleosome-free regions, may help to exclude a subset of the false-positives from predictions. Hence, we investigate if the MSP principle combined with an appropriate model may improve the computational prediction of nucleosome positioning as well.

The approach proposed in this work uses a two-stage process. First, we use one classifier to distinguish coding sequences, non-coding sequences, and sequences at the border between coding and non-coding regions, since we anticipate that the signals of nucleosome positioning may be superimposed and, hence, blurred by general properties of coding and non-coding sequences. Second, we use independent component classifiers for each of the three types of sequences that distinguish nucleosome-bound from nucleosome-free sequences. Each component classifier combines simple elementary classifiers by weighted voting. The employed elementary classifiers are selected independently for each component classifier by a greedy approach. For learning the parameters of component classifiers, we employ a novel variant of the MSP principle for soft-labelling. The votings of the component classifiers are also combined by weighted voting. However, we extend previous ensemble approaches by using weights that depend on the sequence to be classified. These weights correspond to the probability that a sequence is coding, non-coding, or at a border according to the first classifier. Additionally, we include information about preferred lengths of the linkers between nucleosomes into the final prediction.

We compare the classification accuracy of this approach to the current state-of-the-art approach on a ground truth obtained by parallel sequencing of nucleosome-bound sequences in yeast. We find that the proposed approach distinguishes nucleosome-bound and nucleosome-free sequences with a considerably higher accuracy than the existing approach for all levels of coverage considered. Interestingly, this improved accuracy is achieved, although the component classifiers do not comprise elementary classifiers that can detect periodic signals, which are known to be relevant for nucleosome positioning. We assume that such periodic signals determine the local positioning of nucleosomes, but do not reflect the general tendency of a sequence to be bound in a nucleosome. In contrast, other known determinants of nucleosome formation are supported by the learned classifiers as the inhibition of nucleosome formation by poly-A/T tracts or the preference for CTG trinucleotides in nucleosome-bound sequences.

To investigate the relevance of the improved classification performance, we compare the predictions of the two approaches in their genomic context. We observe several nucleosome-free regions in putative promoters that are correctly predicted by the proposed approach but not by the previous approach. We also find well-positioned nucleosomes in promoter regions that are only discovered by the proposed approach. Hence, we may conclude that the observed improvement in classification accuracy is especially relevant if we utilize the predicted nucleosome positions to reduce the number of false-positive predictions of transcription factor binding sites.

Splicing is another process that determines the product of a gene. In this work, we propose a novel approach called maximum supervised posterior decomposition for the prediction of donor splice sites, which employs the MSP principle as well. In this case, we use a decision tree model with simple position weight matrix models at its leaves, which has originally been proposed for the generative maximal dependence decomposition approach. Here, we learn the parameters of the decision tree model including the parameters of the position weight matrices by the MSP principle, and we select the structure of the decision tree in a greedy algorithm by means of the supervised posterior. We prove that the supervised posterior using these decision tree models in the proposed parameterization is a log concave function of the parameters. Hence, we obtain globally optimal parameters by numerical optimization regardless of the initialization.

We compare the classification accuracy of maximum supervised posterior decomposition to several other approaches for the prediction of donor splice sites, namely weight array models, permuted variable length Markov models, Markov models learned by the MSP principle, maximum entropy models, and maximal dependence decomposition. As benchmark data sets, we use donor splice sites and decoy sites from *A. thaliana*, *D. melanogaster*, *D. rerio*, *C. elegans*, and *H. sapiens*. We find that maximum supervised posterior decomposition yields a similar or even improved classification performance compared to the existing approaches for all data sets and considered performance measures.

Scrutinizing some of the decision tree models learned, we find many known properties of donor splice sites supported. The exploration of decision trees is supported by a novel variant of sequence logos that facilitates the perception of differences between donor splice sites and decoy sites. The properties discovered include the importance of position +5 on the intron side and position −1 on the exon side and a compensatory effect between the intron and exon side, where a strong binding on the exon side may compensate for a lower binding affinity on the intron side and vice versa. In addition to previous findings, we observe that a lack of the consensus at position +5 greatly reduces the relevance of the adjacent positions +4 and +6. Interestingly, the discrimination of canonical donor splice sites with the consensus GT at positions +1 and +2 and non-canonical donor splice sites seems to be of minor importance for the general recognition of donor splice sites. We additionally use maximum supervised posterior decomposition as an exploratory tool to detect differences between the donor splice sites of different organisms. Comparing donor splice sites of *H. sapiens* and *D. melanogaster*, and *A. thaliana* and *C. elegans*, we observe different levels of conservation for positions −1, +5, and +6, and we find differences in the compensatory effects between intron and exon side.

Finally, we employ the MSP principle for the prediction of miRNA target sites. Here, we propose an extended profile HMM with plan9 architecture called conditional profile HMM for modelling the dependence of target sites on the sequence of the miRNA. While the definition of delete and insert states remains unchanged, the emission probabilities of the match states are extended to conditional probability distributions of the nucleotides in target sites given the nucleotides in the associated miRNAs. In contrast to previous approaches, conditional profile HMMs can learn characteristics of miRNA-mRNA binding from data. In this work, we learn the parameters of conditional profile HMMs by the discriminative MSP principle.

As a proof of concept, we learn the parameters of a classifier comprising a conditional profile HMM for the class of target sites and a homogeneous Markov model of order 1 for non-target

sites on the predictions of existing approaches, namely miRanda, TargetScan, RNAhybrid, and DIANA-microT. We observe that the conditional profile HMM adapts well to the characteristics of these approaches. Subsequently, we learn the same classifier on a data set containing experimentally verified target sites, predicted target sites in UTRs of experimentally verified target genes, and predictions of the four existing approaches considered. We use the learned classifier to predict putative target sites of three miRNAs in UTRs with annotated, verified target sites. We find that the classifier employing the conditional profile HMM discovers verified target sites and predicts additional putative target sites. An alignment of the predicted target sites to the corresponding miRNA shows that these are in accordance with known characteristics of miRNA-mRNA binding.

Summarizing the above, learning the parameters of appropriate models by the discriminative MSP principle improves the prediction of transcription factor binding sites, donor splice sites and nucleosome positions, and the de-novo discovery of single motifs and cis-regulatory modules. We also achieve promising results for the prediction of miRNA target sites. Hence, our findings suggest that the discriminative MSP principle is of general value for statistical analysis and classification of biological sequences.

Although we establish enhanced approaches for the prediction and understanding of many biological processes in this work, the connections between the different applications are not elaborated in this work. On the one hand, an additional study connecting all approaches would have been beyond the scope of this work. On the other hand, the sources of data and the organisms for which these data have been collected are too inhomogeneous to be promising for a unified study. For instance, genome-wide data about nucleosome positioning is currently available only for yeast, where the number of spliced genes is fairly low and transcriptional regulation is by far less complex than in metazoans or plants. And up to now, proteom-wide effects of miRNAs have been studied only for a limited number of over-expressed miRNAs in a special human cell line.

However, it can be expected that the amount of available data will greatly increase within the next years due to improved and affordable experimental techniques like parallel sequencing. In a scenario, where large-scale data about nucleosome positioning, and data about mRNA levels and protein levels under the condition of interest are available, a unified approach could provide a deeper insight into the interaction between the different mechanisms that determine the product of a gene and the rate of its production. Accurate predictions of nucleosome-free regions could guide the de-novo discovery of cis-regulatory modules that up- or down-regulate a set of genes according to mRNA levels, e.g. by an informative and sequence-dependent position distribution. Once these modules are discovered, a genome-wide prediction could identify additional genes that are putatively regulated by the same set of transcription factors. If mRNA and protein levels are measured under the same condition, these could be related to translational repression by miRNAs, and potential feedback, e.g. due to repression of the translation of transcription factors, could be detected. The combination of these information could then result in a deeper understanding of the origins of observed phenotypes.

Since the methods proposed in this work have been successfully applied to each of the individual tasks, we might anticipate that these might also be useful in a unified approach that integrates data from all levels of gene regulation.

Bibliography

Ao, W., Gaudet, J., Kent, W. J., Muttumu, S., and Mango, S. E. (2004). Environmentally induced foregut remodeling by PHA-4/FoxA and DAF-12/NHR. *Science*, 305(5691):1743–1746.

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

Bailey, T. L. and Elkan, C. (1995). Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21(1):51–80.

Bailey, T. L., Williams, N., Misleh, C., and Li, W. W. (2006). MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Research*, 34(suppl_2):W369–373.

Barash, Y., Elidan, G., Friedman, N., and Kaplan, T. (2003). Modelling dependencies in protein-DNA binding sites. In *RECOMB '03: Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 28–37, New York, NY, USA. ACM Press.

Bartel, D. P. (2004). MicroRNAs: Genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281 – 297.

Bates, D. L., Chen, Y., Kim, G., Guo, L., and Chen, L. (2008). Crystal structures of multiple GATA zinc fingers bound to DNA reveal new insights into DNA recognition and self-association by GATA. *Journal of Molecular Biology*, 381(5):1292 – 1306.

Ben-Gal, I., Shani, A., Gohr, A., Grau, J., Arviv, S., Shmilovici, A., Posch, S., and Grosse, I. (2005). Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics*, 21(11):2657–2666.

Benotmane, A. M., Hoylaerts, M. F., Collen, D., and Belayew, A. (1997). Nonisotopic quantitative analysis of protein-DNA interactions at equilibrium. *Analytical Biochemistry*, 250:181–185.

