Predicting DNA binding sites using generative, discriminative, and hybrid learning principles

Dissertation

zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften (Dr. rer. nat.)

der Naturwissenschaftliche Fakultät III
Agrar-, Geowissenschaften, Mathematik und Informatik
der Martin-Luther-Universität Halle-Wittenberg

vorgelegt von

Jens Keilwagen

geb. am 02. Oktober 1981 in Oschatz

Halle (Saale), im April 2010

Dipl.-Bioinform. Jens Keilwagen
Arbeitsgruppe Dateninspektion
Abteilung Molekulare Genetik
Leibniz Institut für Pflanzengenetik und Kulturpflanzenforschung

Dienstanschrift:

Leibniz Institut für Pflanzengenetik und Kulturpflanzenforschung
Correnssstraße 3
06466 Gatersleben

Dissertation,
der Naturwissenschaftliche Fakultät III
der Martin-Luther-Universität Halle-Wittenberg
vorgelegt am 01.04.2010,
verteidigt am 08.07.2010.

Gutachter:

1. Prof. Dr. Ivo Grosse,
   Martin-Luther-Universität Halle-Wittenberg
2. Dr. Gunnar Rätsch,
   Friedrich-Miescher-Laboratorium, Max-Planck-Institut Tübingen

For Daniela.

Acknowledgements Above all, I would like to thank three wonderful colleagues for their permanent support: Ivo Grosse, who inspired me to dip into the fascinating field of probabilistic models, Jan Grau, who shares my enthusiasm for solving problems, and Marc Strickert, who constantly encouraged me to continue my studies without getting lost.

For their moral support, I like to thank my family. Specifically, I like to thank my beloved wife Daniela for showing me that there is life outside of science, my parents for piquing my curiosity about mathematics, and Lore Wendler for encouraging me to study despite all obstacles.

Furthermore, I like to thank Thomas Abeel, Lothar Altschmied, Helmut Bäumlein, Jan Baumbach, Guillaume Bouchard, Robert Castelo, Jesús Cerquides, Udo Conrad, Carolin Delker, André Gohr, Swetlana Friedel, Berit Haldemann, Dirk Holste, Astrid Junker, Thomas Kohl, Simon Lacoste-Julien, Chaim Linart, Stefanie Lück, Hendrik Mehlhorn, Birgit Möller, Gudrun Mönke, Michaela Mohr, Ivan Paponov, Yvonne Pöschl, Stefan Posch, Marcel Quint, Gunnar Rätsch, Michael Scharfe, Roland Schnee, Stefan Schmidt, Michael Seifert, Andrew Smith, Sören Sonnenburg, Thomas Thiel, Michael Zhang, Valentin Ziegler, and Alexander Zien for fruitful discussions.

All of the work in this thesis has been done at IPK Gatersleben. The computational experiments conducted for this thesis profited a lot from the exceptional computing environments at IPK Gatersleben and URZ Halle. This work was supported in part by the grants 0312706A/D by the German Ministry of Education and Research (BMBF) and XP3624HP/0606T by the Ministry of Education and Cultural Affairs of Saxony-Anhalt.

"The scientist finds his reward in what Henri Poincaré calls the joy of comprehension, and is not in the possibilities of application to which any discovery of his may lead."

Albert Einstein

Abstract

Each cell of an organism contains the same genomic DNA encoding the same information, but cells, tissues, and organs differ in appearance, functions, and many other aspects. These differences are promoted by a number of complex regulatory processes on the DNA and its transcribed RNA. Over the last years, new high-throughput technologies and the development of new bioinformatics methods for the prediction of short signal sequences have provided many new insights and hypotheses. Despite the overwhelming scientific progress of the last years, many aspects of how the cellular machinery recognizes and binds these short sequences with high accuracy are still not fully understood.

Improved computational methods often apply either a new model or a new learning principle gaining a superior accuracy for a specific source of data. It is therefore hard to compare different approaches on the same basis allowing to examine their strengths or weaknesses. In this work, we present a formal framework for learning probabilistic models from the family of Markov random fields that allows an interpolation between several established learning principles including generative, discriminative, hybrid, Bayesian, and non-Bayesian learning principles. This framework enables comparisons of different models and different learning principles using the same a-priori knowledge. We implement this framework as an open-source Java library that is easily extensible and that allows modular combinations of different probabilistic models, a-priori knowledge, and learning principles for building sophisticated classifiers. Applying these models and learning principles to the tasks of splice site recognition, transcription start site recognition, database curation, and de-novo motif discovery, we obtain competitive or superior performances compared to state-of-the-art approaches.

Contents

1 Introduction 5

2 Notation 8

3 Classification 9
- 3.1 Learning principles . . . 10
  - 3.1.1 Established learning principles . . . 10
  - 3.1.2 Unified generative-discriminative learning principle . . . 13
  - 3.1.3 Optimization . . . 17
- 3.2 Classification measures . . . 17

4 Probabilistic models for DNA sequences 20
- 4.1 Models for sequences with arbitrary length . . . 20
  - 4.1.1 Homogeneous Markov models . . . 21
  - 4.1.2 Cyclic Markov models . . . 21
- 4.2 Models for sequences with fixed length . . . 22
  - 4.2.1 Moral Bayesian networks . . . 24
  - 4.2.2 Markov random fields . . . 27
- 4.3 Models for sequence length . . . 28
  - 4.3.1 Skew normal models . . . 29
- 4.4 Composite models . . . 30
  - 4.4.1 Mixture models . . . 30
  - 4.4.2 DNA Strand models . . . 31
  - 4.4.3 Extended ZOOPS models . . . 31
  - 4.4.4 Independent product models . . . 32

5 Priors **34**
5.1 Dirichlet density . . . 34
5.1.1 Dirichlet prior using softmax basis . . . 34
5.1.2 Product-Dirichlet prior for Markov random fields . . . 35
5.2 Prior of skew normal models . . . 44
5.3 Prior of composite models . . . 45

6 Implementation **47**

7 Comparison of learning principles **50**
7.1 Comparison of the ML and the MCL principle . . . 50
7.2 Comparison of the MAP and the MSP principle . . . 53
7.2.1 Case study 1: Mixture models for donor splice sites . . . 53
7.2.2 Case study 2: Influence of the size of the training data set . . . 54
7.3 Applications of the unified generative-discriminative learning principle . . . 56
7.3.1 Case study 1: Transcription factor binding site recognition . . . 56
7.3.2 Case study 2: Influence of the size of the training data set and of the prior . . . 61
7.3.3 Case study 3: Donor splice site recognition . . . 65
7.4 Conclusions . . . 66

8 Donor splice site recognition in Caenorhabditis elegans **68**

9 Recognition of human transcription start sites **72**
9.1 Established protocols . . . 72
9.2 Approach . . . 75
9.3 Results and conclusions . . . 76
9.4 Conclusions . . . 80

10 Computational reassessment of transcription factor binding site annotations **82**
10.1 Approach . . . 84
10.2 Results and Discussion . . . 85
10.2.1 Results for seven data sets of *Escherichia coli* . . . 85
10.2.2 Validation of MotifAdjuster results for NarL . . . 87
10.2.3 Prediction of a novel NarL binding site . . . 89
10.3 Conclusions . . . 91

11 Discriminative de-novo motif discovery utilizing positional preference 92
11.1 Approach 94
11.2 Results and Discussion 100
11.2.1 Comparison of Dispom with existing tools 101
11.2.2 Finding two motifs simultaneously 107
11.2.3 Applying Dispom to promoters of auxin-responsive genes 107
11.3 Conclusions 111

12 Conclusions and Outlook 112

Glossary 114

List of Figures 116

List of Tables 118

Bibliography 119

Chapter 1

Introduction

In the bestseller *"Megatrends 2000: Ten new directions for the 1990's"*, John Naisbitt and Patricia Aburdene predict that the next era will be the *"Age of Biology."* Specifically, they write *"As we move through the next millennium, biotechnology will be as important as the computer"* [Naisbitt and Aburdene, 1990]. Today, almost 20 years later, we have to state that their prediction has been correct. Biology has undergone a serious development as several new experimental techniques arose allowing to gain new insights in fields that seemed to be too far away even a few years ago. This development already affects our daily life, and currently we do not see an end to this development, which is absolutely in accordance with the thesis of made in *"Age of Biology."*

Nowadays, biology utilizes concepts and techniques from many areas of science such as chemistry, physics, mathematics, and informatics, to deepen our understanding of many fundamental biological processes. These sciences contribute to the wide area of the life sciences including, besides pure biology, for instance biochemistry, biophysics, biomathematics, and bioinformatics. Each of these life sciences contributes its individual strength to write this story of success and at the same time the borders between individual sciences become blurred. Considering the role of bioinformatics, the classification of unlabeled data is one of the main challenges that can be tackled. Based on classification the generation of hypotheses and experimental design belong to the field of bioinformatics as well. Besides the classification of unlabeled data, bioinformatics assists with data storage [Roos, 2001, Cochrane and Galperin, 2010], data visualization [Barrett et al., 2005, Shannon et al., 2003], and comparison of different data sources [Altschul et al., 1990, Bennetzen and Ma, 2003, Yang, 2007].

Classification is required manifold and includes, for instance, the recognition of patterns from image data [Carpenter et al., 2006, Peng, 2008], the prediction of protein and RNA structure [McGuffin et al., 2000, Hofacker, 2003], as well as gene prediction [Bernal et al., 2007, Schweikert et al., 2009] and the inference of coexpression or transcriptional regulatory networks [Stuart et al., 2003, Lee et al., 2002]. Focusing on DNA sequence analysis, classification is often connected to recognition of various binding sites (BSs) of some biological process driving the activity of one or a few genes. The recognition of DNA BSs includes, for instance, the recognition of transcription factor binding sites (TFBSs) [Bailey and Eklan, 1994, Pavesi et al., 2001, Kim et al., 2008], transcription start sites (TSS) [Davuluri et al., 2001, Sonnenburg et al., 2006, Abeel et al., 2009], translation initiation sites [Hatzigeorgiou, 2002, Saeys et al., 2007], donor and acceptor splice sites [Salzberg, 1997a, Yeo and Burge, 2004, Sonnenburg et al., 2007], alternative splice sites [Foissac and Schiex, 2005, Sinha et al., 2009], splicing enhancer and silencer [Fairbrother et al., 2002, Wang et al., 2004], nucleosome BSs [Segal et al., 2006, Peckham et al., 2007, Field et al., 2008], insulator BSs [Kim et al., 2007], and micro-RNA BSs [Krek et al., 2005, Sethupathy et al., 2006]. Since these BSs are conserved to a certain degree, they are also used as building blocks for subsequent bioinformatics tasks, such as gene finding [Fickett, 1996, Burge and Karlin, 1997, Schweikert et al., 2009] and the computation of spliced alignments [Gelfand et al., 1996, Florea et al., 1998, De Bona et al., 2008].

DNA, which carries the genetic information of higher organisms, is composed of repeating units called nucleotides. Each of these nucleotides comprises one of the four organic bases adenine, cytosine, guanine, and thymine abbreviated as A, C, G, and T. The sequence of these bases along the sugar phosphate backbone of the DNA encodes the genetic information. However, the DNA sequence in higher organisms is very long, as for instance human DNA contains approximately 3.3 billion bases. In addition, the double stranded DNA allows that BSs are located on either of the reverse complementary strands. Hence, the recognition of few functional BSs is challenging. Here, we consider the recognition of three different types of BSs, namely TFBSs, TSSs, and splice sites.

Transcription factors (TFs) are proteins which contain one or more DNA-binding domains, and attach to specific patterns of DNA, so-called TFBSs. Binding of TFs to their specific TFBSs in the promoter region of a gene can activate or repress the transcription of this gene, and therefore affects all downstream processes. However, the ability to control the transcription of a target gene may depend on the TFBS itself, its strand orientation, its position with respect to the TSS, and possibly the presence, conservation, orientation, and distance to additional TFBSs. Hence, the recognition of functional TFBSs is very difficult, and has not yet been solved satisfactorily.

While TFBSs regulate the transcription of genes, the TSS defines a specific site or region where transcription is started and RNA is transcribed. The initiation of transcription is a complex process that functions differently in prokaryotes and eukaryotes. In prokaryotes mainly two specific patterns at positions -35 and -10 bp before the TSS are recognized by specific TFs denoted as σ-factors guiding the RNA polymerase, whereas in eukaryotes the situation is less clear and several TFs seem to mediate the binding of the RNA polymerase. After binding of the RNA polymerase, transcription is initiated and RNA is transcribed from DNA template. Since the TSS defines the start of the RNA, it has a decisive influence on the gene product.