Berman, B. P., Nibu, Y., Pfeiffer, B. D., Tomancak, P., Celniker, S. E., Levine, M., Rubin, G. M., and Eisen, M. B. (2002). Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome. *Proceedings of the National Academy of Sciences of the United States of America*, 99(2):757–762.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 1st edition.

Black, D. L. (2003). Mechanisms of alternative pre-messenger RNA splicing. *Annual Review of Biochemistry*, 72(1):291–336.

Blomquist, P., Belikov, S., and Wrange, O. (1999). Increased nuclear factor 1 binding to its nucleosomal site mediated by sequence-dependent DNA structure. *Nucleic Acids Research*, 27(2):517–525.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., and Schneider, M. (2003). The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Research*, 31(1):365–370.

Boross, G., Orosz, K., and Farkas, I. J. (2009). Human microRNAs co-silence in well-separated groups and have different predicted essentialities. *Bioinformatics*, 25(8):1063–1069.

Brennecke, J., Stark, A., Russell, R. B., and Cohen, S. M. (2005). Principles of microRNA–target recognition. *PLoS Biology*, 3(3).

Brow, D. A. (2002). Allosteric cascade of spliceosome activation. *Annual Review of Genetics*, 36(1):333–360.

Brow, D. A. and Guthrie, C. (1988). Spliceosomal RNA U6 is remarkably conserved from yeast to mammals. *Nature*, 334(6179):213–218.

Brukner, I., Sanchez, R., Suck, D., and Pongor, S. (1995). Sequence-dependent bending propensity of DNA as revealed by DNase I: parameters for trinucleotides. *The EMBO Journal*, 14(8):1812–1818.

Bühlmann, P. and Wyner, A. J. (1999). Variable length Markov chains. *Annals of Statistics*, 27(2):480–513.

Buntine, W. L. (1991). Theory refinement of Bayesian networks. In *Uncertainty in Artificial Intelligence*, pages 52–62. Morgan Kaufmann.

Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268(1):78–94.

Burge, C. B. (1998). Modelling dependencies in pre-mRNA splicing signals. In Salzberg, S. L., Searls, D. B., and Kasif, S., editors, *Computational Methods in Molecular Biology*, volume 32 of *New comprehensive biochemistry*, chapter 8, pages 129–164. Elsevier.

Burset, M., Seledtsov, I. A., and Solovyev, V. V. (2001). SpliceDB: database of canonical and non-canonical mammalian splice sites. *Nucleic Acids Research*, 29(1):255–259.

Carmel, I., Tal, S., Vig, I., and Ast, G. (2004). Comparative analysis detects dependencies among the 5' splice-site positions. *RNA*, 10(5):828–840.

Cawley, G., Talbot, N., and Girolami, M. (2007). Sparse multinomial logistic regression via Bayesian L1 regularisation. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA.

Cerquides, J. and de Mántaras, R. L. (2005). Robust Bayesian linear classifier ensembles. In *Proceedings of the 16th European Conference on Machine Learning*, volume 3720 of *Lecture Notes in Computer Science*, pages 72–83. Springer.

Chekmenev, D. S., Haid, C., and Kel, A. E. (2005). P-Match: transcription factor binding site search by combining patterns and weight matrices. *Nucleic Acids Research*, 33(suppl_2):W432–437.

Chen, S. and Rosenfeld, R. (1999). A Gaussian prior for smoothing maximum entropy models. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Chikhirzhina, G., Al-Shekhadat, R., and Chikhirzhina, E. (2008). Transcription factors of the NF1 family: Role in chromatin remodeling. *Molecular Biology*, 42(3):342–356.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Creanza, T., Horner, D., D'Addabbo, A., Maglietta, R., Mignone, F., Ancona, N., and Pesole, G. (2009). Statistical assessment of discriminative features for protein-coding and non coding cross-species conserved sequence elements. *BMC Bioinformatics*, 10(Suppl 6):S2.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 233–240, New York, NY, USA. ACM.

de Hoon, M. J. L., Makita, Y., Imoto, S., Kobayashi, K., Ogasawara, N., Nakai, K., and Miyano, S. (2004). Predicting gene regulation by sigma factors in Bacillus subtilis from genome-wide data. *Bioinformatics*, 20(suppl_1):i101–108.

DeGroot, M. H. (2004). *Optimal Statistical Decisions (Wiley Classics Library)*. Wiley-Interscience, John Wiley & Sons, Inc., Hoboken, New Jersey.

Eddy, S. R. (1996). Hidden Markov models. *Current Opinion in Structural Biology*, 6(3):361–365.

Ellrott, K., Yang, C., Sladek, F. M., and Jiang, T. (2002). Identifying transcription factor binding sites through Markov chain optimization. *Bioinformatics*, 18(suppl_2):S100–109.

Enright, A., John, B., Gaul, U., Tuschl, T., Sander, C., and Marks, D. (2003). MicroRNA targets in Drosophila. *Genome Biology*, 5(1):R1.

Evans, R. (1988). The steroid and thyroid hormone receptor superfamily. *Science*, 240(4854):889–895.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Field, Y., Kaplan, N., Fondufe-Mittendorf, Y., Moore, I. K., Sharon, E., Lubling, Y., Widom, J., and Segal, E. (2008). Distinct modes of regulation by chromatin encoded through nucleosome positioning signals. *PLoS Comput Biol*, 4(11):e1000216.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann.

Fried, M. and Crothers, D. M. (1981). Equilibria and kinetics of lac repressor-operator interactions by polyacrylamide gel electrophoresis. *Nucleic Acids Research*, 9(23):6505–6525.

Fu, X., Gu, J., Chan, C., Lin, M., Yew, D., Kung, H., and Lai, L. (2007). Possible rules in microRNA target recognition. *International Journal of Integrative Biology*, 1(3):165–171.

Galas, D. J. and Schmitz, A. (1978). DNase footprinting: a simple method for the detection of protein-DNA binding specificity. *Nucleic Acids Research*, 5(9):3157–3170.

Genkin, A., Lewis, D. D., and Madigan, D. (2005). Sparse logistic regression for text categorization. Project Report.

Gershenzon, N. I. and Ioshikhes, I. P. (2005). Synergy of human Pol II core promoter elements revealed by statistical sequence analysis. *Bioinformatics*, 21(8):1295–1300.

Ghosh, Z., Chakrabarti, J., and Mallick, B. (2007). miRNomics–the bioinformatics of microRNA genes. *Biochemical and Biophysical Research Communications*, 363(1):6–11.

Gorin, A. A., Zhurkin, V. B., and K., W. (1995). B-DNA twisting correlates with base-pair morphology. *Journal of Molecular Biology*, 247(1):34–48.

Grau, J., Keilwagen, J., Gohr, A., Grosse, I., and Posch, S. (2008). A Java framework for statistical analysis and classification of biological sequences. http://www.jstacs.de.

Grau, J., Keilwagen, J., Grosse, I., and Posch, S. (2007a). On the relevance of model orders to discriminative learning of markov models. In Hinneburg, A., editor, *LWA: Lernen – Wissen – Adaption*, pages 61–66.

Grau, J., Keilwagen, J., Kel, A., Grosse, I., and Posch, S. (2007b). Supervised posteriors for DNA-motif classification. In Falter, C., Schliep, A., Selbig, J., Vingron, M., and Walther, D., editors, *German Conference on Bioinformatics*, volume 115 of *Lecture Notes in Informatics (LNI) - Proceedings*, Bonn. Gesellschaft für Informatik.

Greiner, R., Su, X., Shen, B., and Zhou, W. (2005). Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Machine Learning Journal*, 59(3):297–322.

Greiner, R. and Zhou, W. (2001). Discriminant parameter learning of belief net classifiers. Technical report, Department of Computing Science, University of Alberta, Canada.

Griffiths-Jones, S., Grocock, R. J., van Dongen, S., Bateman, A., and Enright, A. J. (2006). miRBase: microRNA sequences, targets and gene nomenclature. *Nucleic Acids Research*, 34(suppl_1):D140–144.

Griffiths-Jones, S., Saini, H. K., van Dongen, S., and Enright, A. J. (2008). miRBase: tools for microRNA genomics. *Nucleic Acids Research*, 36(suppl_1):D154–158.

Grosse, I., Bernaola-Galván, P., Carpena, P., Román-Roldán, R., Oliver, J., and Stanley, H. E. (2002). Analysis of symbolic sequences using the Jensen-Shannon divergence. *Physical Review E*, 65(4):041905.

Grossman, D. and Domingos, P. (2004). Learning Bayesian network classifiers by maximizing conditional likelihood. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, pages 361–368, New York, NY, USA. ACM Press.

Grünwald, P., Kontkanen, P., Myllymäki, P., Roos, T., Tirri, H., and Wettig, H. (2002). Supervised posterior distributions. Presented at the Seventh Valencia International Meeting on Bayesian Statistics.

GuhaThakurta, D. and Stormo, G. D. (2001). Identifying target sites for cooperatively binding factors. *Bioinformatics*, 17(7):608–621.

Gunewardena, S. and Zhang, Z. (2008). A hybrid model for robust detection of transcription factor binding sites. *Bioinformatics*, 24(4):484–491.