Similarly, splicing has a decisive influence on the RNA and therefore on the translated amino acid sequence. Pre-mRNA, which is transcribed from DNA, consists of two different types of segments, called exons and introns. Introns are removed during splicing, while exons are retained in the mRNA. In most cases, splicing is catalyzed by a complex of small nuclear ribonucleoproteins called the spliceosome. Splice sites are the borders between exons and introns in the pre-mRNA. The border between an exon and an intron is denoted as *donor* or $5'$ splice site, whereas the border between an intron and an exon is denoted as *acceptor* or $3'$ splice site.

Despite the common goal of recognizing BSs, the tools used for the specific tasks differ in various aspects. However, many tools use probabilistic models to solve the problem at hand, since they often allow a simple interpretation of the model parameters. In contrast non-probabilistic models, as for instance support vector machines, which utilize discriminative learning, outperform probabilistic models in some applications [Sonnenburg et al., 2006, Sonnenburg et al., 2007] but are harder to interpret. However, the performance of probabilistic models is determined by the model parameters. These parameters are estimated from training data using a predefined learning principle. Often the models used are quite complex and relatively well adapted to the problem at hand, whereas the learning principle is often chosen arbitrarily.

In most cases the learning principles employed belong to a class of *generative* learning principles. Generative learning principles aim at representing the distribution of the training data in each of the classes as accurately as possible. This may seem the only sensible way of estimating model parameters, but other, so-called *discriminative*, learning principles that are more directly linked to the classification task have been proposed. These learning principles aim at discriminating the training data well. In this work, we present a *hybrid* learning principle in subsection *Unified generative-discriminative learning principle* (page 13) that contains several well-known learning principles and allows to interpolate between them. In addition, we present a prior in subsection *Product-Dirichlet prior for Markov random fields* (page 35) that can be easily used for this learning principle and allows to incorporate biological prior knowledge. Using this prior, we are able to compare generative and discriminative learning principles more directly by using the same prior knowledge.

The work is structured as follows: First, we introduce some notations used in the following chapters. Second, we present a classifier based on probabilistic models and some learning principles. Subsequently, we present probabilistic models tailored for DNA sequence analysis in chapter *Probabilistic models for DNA sequences* (page 20) and the corresponding priors in chapter *Priors* (page 34). Following the theoretical part of this work, we provide some details about the implementation in chapter *Implementation* (page 47). We implement the open-source Java library Jstacs including all learning principles and models aside from infrastructure for classification problems, which allow to use Jstacs easily for building tools for similar problems. In chapter *Comparison of learning principles* (page 50), we prove the utility of the introduced concept using well known data.

Employing our methods, we demonstrate in four biological applications how probabilistic models and different learning principles can be used to improve the solution of many important bioinformatics problems. In the first case study, we investigate the recognition of donor splice sites for the model organism *Caenorhabditis elegans* using a discriminative learning principle. Similarly in the second case study, we investigate the recognition of human TSSs based on tags of cap-analysis of gene expression using a discriminative learning principle. In the third case study, we investigate the quality of TFBS annotations in the database CoryneRegNet [Baumbach et al., 2009] using a generative learning principle. Finally in the fourth case study, we investigate whether a discriminative learning principle and a learnable position distribution for TFBSs might help to improve de-novo motif discovery.

Chapter 2

Notation

In this chapter, we introduce the notation of this work. Although the learning principles and probabilistic models presented in the following chapters also apply to other sequence data, we focus on DNA sequences, which are composed of nucleotides abbreviated as A, C, G, and T. Hence, we define the alphabet $\Sigma := \{A, C, G, T\}$. We denote a DNA sequence of sequence length L by $\underline{x} := (x_1, x_2, \ldots, x_L)$ where $x_\ell \in \Sigma$ for $\ell \in [1, L]$, and we denote the subsequence from position ℓ_1 to position ℓ_2 of sequence $\underline{x}$ by $\underline{x}_{\ell_1 \ldots \ell_2} := (x_{\ell_1}, x_{\ell_1+1}, \ldots, x_{\ell_2})$. For convenience, we allow that $\ell_1 > \ell_2$ yielding an empty subsequence. We denote the reverse complement of $\underline{x}$ by $\underline{x}^{RC}$, which allows us to search for non-palindromic patterns on both strands of DNA. Handling multiple sequences, we denote a data set of N independent identical distributed (i.i.d.) sequences by $\underline{D} := (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N)$.

Considering a specific classification task, each sequence $\underline{x}$ has a class label $c \in \mathcal{C}$ where $\mathcal{C}$ denotes the set of all possible class labels, as for instance *"BS"* and *"no BS."* In case of two possible class labels, these are often denoted more generally as *positive* and *negative* or *foreground* and *background*. We denote the class labels for data set $\underline{D}$ by $\underline{C} := (c_1, c_2, \ldots, c_N) \in \mathcal{C}^N$ where c_n denotes the class label of sequence $\underline{x}_n$. For shortness of notation, we denote a labeled data set by $(\underline{D}, \mathcal{C})$. For counting certain patterns, we use the Kronecker symbol denoted by $\delta_{i,j}$ which is equal to 1 if both indices are equal and otherwise 0. Depending on the problem, i and j can be replaced by symbols, subsequences, or class labels.

In chapter *Probabilistic models for DNA sequences* (page 20), we introduce probabilistic models, which are used in the later chapters on biological data sets. Although the models are quite diverse, we use a common notation for all models. For any model $\mathcal{M}$, we denote the parameters of the model by $\underline{\lambda}^{(\mathcal{M})}$. Using these parameters, we denote the likelihood of the model $\mathcal{M}$ by $P^{\mathcal{M}}\left(c, \underline{x} \middle| \underline{\lambda}^{(\mathcal{M})}\right)$. In section *Learning principles* (page 10), we consider different ways of estimating the parameters $\underline{\lambda}^{(\mathcal{M})}$. Some of the learning principles presented in this section use prior knowledge aside from training data. Typically, this prior knowledge is given as prior density on the parameters of the model. We denote the prior on the parameters of model $\mathcal{M}$ by $Q^{\mathcal{M}}\left(\underline{\lambda}^{(\mathcal{M})} \middle| \underline{\alpha}^{(\mathcal{M})}\right)$ where $\underline{\alpha}^{(\mathcal{M})}$ denotes the hyper-parameters controlling the characteristics of the prior. In chapter *Priors* (page 34), we consider priors for all models of chapter *Probabilistic models for DNA sequences* (page 20) in more detail. For a compact notation, we omit the subscript $\mathcal{M}$ for parameters and hyper-parameters if their meaning is clear from the context.

Chapter 3

Classification

In this chapter, we present the theoretical framework for classifying sequences, which is of fundamental interest in many biological applications. Classifiers are used to decide whether a sequence is, for instance, a splice site [Salzberg, 1997a, Yeo and Burge, 2004], a transcription factor binding site [Ben-Gal et al., 2005], a nucleosome binding site [Segal et al., 2006] or not. More generally, classifiers are used to assign class labels to unlabeled sequences.

In this work, we consider classifiers based on probabilistic models with parameter vector $\underline{\lambda}$, which are described in more detail in the next chapter. Based on these models the decision criterion [Hastie et al., 2009] of the classifier is defined as

$$\hat{c} := \underset{c \in \mathcal{C}}{\operatorname{argmax}} P\left(c \middle| \underline{x}, \underline{\lambda}\right) = \underset{c \in \mathcal{C}}{\operatorname{argmax}} P\left(c, \underline{x} \middle| \underline{\lambda}\right) = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \left[P\left(c \middle| \underline{\lambda}\right) \cdot P\left(\underline{x} \middle| c, \underline{\lambda}\right) \right], \tag{3.1}$$

where $P\left(c \middle| \underline{x}, \underline{\lambda}\right)$ is the conditional likelihood of class label c given sequence $\underline{x}$ and parameter vector $\underline{\lambda}$, $P\left(c, \underline{x} \middle| \underline{\lambda}\right)$ is the likelihood of class label c and sequence $\underline{x}$ given parameter vector $\underline{\lambda}$, $P\left(c \middle| \underline{\lambda}\right)$ is the probability of class label c given parameter vector $\underline{\lambda}$, and $P\left(\underline{x} \middle| c, \underline{\lambda}\right)$ is the conditional probability of sequence $\underline{x}$ given class label c and parameter vector $\underline{\lambda}$. In the case of a binary classifier, i. e., $\mathcal{C} = \{0, 1\}$, we denote the classes by *foreground class* and *background class*. For such a binary classifier, the decision criterion can be reformulated in terms of likelihood ratio as

$$\hat{c} := \begin{cases} 0 & , \frac{P(0, \underline{x} | \underline{\lambda})}{P(1, \underline{x} | \underline{\lambda})} \geq 1 \\ 1 & , \text{otherwise} \end{cases}. \tag{3.2a}$$

Decomposing the likelihood and substituting the ratio of probabilities $\frac{P(1|\underline{\lambda})}{P(0|\underline{\lambda})}$ by a threshold $T \in \mathbb{R}_0^+$, we obtain the simplified decision criterion based on the conditional probabilities of sequence $\underline{x}$

$$\hat{c} := \begin{cases} 0 & , \frac{P(\underline{x} | 0, \underline{\lambda})}{P(\underline{x} | 1, \underline{\lambda})} \geq T \\ 1 & , \text{otherwise} \end{cases}. \tag{3.2b}$$

With this decision criterion at hand, there are two important questions:

1. How can we estimate the parameter vector of a classifier from training data?
2. How can we compare different classifiers regarding their performance?

We address both questions in the remainder of this chapter.

In this work, we use a single parameter vector $\underline{\lambda}$ for classification. Complementary to this approach, there are Bayesian methods that do not use a single parameter vector but rather a density of parameter vectors [Geman and Geman, 1984, Casella and George, 1992, Liu, 2002]

3.1 Learning principles

In this section, we consider the first question *"How can we estimate the parameter vector of a classifier from training data?"* For this purpose, we present learning principles for the parameter vector of classifiers for a data set $\underline{D}$ and the corresponding class labels $\underline{C}$.

In the first subsection, we present six learning principles that are established in the machine-learning community and which are nowadays also used in the field of bioinformatics. In the second subsection, we propose a generalization of these six learning principles which allows to make a more detailed comparison of different learning principles.

3.1.1 Established learning principles

Learning principles can be categorized by two criteria. On the one hand, they can be divided by their objective into generative, discriminative, and hybrid learning principles. Generative learning principles aim at an accurate representation of the distribution of the training data in each of the classes, discriminative learning principles aim at an accurate classification of the training data into the classes, whereas hybrid learning principles are interpolations between generative and discriminative learning principles. On the other hand, learning principles can be divided by the utilization of prior knowledge into Bayesian and non-Bayesian. We call learning principles Bayesian if they incorporate a prior density $Q\left(\underline{\lambda}\middle|\underline{\alpha}\right)$ on the parameter vector $\underline{\lambda}$, where $\underline{\alpha}$ denotes a vector of hyper-parameters, while we call learning principles non-Bayesian if they only use the data - without any prior - to estimate the parameter vector. In Table 3.1, we show six established learning principles and the assignment to the above mentioned criteria. These learning principles are described in more detail in the remainder of this section.

Generative learning principles

The maximum likelihood (ML) principle is one of the first learning principles used in bioinformatics. Originally, it was proposed by R. A. Fisher at the beginning of the 20^{th} century [Fisher, 1922, Aldrich, 1997]. The ML principle aims at finding the parameter vector $\hat{\underline{\lambda}}_{\mathrm{ML}}$ that maximizes the likelihood of the labeled data set $(\underline{C}, \underline{D})$ given the parameter vector $\underline{\lambda}$,

$$\hat{\underline{\lambda}}_{\mathrm{ML}} := \underset{\underline{\lambda}}{\operatorname{argmax}}\, P\left(\underline{C}, \underline{D}\middle|\underline{\lambda}\right) \tag{3.3a}$$

$$= \underset{\underline{\lambda}}{\operatorname{argmax}} \left[\sum_{n=1}^{N} \log P\left(c_n, \underline{x}_n\middle|\underline{\lambda}\right)\right]. \tag{3.3b}$$

		prior knowledge	
		non-Bayesian	**Bayesian**
objective	**generative**	ML	MAP
	hybrid	GDT	PGDT
	discriminative	MCL	MSP

Table 3.1: Characterisation of learning principles. The table shows six established learning principles that can be distinguished by two criteria, namely the objective of the learning principle, which is either generative, hybrid, or discriminative, and the usage of prior knowledge with the two possibilities non-Bayesian and Bayesian. The four elementary learning principles are the generative, non-Bayesian maximum likelihood (ML) principle, the generative, Bayesian maximum a posteriori (MAP) principle, the discriminative, non-Bayesian maximum conditional likelihood (MCL) principle, and the discriminative, Bayesian maximum supervised posterior (MSP) principle. The hybrid learning principles, which interpolate between generative and discriminative learning principles, are the non-Bayesian generative-discriminative trade-off (GDT) principle and the penalized generative-discriminative trade-off (PGDT) principle.