Guo, Y., Wilkinson, D., and Schuurmans, D. (2005). Maximum margin Bayesian networks. In *21st Conference on Uncertainty in Artificial Intelligence*, pages 233–242. American Association for Artificial Intelligence.

Gupta, M. and Liu, J. S. (2005). De novo cis-regulatory module elicitation for eukaryotic genomes. *Proceedings of the National Academy of Sciences of the United States of America*, 102(20):7079–7084.

Gupta, S., Dennis, J., Thurman, R. E., Kingston, R., Stamatoyannopoulos, J. A., and Noble, W. S. (2008). Predicting human nucleosome occupancy from primary sequence. *PLoS Comput Biol*, 4(8):e1000134.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. In *Machine Learning*, pages 197–243.

Heckerman, D. and Meek, C. (1997). Models and selection criteria for regression and classification. Technical Report MSR-TR-97-08, Microsoft Research, Advanced Technology Division.

Hellman, L. M. and Fried, M. G. (2007). Electrophoretic mobility shift assay (EMSA) for detecting protein-nucleic acid interactions. *Nature Protocols*, 2(8):1849–1861.

Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M., and Schuster, P. (1994). Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie / Chemical Monthly*, 125(2):167–188.

Holland, R. C. G., Down, T. A., Pocock, M., Prlic, A., Huen, D., James, K., Foisy, S., Dräger, A., Yates, A., Heuer, M., and Schreiber, M. J. (2008). BioJava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–2097.

Hsu, S.-D., Chu, C.-H., Tsou, A.-P., Chen, S.-J., Chen, H.-C., Hsu, P. W.-C., Wong, Y.-H., Chen, Y.-H., Chen, G.-H., and Huang, H.-D. (2008). miRNAMap 2.0: genomic maps of microRNAs in metazoan genomes. *Nucleic Acids Research*, 36(suppl_1):D165–169.

Ioshikhes, I. P., Albert, I., Zanton, S. J., and Pugh, B. F. (2006). Nucleosome positions predicted through comparative genomics. *Nature Genetics*, 38(10):1210–1215.

Jeziorska, D. M., Jordan, K. W., and Vance, K. W. (2009). A systems biology approach to understanding cis-regulatory module function. *Seminars in Cell & Developmental Biology*, 20(7):856 – 862.

Jiang, B., Zhang, M. Q., and Zhang, X. (2007). OSCAR: One-class SVM for accurate recognition of cis-elements. *Bioinformatics*, 23(21):2823–2828.

Jing, Y., Pavlović, V., and Rehg, J. M. (2005). Efficient discriminative learning of Bayesian network classifier via boosted augmented naive Bayes. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 369–376, New York, NY, USA. ACM Press.

John, B., Enright, A. J., Aravin, A., Tuschl, T., Sander, C., and Marks, D. S. (2004). Human microRNA targets. *PLoS Biology*, 2(11):e363.

Johnson, D. S., Mortazavi, A., Myers, R. M., and Wold, B. (2007). Genome-wide mapping of in vivo protein-DNA interactions. *Science*, 316(5830):1497–1502.

Kaplan, T., Friedman, N., and Margalit, H. (2005). Ab initio prediction of transcription factor targets using structural knowledge. *PLoS Computational Biology*, 1(1):e1.

Karas, H., Knuppel, R., Schulz, W., Sklenar, H., and Wingender, E. (1996). Combining structural analysis of DNA with search routines for the detection of transcription regulatory elements. *Computer Applications in the Biosciences*, 12(5):441–446.

Karin, M., Liu, Z., and Zandi, E. (1997). AP-1 function and regulation. *Current Opinion in Cell Biology*, 9(2):240 – 246.

Keilwagen, J., Grau, J., Paponov, I. A., Posch, S., Strickert, M., and Grosse, I. (2010a). De-novo discovery of differentially abundant transcription factor binding sites including their positional preference. Under review.

Keilwagen, J., Grau, J., Posch, S., and Grosse, I. (2007). Recognition of splice sites using maximum conditional likelihood. In Hinneburg, A., editor, *LWA: Lernen – Wissen – Adaption*, pages 67–72.

Keilwagen, J., Grau, J., Posch, S., and Grosse, I. (2010b). Apples and oranges: avoiding different priors in Bayesian DNA sequence analysis. *BMC Bioinformatics*, 11(1):149.

Keilwagen, J., Grau, J., Posch, S., Strickert, M., and Grosse, I. (2010c). Unifying generative and discriminative learning principles. *BMC Bioinformatics*, 11(1):98.

Kel, A. E., Gössling, E., Reuter, I., Cheremushkin, E., Kel-Margoulis, O. V., and Wingender, E. (2003). Match: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Research*, 31(13):3576–3579.

Keles, S., van der Laan, M. J., Dudoit, S., Xing, B., and Eisen, M. B. (2003). Supervised detection of regulatory motifs in DNA sequences. *Statistical Applications in Genetics and Molecular Biology*, 2(1):5.

Kertesz, M., Iovino, N., Unnerstall, U., Gaul, U., and Segal, E. (2007). The role of site accessibility in microRNA target recognition. *Nature Genetics*, 39(10):1278–1284.

Kim, M. and Pavlovic, V. (2005). Discriminative learning of mixture of Bayesian network classifiers for sequence classification. Technical Report RU-DCS-TR588, Dept. of Computer Science, Rutgers University.

Kim, N.-K., Tharakaraman, K., Marino-Ramirez, L., and Spouge, J. (2008). Finding sequence motifs with Bayesian models incorporating positional information: an application to transcription factor binding sites. *BMC Bioinformatics*, 9(1):262.

King, O. D. and Roth, F. P. (2003). A non-parametric model for transcription factor binding sites. *Nucleic Acids Research*, 31(19):e116–.

Ko, L. J. and Engel, J. D. (1993). DNA-binding specificities of the GATA transcription factor family. *Molecular and Cellular Biology*, 13(7):4011–4022.

Krek, A., Grun, D., Poy, M. N., Wolf, R., Rosenberg, L., Epstein, E. J., MacMenamin, P., da Piedade, I., Gunsalus, K. C., Stoffel, M., and Rajewsky, N. (2005). Combinatorial microRNA target predictions. *Nature Genetics*, 37(5):495–500.

Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D. (1994). Hidden markov models in computational biology : Applications to protein modeling. *Journal of Molecular Biology*, 235(5):1501 – 1531.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86.

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

Lawrence, C. E. and Reilly, A. A. (1990). An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function, and Genetics*, 7(1):41–51.

Lee, W., Tillo, D., Bray, N., Morse, R. H., Davis, R. W., Hughes, T. R., and Nislow, C. (2007). A high-resolution atlas of nucleosome occupancy in yeast. *Nature Genetics*, 39(10):1235–1244.

Levine, M. and Tjian, R. (2003). Transcription regulation and animal diversity. *Nature*, 424(6945):147–151.

Lewis, B. P., Burge, C. B., and Bartel, D. P. (2005). Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *Cell*, 120(1):15 – 20.

Lewis, B. P., Shih, I., Jones-Rhoades, M. W., Bartel, D. P., and Burge, C. B. (2003). Prediction of mammalian microRNA targets. *Cell*, 115(7):787 – 798.

Liao, J. and Chin, K.-V. (2007). Logistic regression for disease classification using microarray data: model selection in a large p and small n case. *Bioinformatics*, 23(15):1945–1951.

Lin, J. (2002). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Lisser, S. and Margalit, H. (1993). Compilation of E. coli mRNA promoter sequences. *Nucleic Acids Research*, 21(7):1507–1516.

Liu, H., Wu, J., Xie, J., Yang, X., Lu, Z., and Sun, X. (2008). Characteristics of Nucleosome Core DNA and Their Applications in Predicting Nucleosome Positions. *Biophysical Journal*, 94(12):4597–4604.

Lo, P., Roy, D., and Mount, S. M. (1994). Suppressor U1 snRNAs in Drosophila. *Genetics*, 138(2):365–378.

Lockhart, D. J. and Winzeler, E. A. (2000). Genomics, gene expression and DNA arrays. *Nature*, 405(6788):827–836.

Loo, P. V. and Marynen, P. (2009). Computational methods for the detection of cis-regulatory modules. *Briefings in Bioinformatics*, 10(5):509–524.

Lubliner, S. and Segal, E. (2009). Modeling interactions between adjacent nucleosomes improves genome-wide predictions of nucleosome occupancy. *Bioinformatics*, 25(12):i348–355.

Madigan, D., Genkin, A., Lewis, D. D., Argamon, S., Fradkin, D., and Ye, L. (2005). Author identification on the large scale. In *Joint Annual Meeting of the Interface and the Classification Society of North America*.

Man, O. and Pilpel, Y. (2007). Differential translation efficiency of orthologous genes is involved in phenotypic divergence of yeast species. *Nature Genetics*, 39(3):415–421.

Maragkakis, M., Reczko, M., Simossis, V. A., Alexiou, P., Papadopoulos, G. L., Dalamagas, T., Giannopoulos, G., Goumas, G., Koukis, E., Kourtis, K., Vergoulis, T., Koziris, N., Sellis, T., Tsanakas, P., and Hatzigeorgiou, A. G. (2009). DIANA-microT web server: elucidating microRNA functions through target prediction. *Nucleic Acids Research*, 37(suppl_2):W273–276.