However, for many applications, the amount of sequence data available for the training is very limited. For this reason, the ML principle often leads to suboptimal classification performance, e.g. due to zero-occurrences of some nucleotides or oligonucleotides in the training data sets.

The maximum a posteriori (MAP) principle, which applies a prior $Q\left(\underline{\lambda}\middle|\underline{\alpha}\right)$ to the parameter vector, establishes a theoretical foundation to alleviate this problem, and at the same time it allows the inclusion of prior knowledge aside from the training data [Bishop, 2006]. The MAP principle aims at finding the parameter vector $\hat{\underline{\lambda}}_{\mathrm{MAP}}$ that maximizes the posterior density,

$$\begin{aligned}
\hat{\underline{\lambda}}_{\mathrm{MAP}} &:= \operatorname*{argmax}_{\underline{\lambda}} P\left(\underline{\lambda}\middle|\underline{C},\underline{D},\underline{\alpha}\right) && (3.4a)\\
&= \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C},\underline{D}\middle|\underline{\lambda}\right)\cdot Q\left(\underline{\lambda}\middle|\underline{\alpha}\right)\right] && (3.4b)\\
&= \operatorname*{argmax}_{\underline{\lambda}} \left[\left[\sum_{n=1}^{N}\log P\left(c_n,\underline{x}_n\middle|\underline{\lambda}\right)\right]+\log Q\left(\underline{\lambda}\middle|\underline{\alpha}\right)\right]. && (3.4c)
\end{aligned}$$

If for a given family of likelihood functions $P\left(\underline{C},\underline{D}\middle|\underline{\lambda}\right)$ the posterior $P\left(\underline{\lambda}\middle|\underline{C},\underline{D},\underline{\alpha}\right)$ is in the same family of distributions as the prior $Q\left(\underline{\lambda}\middle|\underline{\alpha}\right)$, i. e., if

$$Q\left(\underline{\lambda}\middle|\tilde{\underline{\alpha}}\right)=P\left(\underline{\lambda}\middle|\underline{C},\underline{D},\underline{\alpha}\right)\propto P\left(\underline{C},\underline{D}\middle|\underline{\lambda}\right)\cdot Q\left(\underline{\lambda}\middle|\underline{\alpha}\right), \tag{3.5}$$

the prior is said to be *conjugate* to this class of likelihood functions, and the hyper-parameter vector $\tilde{\underline{\alpha}}$ incorporates both prior knowledge and training data. Conjugate priors often allow an interpretation of the hyper-parameter vector as stemming from an a-priorily observed set of *"pseudo-data."* In addition, it allows finding the optimal parameter vector $\hat{\underline{\lambda}}_{\mathrm{MAP}}$ analytically provided one can determine the maximum of the prior analytically.

Discriminative learning principles

Recently, discriminative learning principles instead of generative ones have been shown to be promising in the field of bioinformatics [Yakhnenko et al., 2005, Culotta et al., 2005, Redhead and Bailey, 2007, Grau et al., 2007, Keilwagen et al., 2007]. The discriminative analogue to the ML principle is the maximum conditional likelihood (MCL) principle [Wettig et al., 2002, Grossman and Domingos, 2004, Greiner et al., 2005, Pernkopf and Bilmes, 2005, Feelders and Ivanovs, 2006] that aims at finding the parameter vector $\underline{\hat{\lambda}}_{\mathrm{MCL}}$ that maximizes the conditional likelihood of the labels $\underline{C}$ given the data $\underline{D}$ and the parameter vector $\underline{\lambda}$,

$$\underline{\hat{\lambda}}_{\mathrm{MCL}} := \operatorname*{argmax}_{\underline{\lambda}} P\left(\underline{C}\middle|\underline{D},\underline{\lambda}\right) \tag{3.6a}$$

$$= \operatorname*{argmax}_{\underline{\lambda}} \left[\sum_{n=1}^{N} \log P\left(c_n\middle|\underline{x}_n,\underline{\lambda}\right)\right]. \tag{3.6b}$$

The effects of limited data may be even more severe when using the MCL principle compared to generative learning principles [Ng and Jordan, 2002]. To overcome this problem, the maximum supervised posterior (MSP) principle [Grünwald et al., 2002, Cerquides and de Mántaras, 2005] has been proposed as discriminative analogue to the MAP principle. In analogy to Equation (3.4b), the MSP principle aims at finding the parameter vector $\underline{\hat{\lambda}}_{\mathrm{MSP}}$ that maximizes the product of the conditional likelihood and the prior density,

$$\underline{\hat{\lambda}}_{\mathrm{MSP}} := \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C}\middle|\underline{D},\underline{\lambda}\right)\cdot Q\left(\underline{\lambda}\middle|\underline{\alpha}\right)\right] \tag{3.7a}$$

$$= \operatorname*{argmax}_{\underline{\lambda}} \left[\left[\sum_{n=1}^{N} \log P\left(c_n\middle|\underline{x}_n,\underline{\lambda}\right)\right] + \log Q\left(\underline{\lambda}\middle|\underline{\alpha}\right)\right]. \tag{3.7b}$$

Generative-discriminative trade-offs

During the last years, different hybrid learning principles have been proposed in the machine-learning community [Bouchard and Triggs, 2004, Lasserre et al., 2006, Bouchard, 2007]. Hybrid learning principles aim at combining the strengths of generative and discriminative learning principles. Here, we follow the ideas of Bouchard and co-workers who propose an interpolation between the generative ML principle and the discriminative MCL principle [Bouchard and Triggs, 2004] as well as the generative MAP principle and the discriminative MSP principle [Bouchard, 2007].