Marin, M., Karis, A., Visser, P., Grosveld, F., and Philipsen, S. (1997). Transcription factor Sp1 is essential for early embryonic development but dispensable for cell growth and differentiation. *Cell*, 89(4):619–628.

Mathews, D. H., Sabina, J., Zuker, M., and Turner, D. H. (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288(5):911 – 940.

Matys, V., Kel-Margoulis, O. V., Fricke, E., Liebich, I., Land, S., Barre-Dirrie, A., Reuter, I., Chekmenev, D., Krull, M., Hornischer, K., Voss, N., Stegmaier, P., Lewicki-Potapov, B., Saxel, H., Kel, A. E., and Wingender, E. (2006). TRANSFAC(R) and its module TRANSCompel(R): transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, 34(suppl_1):D108–110.

Meila-Predoviciu, M. (1999). *Learning with Mixtures of Trees*. PhD thesis, Massachusetts Institute of Technology.

Meinicke, P., Tech, M., Morgenstern, B., and Merkl, R. (2004). Oligo kernels for datamining on biological sequences: a case study on prokaryotic translation initiation sites. *BMC Bioinformatics*, 5(1):169.

Miele, V., Vaillant, C., d'Aubenton Carafa, Y., Thermes, C., and Grange, T. (2008). DNA physical properties determine nucleosome occupancy from yeast to fly. *Nucleic Acids Research*, 36(11):3746–3756.

Miller, M., Shuman, J. D., Sebastian, T., Dauter, Z., and Johnson, P. F. (2003). Structural Basis for DNA Recognition by the Basic Region Leucine Zipper Transcription Factor CCAAT/Enhancer-binding Protein α. *Journal of Biological Chemistry*, 278(17):15178–15184.

Mitchell, P. J., Carothers, A. M., Han, J. H., Harding, J. D., Kas, E., Venolia, L., and Chasin, L. A. (1986). Multiple transcription start sites, DNase I-hypersensitive sites, and an opposite-strand exon in the 5' region of the CHO dhfr gene. *Molecular and Cellular Biology*, 6(2):425–440.

Mizukami, Y., Huang, H., Tudor, M., Hu, Y., and Ma, H. (1996). Functional Domains of the Floral Regulator AGAMOUS: Characterization of the DNA Binding Domain and Analysis of Dominant Negative Mutations. *Plant Cell*, 8(5):831–845.

Mönke, G., Altschmied, L., Tewes, A., Reidt, W., Mock, H.-P., Bäumlein, H., and Conrad, U. (2004). Seed-specific transcription factors ABI3 and FUS3: molecular interaction with DNA. *Planta*, 219(1):158–166.

Morozov, A. V., Fortney, K., Gaykalova, D. A., Studitsky, V. M., Widom, J., and Siggia, E. D. (2009). Using DNA mechanics to predict in vitro nucleosome positions and formation energies. *Nucleic Acids Research*, 37(14):4707–4722.

Narlikar, L., Gordân, R., and Hartemink, A. J. (2007). A nucleosome-guided map of transcription factor binding sites in yeast. *PLoS Computational Biology*, 3(11):e215.

Ng, A. and Jordan, M. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Dietterich, T., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems*, volume 14, pages 605–610. MIT Press, Cambridge, MA.

Nicorici, D. and Astola, J. (2004). Segmentation of DNA into coding and noncoding regions based on recursive entropic segmentation and stop-codon statistics. *EURASIP J. Appl. Signal Process.*, 2004:81–91.

Olson, W. K., Gorin, A. A., Lu, X.-J., Hock, L. M., and Zhurkin, V. B. (1998). DNA sequence-dependent deformability deduced from protein–DNA crystal complexes. *Proceedings of the National Academy of Sciences of the United States of America*, 95(19):11163–11168.

Ong, M. S., Richmond, T. J., and Davey, C. A. (2007). DNA stretching and extreme kinking in the nucleosome core. *Journal of Molecular Biology*, 368(4):1067 – 1074.

Ornstein, R. L., Rein, R., Breen, D. L., and Macelroy, R. D. (1978). An optimized potential function for the calculation of nucleic acid interaction energies – I. Base stacking. *Biopolymers*, 17(10):2341–2360.

Papadopoulos, G. L., Reczko, M., Simossis, V. A., Sethupathy, P., and Hatzigeorgiou, A. G. (2009). The database of experimentally supported targets: a functional update of TarBase. *Nucleic Acids Research*, 37(suppl_1):D155–158.

Pape, U. J., Klein, H., and Vingron, M. (2009). Statistical detection of cooperative transcription factors with similarity adjustment. *Bioinformatics*, 25(16):2103–2109.

Paponov, I. A., Paponov, M., Teale, W., Menges, M., Chakrabortee, S., Murray, J. A. H., and Palme, K. (2008). Comprehensive transcriptome analysis of auxin responses in Arabidopsis. *Molecular Plant*, 1(2):321–337.

Pavesi, G., Mauri, G., and Pesole, G. (2001). An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics*, 17:S207–214.

Peckham, H. E., Thurman, R. E., Fu, Y., Stamatoyannopoulos, J. A., Noble, W. S., Struhl, K., and Weng, Z. (2007). Nucleosome positioning signals in genomic DNA. *Genome Research*, 17(8):1170–1177.

Posch, S., Grau, J., Gohr, A., Ben-Gal, I. E., Kel, A. E., and Grosse, I. (2007). Recognition of cis-regulatory elements with Vombat. *Journal Bioinformatics and Computational Biology*, 5(2b):561–577.

Ramji, D. P. and Foka, P. (2002). CCAAT/enhancer-binding proteins: structure, function and regulation. *Biochemical Journal*, 365(3):561–575.

Rani, T. S., Bhavani, S. D., and Bapi, R. S. (2007). Analysis of E. coli promoter recognition problem in dinucleotide feature space. *Bioinformatics*, 23(5):582–588.

Redhead, E. and Bailey, T. (2007). Discriminative motif discovery in DNA and protein sequences using the DEME algorithm. *BMC Bioinformatics*, 8(1):385.

Rehmsmeier, M., Steffen, P., Höchsmann, M., and Giegerich, R. (2004). Fast and effective prediction of microRNA/target duplexes. *RNA*, 10(10):1507–1517.

Rein, R. (1973). On physical properties and interactions of polyatomic molecules: With application to molecular recognition in biology. volume 7 of *Advances in Quantum Chemistry*, pages 335 – 396. Academic Press.

Reinhart, B. J., Weinstein, E. G., Rhoades, M. W., Bartel, B., and Bartel, D. P. (2002). MicroRNAs in plants. *Genes & Development*, 16(13):1616–1626.

Rhoades, M. W., Reinhart, B. J., Lim, L. P., Burge, C. B., Bartel, B., and Bartel, D. P. (2002). Prediction of plant microRNA targets. *Cell*, 110(4):513–520.

Richmond, T. J. and Davey, C. A. (2003). The structure of DNA in the nucleosome core. *Nature*, 423(6936):145–150.

Rissanen, J. (1983). A universal data compression system. *IEEE Transactions on Information Theory*, 29(5):656–664.

Roni, V., Carpio, R., and Wissinger, B. (2007). Mapping of transcription start sites of human retina expressed genes. *BMC Genomics*, 8(1):42.

Roos, T., Wettig, H., Grunwald, P., Myllymaki, P., and Tirri, H. (2005). On discriminative Bayesian network classifiers and logistic regression. *Machine Learning*, 59(3):267–296.

Salgado, H., Santos-Zavaleta, A., Gama-Castro, S., Millan-Zarate, D., Blattner, F. R., and Collado-Vides, J. (2000). RegulonDB (version 3.0): transcriptional regulation and operon organization in Escherichia coli K-12. *Nucleic Acids Research*, 28(1):65–67.

Salzberg, S. L. (1997). A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Computer Applications in the Biosciences*, 13(4):365–376.

Sandelin, A., Alkema, W., Engström, P., Wasserman, W. W., and Lenhard, B. (2004). Jaspar: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research*, 32(Database issue):D91–D94.

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

Schneider, T. D. and Stephens, R. M. (1990). Sequence logos: A new way to display consensus sequences. *Nucleic Acids Research*, 18(20):6097–6100.

Schultheiss, S. J., Busch, W., Lohmann, J. U., Kohlbacher, O., and Rätsch, G. (2009). KIRMES: kernel-based identification of regulatory modules in euchromatic sequences. *Bioinformatics*, 25(16):2126–2133.

Schweikert, G., Zien, A., Zeller, G., Behr, J., Dieterich, C., Ong, C. S., Philips, P., De Bona, F., Hartmann, L., Bohlen, A., Krüger, N., Sonnenburg, S., and Rätsch, G. (2009). mgene: Accurate svm-based gene finding with an application to nematode genomes. *Genome Research*, 19(11):2133–2143.

Segal, E., Fondufe-Mittendorf, Y., Chen, L., Thåström, A., Field, Y., Moore, I. K., Wang, J.-P. Z., and Widom, J. (2006). A genomic code for nucleosome positioning. *Nature*, 442(7104):772–778.

Segal, E. and Widom, J. (2009). Poly(dA:dT) tracts: major determinants of nucleosome organization. *Current Opinion in Structural Biology*, 19(1):65–71.

Selbach, M., Schwanhausser, B., Thierfelder, N., Fang, Z., Khanin, R., and Rajewsky, N. (2008). Widespread changes in protein synthesis induced by microRNAs. *Nature*, 455(7209):58–63.

Sivolob, A. V. and Khrapunov, S. N. (1995). Translational positioning of nucleosomes on DNA: The role of sequence-dependent isotropic DNA bending stiffness. *Journal of Molecular Biology*, 247(5):918–931.

Smith, A. D., Sumazin, P., and Zhang, M. Q. (2005). Identifying tissue-selective transcription factor binding sites in vertebrate promoters. *Proceedings of the National Academy of Sciences of the United States of America*, 102(5):1560–1565.

Smola, A. J. and Schölkopf, B. (1998). On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 22:211–231.

Sonnenburg, S., Schweikert, G., Philips, P., Behr, J., and Rätsch, G. (2007). Accurate splice site prediction using support vector machines. *BMC Bioinformatics*, 8(Suppl 10):S7.

Sonnenburg, S., Zien, A., Philips, P., and Rätsch, G. (2008). POIMs: positional oligomer importance matrices – understanding support vector machine-based signal detectors. *Bioinformatics*, 24(13):i6–14.

Sonnenburg, S., Zien, A., and Ratsch, G. (2006). ARTS: accurate recognition of transcription starts in human. *Bioinformatics*, 22(14):e472–480.

Staden, R. (1984). Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Research*, 12:505–519.

Stark, A., Brennecke, J., Russell, R. B., and Cohen, S. M. (2003). Identification of Drosophila microRNA targets. *PLoS Biology*, 1(3):e60.

Stepanova, M., Tiazhelova, T., Skoblov, M., and Baranova, A. (2005). A comparative analysis of relative occurrence of transcription factor binding sites in vertebrate genomes and gene promoter areas. *Bioinformatics*, 21(9):1789–1796.

Stormo, G. D., Schneider, T. D., Gold, L. M., and Ehrenfeucht, A. (1982). Use of the 'perceptron' algorithm to distinguish translational initiation sites. *Nucleic Acids Research*, 10(9):2997–3010.

Sugimoto, N., Nakano, S., Yoneyama, M., and Honda, K. (1996). Improved thermodynamic parameters and helix initiation factor to predict stability of DNA duplexes. *Nucleic Acids Research*, 24(22):4501–4505.

Sun, L. V., Chen, L., Greil, F., Negre, N., Li, T.-R., Cavalli, G., Zhao, H., Steensel, B. V., and White, K. P. (2003). Protein-DNA interaction mapping using genomic tiling path microarrays in Drosophila. *Proceedings of the National Academy of Sciences of the United States of America*, 100(16):9428–9433.

Suter, B., Schnappauf, G., and Thoma, F. (2000). Poly(dA·dT) sequences exist as rigid DNA structures in nucleosome-free yeast promoters in vivo. *Nucleic Acids Research*, 28(21):4083–4089.

Suzuki, M., Yagi, N., and Finch, J. T. (1996). Role of base-backbone and base-base interactions in alternating DNA conformations. *FEBS Letters*, 379(2):148 – 152.

Swarbreck, D., Wilks, C., Lamesch, P., Berardini, T. Z., Garcia-Hernandez, M., Foerster, H., Li, D., Meyer, T., Muller, R., Ploetz, L., Radenbaugh, A., Singh, S., Swing, V., Tissier, C., Zhang, P., and Huala, E. (2008). The Arabidopsis Information Resource (TAIR): gene structure and function annotation. *Nucleic Acids Research*, 36(Database issue):D1009–D1014.

The ENCODE Project Consortium (2007). Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, 447(7146):799–816.

Thompson, W., Rouchka, E. C., and Lawrence, C. E. (2003). Gibbs Recursive Sampler: finding transcription factor binding sites. *Nucleic Acids Research*, 31(13):3580–3585.

Thompson, W. A., Newberg, L. A., Conlan, S., McCue, L. A., and Lawrence, C. E. (2007). The gibbs centroid sampler. *Nucleic Acids Research*, 35(suppl_2):W232–237.

Tibshirani, R. (1994). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288.

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

Ucar, D., Beyer, A., Parthasarathy, S., and Workman, C. T. (2009). Predicting functionality of protein-DNA interactions by integrating diverse evidence. *Bioinformatics*, 25(12):i137–144.

Valen, E., Sandelin, A., Winther, O., and Krogh, A. (2009). Discovery of regulatory elements is improved by a discriminatory approach. *PLoS Computational Biology*, 5(11):e1000562.

Wallach, H. M. (2004). Conditional random fields: An introduction. Technical Report Technical Report MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania.

Wang, J.-P., Fondufe-Mittendorf, Y., Xi, L., Tsai, G.-F., Segal, E., and Widom, J. (2008). Preferentially quantized linker DNA lengths in Saccharomyces cerevisiae. *PLoS Computational Biology*, 4(9):e1000175.

Wang, Y., Amirhaeri, S., Kang, S., Wells, R., and Griffith, J. (1994). Preferential nucleosome assembly at DNA triplet repeats from the myotonic dystrophy gene. *Science*, 265(5172):669–671.

Weindl, J., Hanus, P., Dawy, Z., Zech, J., Hagenauer, J., and Mueller, J. C. (2007). Modeling DNA-binding of Escherichia coli σ^{70} exhibits a characteristic energy landscape around strong promoters. *Nucleic Acids Research*, 35(20):7003–7010.

Wettig, H., Grünwald, P., Roos, T., Myllymäki, P., and Tirri, H. (2002). On supervised learning of Bayesian network parameters. Technical Report HIIT 2002-1, Helsinki Institute for Information Technology.

Wettig, H., Grünwald, P., Roos, T., Myllymäki, P., and Tirri, H. (2003). When discriminative learning of bayesian network parameters is easy. In Gottlog, G. and Walsh, T., editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann.

Wingender, E., Dietze, P., Karas, H., and Knuppel, R. (1996). Transfac: A database on transcription factors and their DNA binding sites. *Nucleic Acids Research*, 24(1):238–241.

Wu, J., Smith, L. T., Plass, C., and Huang, T. H.-M. (2006). ChIP-chip Comes of Age for Genome-wide Functional Analysis. *Cancer Research*, 66(14):6899–6902.

Wuchty, S., Fontana, W., Hofacker, I. L., and Schuster, P. (1998). Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49:145–165.

Xiao, F., Zuo, Z., Cai, G., Kang, S., Gao, X., and Li, T. (2009). miRecords: an integrated resource for microRNA-target interactions. *Nucleic Acids Research*, 37(suppl_1):D105–110.

Yakhnenko, O., Silvescu, A., and Honavar, V. (2005). Discriminatively trained markov model for sequence classification. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 498–505, Washington, DC, USA. IEEE Computer Society.

Yassour, M., Kaplan, T., Jaimovich, A., and Friedman, N. (2008). Nucleosome positioning from tiling microarray data. *Bioinformatics*, 24(13):i139–146.

Yeo, G. and Burge, C. B. (2004). Maximum Entropy Modeling of Short Sequence Motifs with Applications to RNA Splicing Signals. *Journal of Computational Biology*, 11(2-3):377–394.

Yuan, G.-C. and Liu, J. S. (2008). Genomic sequence is highly predictive of local nucleosome depletion. *PLoS Computational Biology*, 4(1):e13.

Yuan, G.-C., Liu, Y.-J., Dion, M. F., Slack, M. D., Wu, L. F., Altschuler, S. J., and Rando, O. J. (2005). Genome-Scale Identification of Nucleosome Positions in S. cerevisiae. *Science*, 309(5734):626–630.

Zhang, M. and Marr, T. (1993). A weight array method for splicing signal analysis. *Computer Applications in the Biosciences*, 9(5):499–509.

Zhao, X., Huang, H., and Speed, T. P. (2005). Finding short DNA motifs using permuted Markov models. *Journal of Computational Biology*, 12(6):894–906.

Zhou, Q. and Wong, W. H. (2004). CisModule: De novo discovery of cis-regulatory modules by hierarchical mixture modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 101(33):12114–12119.

A. Appendix

A.1. Log concavity of conditional likelihood and transformed product-Dirichlet prior

Here, prove log concavity for a function g that includes conditional likelihood for Markov models and decision trees and the transformed product-Dirichlet prior as special cases. To this end, we renumber the parameters $\boldsymbol{\xi}$ using an abstract index n, i.e. we consider a vector of parameters $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)$. We define g as a function of these parameters as

$$g(\boldsymbol{\xi}) = \frac{\exp\left(h(\boldsymbol{\xi})\right)}{\left(\sum_i \exp\left(f_i(\boldsymbol{\xi})\right)\right)^{\gamma}}, \tag{A.1}$$

where h and the f_j are linear functions of the parameters

$$h(\boldsymbol{\xi}) = \sum_n b_n \xi_n, \tag{A.2}$$

$$f_j(\boldsymbol{\xi}) = \sum_n a_{j,n} \xi_n, \tag{A.3}$$

where $b_n \in \mathbb{R}$, and the factors $a_{j,n} \in \mathbb{R}$ may be different for each function f_j.

In case of conditional likelihood given a sequence $\boldsymbol{x}$, the function h in the enumerator corresponds to one function f_k of the functions f_i in the denominator and, hence, $\forall n : b_n = a_{k,n}$. The parameters of Markov models and decision trees that are used depend on the sequence $\boldsymbol{x}$ and the selection of parameters corresponds to setting the values of the b_n and $a_{k,n}$ either to 0 or to 1. The definition of conditional likelihood for inhomogeneous Markov models that corresponds to this functional form is given in equation (3.30) (p 19), and the corresponding definition for decision tree models is given in equation (4.81) (p. 137).