The generative-discriminative trade-off (GDT) proposed in [Bouchard and Triggs, 2004] aims at finding the parameter vector $\underline{\lambda}$ that maximizes the weighted product of conditional likelihood and likelihood, i. e.,

$$\hat{\underline{\lambda}}_{\mathrm{GDT}} := \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C} \middle| \underline{D}, \underline{\lambda}\right)^{1-\gamma} \cdot P\left(\underline{C}, \underline{D} \middle| \underline{\lambda}\right)^{\gamma} \right] \tag{3.8a}$$

$$= \operatorname*{argmax}_{\underline{\lambda}} \left[(1-\gamma) \left[\sum_{n=1}^{N} \log P\left(c_n \middle| \underline{x}_n, \underline{\lambda}\right) \right] + \gamma \left[\sum_{n=1}^{N} \log P\left(c_n, \underline{x}_n \middle| \underline{\lambda}\right) \right] \right], \tag{3.8b}$$

for given weight $\gamma \in [0,1]$. As special cases of the GDT principle, we obtain the ML principle for $\gamma = 1$ and the MCL principle for $\gamma = 0$. By varying γ between 0 and 1, different beneficial trade-offs can be obtained for classification

In close analogy to the MAP and the MSP principle, which are obtained by multiplying a prior to the likelihood and conditional likelihood, respectively, the penalized generative-discriminative trade-off (PGDT) principle aims at finding the parameter vector $\underline{\lambda}$ that maximizes the product of the objective function of the GDT principle and the prior,

$$\hat{\underline{\lambda}}_{\mathrm{PGDT}} := \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C} \middle| \underline{D}, \underline{\lambda}\right)^{1-\gamma} \cdot P\left(\underline{C}, \underline{D} \middle| \underline{\lambda}\right)^{\gamma} \cdot Q\left(\underline{\lambda} \middle| \underline{\alpha}\right) \right] \tag{3.9a}$$

$$= \operatorname*{argmax}_{\underline{\lambda}} \left[(1-\gamma) \left[\sum_{n=1}^{N} \log P\left(c_n \middle| \underline{x}_n, \underline{\lambda}\right) \right] + \gamma \left[\sum_{n=1}^{N} \log P\left(c_n, \underline{x}_n \middle| \underline{\lambda}\right) \right] + \log Q\left(\underline{\lambda} \middle| \underline{\alpha}\right) \right], \tag{3.9b}$$

for given weight $\gamma \in [0,1]$. As special cases of the PGDT principle, we obtain the MAP principle for $\gamma = 1$ and the MSP principle for $\gamma = 0$.

3.1.2 Unified generative-discriminative learning principle

Comparing Equations (3.3a), (3.4b), (3.6a), (3.7a), (3.8a), and (3.9a), we find that the following three terms are sufficient for defining the established learning principles:

1. the conditional likelihood $P\left(\underline{C} \middle| \underline{D}, \underline{\lambda}\right)$,
2. the likelihood $P\left(\underline{C}, \underline{D} \middle| \underline{\lambda}\right)$, and
3. the prior $Q\left(\underline{\lambda} \middle| \underline{\alpha}\right)$.

With the goal of unifying and generalizing the six learning principles presented, we propose a unified generative-discriminative learning principle that aims at finding the parameter vector $\underline{\lambda}$ that maximizes the weighted product of conditional likelihood, likelihood, and prior [Keilwagen et al., 2010c], i. e.,

$$\hat{\underline{\lambda}} := \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C} \middle| \underline{D}, \underline{\lambda}\right)^{\beta_0} \cdot P\left(\underline{C}, \underline{D} \middle| \underline{\lambda}\right)^{\beta_1} \cdot Q\left(\underline{\lambda} \middle| \underline{\alpha}\right)^{\beta_2} \right] \tag{3.10a}$$

$$= \operatorname*{argmax}_{\underline{\lambda}} \left[\beta_0 \left[\sum_{n=1}^{N} \log P\left(c_n \middle| \underline{x}_n, \underline{\lambda}\right) \right] + \beta_1 \left[\sum_{n=1}^{N} \log P\left(c_n, \underline{x}_n \middle| \underline{\lambda}\right) \right] + \beta_2 \log Q\left(\underline{\lambda} \middle| \underline{\alpha}\right) \right], \tag{3.10b}$$

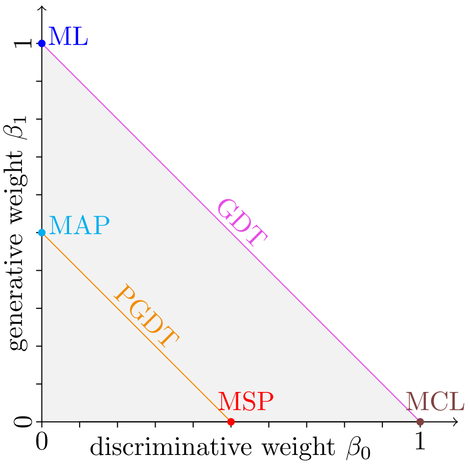


Figure 3.1: Two-dimensional projection of the simplex spanned by the weights $\underline{\beta}$ of the unified generative-discriminative learning principle and the established learning principles for the specific weights encoded by colors. The points $(0, 1)$, $(0, 0.5)$, $(1, 0)$, and $(0.5, 0)$ refer to the ML, the MAP, the MCL, and the MSP principle, respectively, while the lines $\beta_1 = 1 - \beta_0$ and $\beta_1 = 0.5 - \beta_0$ refer to the GDT and the PGDT principle, respectively.

with the weighting factors $\underline{\beta} := (\beta_0, \beta_1, \beta_2)$, $\beta_0, \beta_1, \beta_2 \in [0, 1]$, and $\beta_0 + \beta_1 + \beta_2 = 1$. The six established learning principles can be obtained as special cases of Equation (3.10a) as follows

- the ML principle if $\underline{\beta} = (0, 1, 0)$,
- the MAP principle if $\underline{\beta} = (0, 0.5, 0.5)$,
- the MCL principle if $\underline{\beta} = (1, 0, 0)$,
- the MSP principle if $\underline{\beta} = (0.5, 0, 0.5)$,
- the GDT principle if $\beta_2 = 0$, and
- the PGDT principle if $\beta_2 = 0.5$.

Although, the simplex is embedded in three dimensions, we can visualize it in two dimensions due to the constraint, $\beta_0 + \beta_1 + \beta_2 = 1$, on the weights $\underline{\beta}$, which simplifies further discussions. In Figure 3.1, we show a projection of the simplex $\underline{\beta}$ onto the (β_0, β_1)-plane and its relation to the six established learning principles.