In case of the transformed product-Dirichlet prior, the factors b_n of h correspond to the hyperparameters of the parameters ξ_n. The sum over the different functions f_i in the denominator corresponds to the normalization constant $Z(\boldsymbol{\xi})$, and γ corresponds to the sum over the ESSs for the different classes. The definition of the product-Dirichlet prior in this functional form is given in equation (3.79) (p. 31) for inhomogeneous Markov models and in equation (4.84) (p. 138) for decision tree models.

To prove that g is a log concave function of the parameters, the parameters must be a convex set, which is the case, since $\boldsymbol{\xi} \in \mathbb{R}^N$, and the following inequality (Jensen's inequality) must

hold

$$\log g\left(\alpha\boldsymbol{\xi}^{(1)}+(1-\alpha)\boldsymbol{\xi}^{(2)}\right)\geq\alpha\log g\left(\boldsymbol{\xi}^{(1)}\right)+(1-\alpha)\log g\left(\boldsymbol{\xi}^{(2)}\right), \tag{A.4}$$

where $\alpha\in[0,1]$. Since g has no discontinuities, it is enough to show that this inequality holds for $\alpha=\frac{1}{2}$, i.e.

$$\log g\left(\frac{\boldsymbol{\xi}^{(1)}+\boldsymbol{\xi}^{(2)}}{2}\right)\geq\frac{1}{2}\log g\left(\boldsymbol{\xi}^{(1)}\right)+\frac{1}{2}\log g\left(\boldsymbol{\xi}^{(2)}\right). \tag{A.5}$$

We insert the definition of g into this inequality, and obtain

$$\begin{aligned}&h\left(\frac{\boldsymbol{\xi}^{(1)}+\boldsymbol{\xi}^{(2)}}{2}\right)-\gamma\log\left(\sum_i\exp\left(f_i\left(\frac{\boldsymbol{\xi}^{(1)}+\boldsymbol{\xi}^{(2)}}{2}\right)\right)\right)\overset{!}{\geq} \\ &\frac{1}{2}h\left(\boldsymbol{\xi}^{(1)}\right)+\frac{1}{2}h\left(\boldsymbol{\xi}^{(2)}\right)-\frac{1}{2}\gamma\log\left(\sum_i\exp\left(f_i\left(\boldsymbol{\xi}^{(1)}\right)\right)\right)-\frac{1}{2}\gamma\log\left(\sum_i\exp\left(f_i\left(\boldsymbol{\xi}^{(2)}\right)\right)\right).\end{aligned} \tag{A.6}$$

Since h is a linear function of the parameters, the first term on the left side and the first two terms on the right side of the inequality cancel. Afterwards, we can divide both sides by $\frac{-\gamma}{2}$, which inverts the inequality and results in

$$2\log\left(\sum_i\exp\left(f_i\left(\frac{\boldsymbol{\xi}^{(1)}+\boldsymbol{\xi}^{(2)}}{2}\right)\right)\right)\overset{!}{\leq}\log\left(\sum_i\exp\left(f_i\left(\boldsymbol{\xi}^{(1)}\right)\right)\right)+\log\left(\sum_i\exp\left(f_i\left(\boldsymbol{\xi}^{(2)}\right)\right)\right).$$

We exponentiate both sides, which does not affect the inequality, since the exponential function is strictly monotonic, yielding

$$\left(\sum_i\exp\left(f_i\left(\frac{\boldsymbol{\xi}^{(1)}+\boldsymbol{\xi}^{(2)}}{2}\right)\right)\right)^2\overset{!}{\leq}\left(\sum_i\exp\left(f_i\left(\boldsymbol{\xi}^{(1)}\right)\right)\right)\cdot\left(\sum_i\exp\left(f_i\left(\boldsymbol{\xi}^{(2)}\right)\right)\right), \tag{A.7}$$

which we can re-state as

$$\begin{aligned}&\left[\sum_i\exp\left(f_i\left(\frac{\boldsymbol{\xi}^{(1)}}{2}\right)\right)\exp\left(f_i\left(\frac{\boldsymbol{\xi}^{(2)}}{2}\right)\right)\right]^2\overset{!}{\leq} \\ &\left[\sum_i\exp\left(f_i\left(\frac{\boldsymbol{\xi}^{(1)}}{2}\right)\right)^2\right]\cdot\left[\sum_i\exp\left(f_i\left(\frac{\boldsymbol{\xi}^{(2)}}{2}\right)\right)^2\right].\end{aligned} \tag{A.8}$$

The last inequality holds according to the Cauchy-Schwarz inequality.

Thus, g is a concave function of the parameters $\boldsymbol{\xi}$. This result can be directly transferred to conditional likelihood using Markov models and decision trees, and the associated transformed product-Dirichlet priors.

A.2. Calls of de-novo motif discovery programs

In the following, we specify the calls of the de-novo discovery programs used in section 4.2. The placeholder `<target>` represents the target data set, `<control>` represents the control data set, and `<length>` represents the length of the correct motif.

A-GLAM

unknown length:

```
./aglam -4 500 <target>
```

known length:

```
./aglam -a <length> -b <length> -4 500 <target>
```

Meaning of additional arguments:

- `-4` anchor position of the position distribution
- `-a, -b` minimum and maximum length of the motif

DEME

unknown length:

```
./deme -p <target> -n <control> -w 15
```

known length:

```
./deme -p <target> -n <control> -w <length>
```

Meaning of additional arguments:

- `-w` length of the motif

DME

unknown length:

```
./dme2 -v -n 200 -w 15 -o <outfile> -b <control> <target>
```

known length:

```
./dme2 -v -n 200 -w <length> -o <outfile> -b <control> <target>
```

Meaning of additional arguments:

- `-o` followed by path to the output-file `<outfile>`
- `-v` verbose output
- `-n` number of motifs to produce

Gibbs sampler

unknown length:

```
./Gibbs <target> 15 -n
```

known length:

```
./Gibbs <target> <length> -n
```

Meaning of additional arguments:

- `-n` use nucleic acid alphabet

Centroid Gibbs sampler

unknown length:

```
./Gibbs <target> 15 100 -E 1 -bayes -n
```

known length:

```
./Gibbs <target> <length> 100 -E 1 -bayes -n
```

Meaning of additional arguments:

- `-n` use nucleic acid alphabet
- `-E` maximum sites per sequence, use recursive sampler
- `100` expected total number of binding sites
- `-bayes` use Bayesian sampling

Improbizer

unknown length:

```
./ameme good=<target> bad=<control> numMotifs=1 rcToo=on \
        motifOutput=<target>-motif.txt
./ameme motifMatcher=on seqFile=<target> rcToo=on \
        motifs=<target>-motif.txt hits=<target>-hits.txt
```

known length:

```
./ameme good=<target> bad=<control> numMotifs=1 rcToo=on constrainer=1000 \
        tileSize=<length> motifOutput=<target>-motif.txt
./ameme motifMatcher=on seqFile=<target> rcToo=on \
        motifs=<target>-motif.txt hits=<target>-hits.txt
```

Meaning of additional arguments:

- `numMotifs` number of motifs
- `tileSize` length of motif
- `constrainer=1000` fix motif length
- `motifMatcher=on` predict motif occurrences
- `rcToo=on` search on both strands

MEME

unknown length:

```
./meme -dna -mod zoops -minw 6 -maxw 20 -nmotifs 1 -revcomp -text <target>
```

known length:

```
./meme -dna -mod zoops -w <length> -nmotifs 1 -revcomp -text <target>
```

Meaning of additional arguments:

- `-dna` use nucleic acid alphabet
- `-mod zoops` use ZOOPS model
- `-minw,-maxw` minimum and maximum motif length
- `-w` motif length
- `-nmotifs` number of motifs
- `-revcomp` search on both strands
- `-text` text output instead of HTML

Weeder

unknown length:

```
./weederlauncher.out <target> <organism> large S
```

known length:

```
./weederTFBS.out -f <target> -R 50 -O <organism> -W <length> -e 3 -S -T 10
./adviser.out <target> S
```

Meaning of additional arguments:

- `[-O` ¡organism¿] organism, AT for *A. thaliana*, DM for *D. melanogaster*, HS for *H. sapiens*
- `large` search for motifs of maximum length 12 with at most 4 mismatches
- `S,-S` search on both strands
- `-R 50` percentage of sequence that must contain the motif
- `-W` length of the motif
- `-e` number of allowed mismatches
- `-T` number of reported motifs

CisModule

```
./CisModuleU -i <target> -o <outfile> -K 2 -w 5 -W 15
```

Meaning of additional arguments:

- `-K 2` search for two motifs
- `-w, -W` minimum and maximum length of the motifs
- `-o` followed by path to the output-file `<outfile>`

CoBind

```
./co-bind -p <target> -n <control> -a <alphabetfile> -Z 100 -c 1 -m 20
```

Meaning of additional arguments:

- `-a` path to the file specifying the alphabet `<alphabetfile>`
- `-Z 100` maximum distance between sites, default was 50
- `-c 1` search on both strands
- `-m 20` number of training runs

MoAn

```
moan -D -c <target> <control>
```

Meaning of additional arguments:

- `-D` search on both strands
- `-c` search for two motifs

A.3. Distribution of poly-A or poly-T tracts

In this section, we derive the joint cumulative distribution $P(K \geq k, N \geq n|L)$ of finding at least k poly-A/T tracts of minimum length n in a random sequence of length L. Let p_a be the probability to observe nucleotide $a \in \{A, C, G, T\}$ at a given position of the random sequence.

We derive the joint cumulative distribution in a recursive manner. To this end, we define an additional random variable Z of a state that represents the length of the currently observed poly-A/T tract. If $Z = 0$, the current poly-A/T tract has length 0, if $Z = l, l > 0$ the current poly-A tract has length l, and if $Z = -l, l > 0$ the current poly-T tract has length l.

Admissible transitions between the states are

- a transition from any state to $Z = 0$, i.e. stopping the current poly-A/T tract or proceeding with non-A/T nucleotides,
- a transition from any state $Z < 1$ to $Z = 1$, i.e. starting a poly-A tract,
- a transition from any state $Z > -1$ to $Z = -1$, i.e. starting a poly-T tract,
- a transition from state $Z = i, i > 0$ to $Z = i + 1$, i.e. the elongation of a poly-A tract, and
- a transition from state $Z = -i, i > 0$ to $Z = -i - 1$, i.e. the elongation of a poly-T tract.

In the following, we consider the joint probability $P(Z = z, K \geq k, N \geq n|L = \ell)$ of being in state z and finding at least k poly-A/T tracts of minimum length n in a random sequence of length L.

From any state z, we can go to state $Z = 0$ with transition probability $1 - p_A - p_T$, while the number k of poly-A/T tracts of minimum length n does not change, i.e.

$$P(Z = 0, K \geq k, N \geq n|L = \ell) = P(Z = z, K \geq k, N \geq n|L = \ell - 1)(1 - p_A - p_T). \quad \text{(A.9)}$$

Due to this transition, we emit a symbol different from A or T and elongate the random sequence by one position to length $L = \ell$

For the probability of being in state $Z = 1$ and finding at least k poly-A/T tracts of minimum length n, we must distinguish two cases. If $n = 1$, the transition to $Z = 1$ from some state $Z < 1$ opens a new poly-A tract of the required minimum length, i.e.

$$P(Z = 1, K \geq k, N \geq 1|L = \ell) = P(Z < 1, K \geq k - 1, N \geq 1|L = \ell - 1)p_A. \quad \text{(A.10)}$$

Otherwise, k and n are not changed, and we obtain

$$P(Z = 1, K \geq k, N \geq n|L = \ell) = P(Z < 1, K \geq k, N \geq n|L = \ell - 1)p_A. \quad \text{(A.11)}$$

In analogy, we derive for state $Z = -1$, i.e. opening a new poly-T tract

$$P(Z = -1, K \geq k, N \geq 1|L = \ell) = P(Z > 1, K \geq k - 1, N \geq 1|L = \ell - 1)p_T, \text{ and} \quad \text{(A.12)}$$
$$P(Z = -1, K \geq k, N \geq n|L = \ell) = P(Z > 1, K \geq k, N \geq n|L = \ell - 1)p_T \quad \text{(A.13)}$$

Similarly, we distinguish two cases when elongating an existing poly-A tract, i.e. $Z > 1$. Either, the elongated poly-A tract now reaches the required minimum length n , i.e. $Z = n$, and we obtain

$$P(Z = n, K \geq k, N \geq n|L = \ell) = P(Z = n - 1, K \geq k - 1, N \geq n|L = \ell - 1)p_A, \quad \text{(A.14)}$$

or we either have not reached the minimum required length yet or we already reached this length in an earlier transition, i.e. $Z \neq n$, and obtain

$$P(Z = z, K \geq k, N \geq n|L = \ell) = P(Z = z - 1, K \geq k, N \geq n|L = \ell - 1)p_A. \quad \text{(A.15)}$$

In analogy, we obtain for poly-T tracts

$$P(Z = -n, K \geq k, N \geq n|L = \ell) = P(Z = -n + 1, K \geq k - 1, N \geq n|L = \ell - 1)p_T, \text{ and} \quad \text{(A.16)}$$

$$P(Z = -z, K \geq k, N \geq n|L = \ell) = P(Z = -z + 1, K \geq k, N \geq n|L = \ell - 1)p_T. \quad \text{(A.17)}$$

We initialize the recursion for a random sequence of length $L = 1$. We set the probability of being in state $Z = 0$, i.e. emitting a single nucleotide different from A or T, and observing at least 0 poly-A/T tracts of minimum length n to $P(Z = 0, K \geq 0, N \geq n|L = 1) = (1-p_A-p_T)$. We further set the probability of being in state $Z = 1$, i.e. emitting a single A, and observing at least 0 poly-A/T tracts of minimum length n to $P(Z = 1, K \geq 0, N \geq n|L = 1) = p_A$. In the special case that $n = 1$, we also intitialize $P(Z = 1, K \geq 1, N \geq 1|L = 1) = p_A$. In analogy, we define for the poly-T tracts $P(Z = -1, K \geq 0, N \geq n|L = 1) = p_T$ and $P(Z = -1, K \geq 1, N \geq 1|L = 1) = p_T$. All other probabilities are set to $P(Z = z, K \geq k, N \geq n|L = 1) = 0$.

We finally obtain the desired joint cumulative distribution $P(K \geq k, N \geq n|L)$ by a marginalization over all possible states, i.e.

$$P(K \geq k, N \geq n|L = \ell) = \sum_{z=-\ell}^{\ell} P(Z = z, K \geq k, N \geq n|L = \ell). \quad \text{(A.18)}$$

A.4. Numerical properties of the DNA helix

Free energy

$$\boldsymbol{r} = \begin{pmatrix} 1.9 & 1.3 & 1.6 & 1.5 \\ 1.9 & 3.1 & 3.6 & 1.6 \\ 1.6 & 3.1 & 3.1 & 1.3 \\ 0.9 & 1.6 & 1.9 & 1.9 \end{pmatrix}$$

Melting temperature

$$\boldsymbol{r} = \begin{pmatrix} 54.5 & 97.73 & 58.42 & 57.02 \\ 54.71 & 85.97 & 72.55 & 58.42 \\ 86.44 & 136.12 & 85.97 & 97.73 \\ 36.73 & 86.44 & 54.71 & 54.5 \end{pmatrix}$$

Base stacking energy

$$\boldsymbol{r} = \begin{pmatrix} -6.09 & -11.25 & -6.62 & -7.21 \\ -6.75 & -8.38 & -9.68 & -6.62 \\ -10.47 & -15.34 & -8.38 & -11.25 \\ -5.26 & -10.47 & -6.75 & -6.09 \end{pmatrix}$$

Persistence length

$$\boldsymbol{r} = \begin{pmatrix} 35.0 & 60.0 & 60.0 & 20.0 \\ 60.0 & 130.0 & 85.0 & 60.0 \\ 60.0 & 85.0 & 130.0 & 60.0 \\ 20.0 & 60.0 & 60.0 & 35.0 \end{pmatrix}$$

Propeller twist

$$\boldsymbol{r} = \begin{pmatrix} -17.3 & -6.7 & -14.3 & -16.9 \\ -8.6 & -12.8 & -11.2 & -14.3 \\ -15.1 & -11.7 & -12.8 & -6.7 \\ -11.1 & -15.1 & -8.6 & -17.3 \end{pmatrix}$$

Rise

$$\boldsymbol{r} = \begin{pmatrix} 3.16 & 3.41 & 3.63 & 3.89 \\ 3.23 & 4.08 & 3.6 & 3.63 \\ 3.47 & 3.81 & 4.08 & 3.41 \\ 3.21 & 3.47 & 3.23 & 3.16 \end{pmatrix}$$

Roll

$$\boldsymbol{r} = \begin{pmatrix} 0.3 & 0.5 & 4.5 & -0.8 \\ 0.5 & 6.0 & -6.2 & 4.5 \\ -1.3 & -6.2 & 6.0 & 0.5 \\ 2.8 & -1.3 & 0.5 & 0.3 \end{pmatrix}$$

Roll complexed

$$\boldsymbol{r} = \begin{pmatrix} 0.8 & -0.2 & 5.6 & 0.0 \\ 6.4 & 3.3 & 6.5 & 5.6 \\ 2.4 & -2.0 & 3.3 & -0.2 \\ 2.7 & 2.4 & 6.4 & 0.8 \end{pmatrix}$$

Slide

$$\boldsymbol{r} = \begin{pmatrix} -0.1 & -0.2 & 0.4 & -0.4 \\ 1.6 & 0.8 & 0.7 & 0.4 \\ 0.0 & 0.4 & 0.8 & -0.2 \\ 0.9 & 0.0 & 1.6 & -0.1 \end{pmatrix}$$

Slide complexed

$$\boldsymbol{r} = \begin{pmatrix} 0.1 & -0.6 & -0.3 & -0.7 \\ 0.4 & -0.1 & 0.7 & -0.3 \\ 0.1 & -0.3 & -0.1 & -0.6 \\ 0.1 & 0.1 & 0.4 & 0.1 \end{pmatrix}$$