In the rest of this subsection, we investigate the simplex $\underline{\beta}$ and its relation to other learning principles. First, we consider the axes of the simplex $\underline{\beta}$. We can write the learning principle that corresponds to the β_0-axis ($\beta_0 > 0$ and $\beta_1 = 0$) using the constraint $\beta_0 = 1 - \beta_2$ for this axis as

$$\begin{aligned}
\underline{\hat{\lambda}} &= \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C} \middle| \underline{D}, \underline{\lambda}\right) \cdot Q\left(\underline{\lambda} \middle| \underline{\alpha}\right)^{\frac{\beta_2}{\beta_0}} \right] && \text{(3.11a)} \\
&= \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C} \middle| \underline{D}, \underline{\lambda}\right) \cdot Q\left(\underline{\lambda} \middle| \underline{\alpha}\right)^{\frac{\beta_2}{1-\beta_2}} \right]. && \text{(3.11b)}
\end{aligned}$$

Similarly, we can write the learning principle corresponding to the β_1-axis (with $\beta_0 = 0$ and $\beta_1 > 0$) as

$$\begin{aligned}
\underline{\hat{\lambda}} &= \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C}, \underline{D} \middle| \underline{\lambda}\right) \cdot Q\left(\underline{\lambda} \middle| \underline{\alpha}\right)^{\frac{\beta_2}{\beta_1}} \right] && \text{(3.11c)} \\
&= \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C}, \underline{D} \middle| \underline{\lambda}\right) \cdot Q\left(\underline{\lambda} \middle| \underline{\alpha}\right)^{\frac{\beta_2}{1-\beta_2}} \right]. && \text{(3.11d)}
\end{aligned}$$

These equations state that each point on the abscissa (β_0-axis) and on the ordinate (β_1-axis) corresponds to the MSP and the MAP principle, respectively, with a weighted prior.

If the prior fulfills the condition

$$Q\left(\underline{\lambda} \middle| \underline{\alpha}\right)^{\xi} \propto Q\left(\underline{\lambda} \middle| \xi \underline{\alpha}\right) \tag{3.12}$$

for any $\xi \in \mathbb{R}^+$, each point $(1 - \beta_2, 0)$ and $(0, 1 - \beta_2)$ on the axes corresponds to either the MSP or the MAP principle using the prior $Q\left(\underline{\lambda} \middle| \frac{\beta_2}{1-\beta_2} \underline{\alpha}\right)$, respectively. In chapter *Priors* (page 34), we consider priors that fulfill the condition of Equation (3.12).

Second, we consider the lines $\beta_1 = \nu - \beta_0$ with $\nu \in [0, 1]$. As visualized in Figure 3.1, the unified generative-discriminative learning principle results in the GDT and the PGDT principle for $\nu = 1$ and $\nu = 0.5$, respectively. Using $\beta_2 \in (0, 1)$ and the condition of Equation (3.12) with $\xi = \frac{\beta_2}{1-\beta_2}$, we find that Equation (3.10a) can be written as

$$\begin{aligned}
\underline{\hat{\lambda}} &= \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C} \middle| \underline{D}, \underline{\lambda}\right)^{\frac{\beta_0}{1-\beta_2}} \cdot P\left(\underline{C}, \underline{D} \middle| \underline{\lambda}\right)^{\frac{\beta_1}{1-\beta_2}} \cdot Q\left(\underline{\lambda} \middle| \underline{\alpha}\right)^{\frac{\beta_2}{1-\beta_2}} \right] && \text{(3.13a)} \\
&= \operatorname*{argmax}_{\underline{\lambda}} \left[P\left(\underline{C} \middle| \underline{D}, \underline{\lambda}\right)^{\frac{\beta_0}{1-\beta_2}} \cdot P\left(\underline{C}, \underline{D} \middle| \underline{\lambda}\right)^{\frac{\beta_1}{1-\beta_2}} \cdot Q\left(\underline{\lambda} \middle| \frac{\beta_2}{1-\beta_2} \cdot \underline{\alpha}\right) \right]. && \text{(3.13b)}
\end{aligned}$$

The second equation is equivalent to Equation (3.9a), stating that, for each β_2, each point on the line $\beta_1 = (1 - \beta_2) - \beta_0$ corresponds to a specific instance of the PGDT principle with prior $Q\left(\underline{\lambda} \middle| \frac{\beta_2}{1-\beta_2} \cdot \underline{\alpha}\right)$. Using this result, the unified generative-discriminative learning principle allows an in-depth analysis of the PGDT principle using different priors.

Finally, we consider a second interpretation of the unified generative-discriminative learning principle. The last two terms of the Equation (3.10a) consisting of the weighted likelihood and the

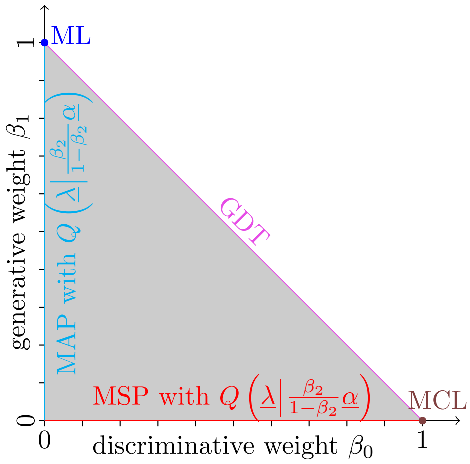

ML
GDT
MCL
MAP with $Q\left(\underline{\lambda}\middle|\frac{\beta_2}{1-\beta_2}\underline{\alpha}\right)$
MSP with $Q\left(\underline{\lambda}\middle|\frac{\beta_2}{1-\beta_2}\underline{\alpha}\right)$
generative weight β_1
discriminative weight β_0
0
1
0
1

Figure 3.2: Illustration of the unified generative-discriminative learning principle. The figure shows a projection of the simplex $\underline{\beta}$ onto the (β_0, β_1)-plane for a conjugate prior that satisfies the condition of Equation (3.12). Each point on the abscissa (β_0-axis) and ordinate (β_1-axis) refers to the MSP and the MAP principle, respectively, using the prior in a weighted version $Q\left(\underline{\lambda}\middle|\frac{\beta_2}{1-\beta_2}\underline{\alpha}\right)$. The simplex colored in gray corresponds to the MSP principle using the weighted posterior $Q\left(\underline{\lambda}\middle|\frac{\beta_1}{\beta_0}\underline{\tilde{\alpha}}\right)$ as prior for the parameter vector $\underline{\lambda}$.