Tilt

$$\boldsymbol{r} = \begin{pmatrix} 0.5 & 0.1 & 2.8 & 0.0 \\ -0.7 & 2.7 & 0.0 & 2.8 \\ 0.9 & 0.0 & 2.7 & 0.1 \\ 0.0 & 0.9 & -0.7 & 0.5 \end{pmatrix}$$

Tilt complexed

$$\boldsymbol{r} = \begin{pmatrix} 1.9 & 0.3 & 1.3 & 0.0 \\ 0.3 & 1.0 & 0.0 & 1.3 \\ 1.7 & 0.0 & 1.0 & -0.1 \\ 0.0 & 1.7 & 0.3 & 1.9 \end{pmatrix}$$

Tip

$$\boldsymbol{r} = \begin{pmatrix} 1.76 & 2.0 & 0.9 & 1.87 \\ -1.64 & 0.71 & 0.22 & 0.9 \\ 1.35 & 2.5 & 0.71 & 2.0 \\ 6.7 & 1.35 & -1.64 & 1.76 \end{pmatrix}$$

Twist

$$\boldsymbol{r} = \begin{pmatrix} 38.9 & 31.12 & 32.15 & 33.81 \\ 41.41 & 34.96 & 32.91 & 32.15 \\ 41.31 & 38.5 & 34.96 & 31.12 \\ 33.28 & 41.31 & 41.41 & 38.9 \end{pmatrix}$$

$\mathbf{V}_{\text{step}}$

$$\boldsymbol{r} = \begin{pmatrix} 2.9 & 2.3 & 2.1 & 1.6 \\ 9.8 & 6.1 & 12.1 & 2.1 \\ 4.5 & 4.0 & 6.1 & 2.3 \\ 6.3 & 4.5 & 9.8 & 2.9 \end{pmatrix}$$

Bendability

$$\boldsymbol{r} = \left(\begin{pmatrix} -0.274 & -0.205 & -0.081 & -0.28 \\ -0.0060 & -0.032 & -0.033 & -0.183 \\ 0.027 & 0.017 & -0.057 & -0.183 \\ 0.182 & -0.11 & 0.134 & -0.28 \end{pmatrix}, \begin{pmatrix} 0.015 & 0.04 & 0.175 & 0.134 \\ -0.246 & -0.012 & -0.136 & -0.057 \\ -0.0030 & -0.077 & -0.136 & -0.033 \\ 0.09 & 0.031 & 0.175 & -0.081 \end{pmatrix}, \right.$$
$$\left. \begin{pmatrix} -0.037 & -0.013 & 0.031 & -0.11 \\ 0.076 & 0.107 & -0.077 & 0.017 \\ 0.013 & 0.107 & -0.012 & -0.032 \\ 0.025 & -0.013 & 0.04 & -0.205 \end{pmatrix}, \begin{pmatrix} 0.068 & 0.025 & 0.09 & 0.182 \\ 0.194 & 0.013 & -0.0030 & 0.027 \\ 0.194 & 0.076 & -0.246 & -0.0060 \\ 0.068 & -0.037 & 0.015 & -0.274 \end{pmatrix} \right)$$

A.5. Listing of source code for section 4.1

```
/* * * * *
 * Set external parameters
 * * * * */
// working directory
String home = args[0];
// foreground data
String fgfile = args[1];
// background data
String bgfile = args[2];
// number of threads for computations
int threads = Integer.parseInt( args[3] );
// number of iterations of the holdout sampling
int numberIterations = Integer.parseInt( args[4] );
// algorithm for numerical optimization
byte alg = Optimizer.QUASI_NEWTON_BFGS;

// set the alphabet to a DNA alphabet
AlphabetContainer dnaAlpabet = new AlphabetContainer(new DNAAlphabet());

/* * * * *
 * Read foreground and background data
 * * * * */
Sample fg = new Sample(dnaAlpabet,new SparseStringExtractor(home+"/"+fgfile));
Sample bg = new Sample(dnaAlpabet,new SparseStringExtractor(home+"/"+bgfile));

// length of trancription factor binding sites
int length = fg.getElementLength();

/* * * * *
 * Define hyper-parameters of priors
 * * * * */
double essFg = 4;
double essBg = 1024;
double[] classMus = new double[]{-8.634};
double[] classVars = new double[]{5.082};
double[] kg = new double[]{2,0.005};
double[] kl = new double[]{0.005,0.002};

/* * * * *
 * Set parameters for numerically optimized
 * classifiers
 * * * * */
GenDisMixClassifierParameterSet parameters = new
    GenDisMixClassifierParameterSet( dnaAlpabet, length, alg, 1E-8,1E-8,1E-2,
    true, KindOfParameter.PLUGIN, true, threads );

/* * * * *
 * Create classifiers to be compared
 * * * * */
LinkedList<AbstractClassifier> list = new LinkedList<AbstractClassifier>();
// foreground orders
for(int i=0;i<2;i++){
  // background orders
```

```
  for(int j=0;j<5;j++){

    // Markov models learned by MAP principle
    AbstractClassifier cl = new ModelBasedClassifier(
        new BayesianNetworkModel(new
            BayesianNetworkModelParameterSet(dnaAlpabet, length, essFg, "",
          ModelType.IMM, (byte)i, LearningType.ML_OR_MAP)),
        new BayesianNetworkModel(new
            BayesianNetworkModelParameterSet(dnaAlpabet, length, essBg, "",
          ModelType.IMM, (byte)j, LearningType.ML_OR_MAP))
        );
    list.add( cl );

    // Markov models learned by MSP principle with transformed
       product-Dirichlet prior
    cl = new MSPClassifier( parameters,
        new CompositeLogPrior(),
        new BayesianNetworkScoringFunction(dnaAlpabet,length,essFg,true,
          new InhomogeneousMarkov(i)),
        new BayesianNetworkScoringFunction(dnaAlpabet,length,essBg,true,
          new InhomogeneousMarkov(j))
    );
    list.add( cl );

    // Markov models learned by MSP principle with Gaussian prior
    cl = new MSPClassifier( parameters,
        new SeparateGaussianLogPrior(kg,classVars,classMus),
        new BayesianNetworkScoringFunction(dnaAlpabet,length,essFg,true,
          new InhomogeneousMarkov(i)),
        new BayesianNetworkScoringFunction(dnaAlpabet,length,essBg,true,
          new InhomogeneousMarkov(j))
    );
    list.add( cl );

    // Markov models learned by MSP principle with transformed Laplace prior
    cl = new MSPClassifier( parameters,
        new SeparateLaplaceLogPrior(kl,classVars,classMus),
        new BayesianNetworkScoringFunction(dnaAlpabet,length,essFg,true,
          new InhomogeneousMarkov(i)),
        new BayesianNetworkScoringFunction(dnaAlpabet,length,essBg,true,
          new InhomogeneousMarkov(j))
    );
    list.add( cl );
  }
}


/* * * * *
 * Create object for assessment by
 * stratified holdout sampling
 * * * * */
RepeatedHoldOutExperiment exp = new RepeatedHoldOutExperiment( list.toArray(
    new AbstractClassifier[0] ));

/* * * * *
```

```
 * Performance, measures to be computed,
 * includes Sn for Sp of 0.999, PPV for Sn of 0.95, and FPR for Sn of 0.95
 * * * * */
MeasureParameters mp = new MeasureParameters(false,0.999,0.95,0.95);

/* * * * *
 * Parameters of assessment
 * * * * */
ClassifierAssessmentAssessParameterSet assessPS = new
    RepeatedHoldOutAssessParameterSet(
  PartitionMethod.PARTITION_BY_NUMBER_OF_SYMBOLS, length, true,
      numberIterations, new double[]{0.1,0.1});

/* * * * *
 * Start holdout sampling
 * * * * */
ListResult lr = exp.assess( mp, assessPS, new DefaultProgressUpdater(), fg,bg
    );

/* * * * *
 * Write results to disk
 * * * * */
FileManager.writeFile( new File(home+"/"+fgfile+"_results.xml"), lr.toXML() );
```

Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig und ohne fremde Hilfe verfasst habe. Ich habe keine anderen als die von mir angegebenen Quellen oder Hilfsmittel verwendet. Wörtliche und sinngemäße Zitate habe ich als solche kenntlich gemacht.
Ich habe mich bisher nicht um den Doktorgrad beworben.

Halle, den 30. April 2010

Jan Grau

Lebenslauf

Persönliche Daten

Name:	Jan Grau
Geburtsdatum:	12. November 1979
Geburtsort:	Bremen
Staatsangehörigkeit:	deutsch
Familienstand:	ledig

Schulbildung

1986–1990	Grundschule Neuenkirchen
1990–1992	Orientierungsstufe, Waldschule Schwanewede
1992–1999	Gymnasialer Zweig, Waldschule Schwanewede
28.06.1999	Abitur

Universitäre Bildung

10/2000-03/2006	Studium der Bioinformatik (Diplom) an der Martin-Luther-Universität Halle–Wittenberg
24.03.2006	Diplom der Bioinformatik

Tätigkeiten

04/2006-03/2010	wissenschaftlicher Mitarbeiter in der Arbeitsgruppe “Bioinformatik und Mustererkennung”, Institut für Informatik, Martin-Luther-Universität Halle–Wittenberg
04/2010-09/2010	wissenschaftlicher Mitarbeiter in der Arbeitsgruppe “Bioinformatik”, Institut für Informatik, Martin-Luther-Universität Halle–Wittenberg

Halle, den 30. April 2010

Jan Grau