weighted prior might be interpreted as a weighted posterior. Using the assumption of conjugacy (Equation (3.5)), the condition of Equation (3.12), and $\beta_0, \beta_1, \beta_2 \in \mathbb{R}^+$, we obtain

$$\hat{\underline{\lambda}} = \underset{\underline{\lambda}}{\operatorname{argmax}} \left[P\left(\underline{C}\middle|\underline{D},\underline{\lambda}\right) \cdot \left[P\left(\underline{C},\underline{D}\middle|\underline{\lambda}\right) \cdot Q\left(\underline{\lambda}\middle|\underline{\alpha}\right)^{\frac{\beta_2}{\beta_1}} \right]^{\frac{\beta_1}{\beta_0}} \right] \tag{3.14a}$$

$$\overset{(3.12)}{=} \underset{\underline{\lambda}}{\operatorname{argmax}} \left[P\left(\underline{C}\middle|\underline{D},\underline{\lambda}\right) \cdot \left[P\left(\underline{C},\underline{D}\middle|\underline{\lambda}\right) \cdot Q\left(\underline{\lambda}\middle|\frac{\beta_2}{\beta_1}\underline{\alpha}\right) \right]^{\frac{\beta_1}{\beta_0}} \right] \tag{3.14b}$$

$$\overset{(3.5)}{=} \underset{\underline{\lambda}}{\operatorname{argmax}} \left[P\left(\underline{C}\middle|\underline{D},\underline{\lambda}\right) \cdot Q\left(\underline{\lambda}\middle|\underline{\tilde{\alpha}}\right)^{\frac{\beta_1}{\beta_0}} \right] \tag{3.14c}$$

$$\overset{(3.12)}{=} \underset{\underline{\lambda}}{\operatorname{argmax}} \left[P\left(\underline{C}\middle|\underline{D},\underline{\lambda}\right) \cdot Q\left(\underline{\lambda}\middle|\frac{\beta_1}{\beta_0}\underline{\tilde{\alpha}}\right) \right] \tag{3.14d}$$

stating that each point on the simplex can be interpreted as MSP principle with an informative prior $Q\left(\underline{\lambda}\middle|\frac{\beta_1}{\beta_0}\underline{\tilde{\alpha}}\right)$ composed of the likelihood and the original prior. Interestingly, the interpretation of each point of the simplex as instance of the MSP principle using the weighted posterior as prior remains valid even for priors that do not fulfill these conditions. Figure 3.2 visualizes these results. In chapter *Comparison of learning principles* (page 50), we will apply the different learning principles to biological data.

3.1.3 Optimization

After presenting the objective functions for different learning principles, we consider the task of determining the optimal parameter vector from the objective functions.

Using the assumption of i.i.d. sequences in Equations (3.3b) and (3.4c), and the assumption of independence between the parameters of the classes, the entire optimization task for the ML and the MAP principle can be reformulated in terms of class-specific optimization tasks that allow to infer the parameters of the foreground and the background class separately

$$\log P\left(\underline{\lambda}\middle|\underline{C},\underline{D},\underline{\alpha}\right) = \sum_{n=1}^{N} \log P\left(c_n\middle|\underline{\lambda}_{\mathcal{C}}\right) + \sum_{c\in\mathcal{C}}\sum_{n=1}^{N} \delta_{c,c_n}\cdot\log P\left(\underline{x}_n\middle|c_n,\underline{\lambda}_c\right), \tag{3.15a}$$

$$\log Q\left(\underline{\lambda}\middle|\underline{\alpha}\right) = \log Q\left(\underline{\lambda}_{\mathcal{C}}\middle|\underline{\alpha}_{\mathcal{C}}\right) + \sum_{c\in\mathcal{C}} \log Q\left(\underline{\lambda}_c\middle|\underline{\alpha}_c\right), \tag{3.15b}$$

where $\underline{\lambda}_{\mathcal{C}}$ and $\underline{\alpha}_{\mathcal{C}}$ denote the parameter vector and hyper-parameter vector of the classes $\mathcal{C}$, respectively. In close analogy, $\underline{\lambda}_c$ and $\underline{\alpha}_c$ denote the parameter vector and hyper-parameter vector of class c. For several simple models like Markov models, the generative learning principles amount to computing smoothed relative frequencies of nucleotides and oligonucleotides [Staden, 1984, Stormo et al., 1982, Zhang and Marr, 1993].

While the generative learning principles often lead to analytic solutions for simple models such as Markov models, one must use numerical optimization procedures for the discriminative learning principles and thus also for the unified generative-discriminative learning principle. If the conditional likelihood, the likelihood, and the prior are log-convex functions, we can use any numerical algorithm to determine the global optimal parameter $\hat{\underline{\lambda}}$. Different numerical methods including steepest descent, conjugate gradient, quasi-Newton methods, and limited-memory quasi-Newton methods have been evaluated in [Wallach, 2002].

In close analogy to [Bouchard and Triggs, 2004], we must choose $\underline{\beta}$ a-priorily for the unified generative-discriminative learning principle, since $\underline{\beta}$ cannot be learned from the data directly via numerical optimization and it is not obvious which values should be chosen. Hence, we compute the results for a grid of given values, which allows to get an impression of the performance for the whole simplex $\underline{\beta}$.

3.2 Classification measures

For the comparison of classifiers, we need measures for evaluating their performance. During the last decade, several measures have been proposed and used in different applications. While [Baldi et al., 2000] give an excellent review about different classification measures, we concentrate on an important subset of measures for binary classifiers only. Almost all of these measures can be defined using a confusion matrix as shown in Table 3.2. The matrix contains the number of observations for each combination of predicted class and real class, defining the values *true positives, false*

		predicted class	
		positive	**negative**
real class	**positive**	$TP \equiv$ true positives	$FN \equiv$ false negatives
	negative	$FP \equiv$ false positives	$TN \equiv$ true negatives

Table 3.2: Confusion matrix for two classes.

negatives, false positives, and *true negatives*.

Based on such a confusion matrix, a number of measures has been defined:

- sensitivity: $\mathrm{Sn} = \frac{TP}{TP+FN}$
- specificity: $\mathrm{Sp} = \frac{TN}{TN+FP}$
- false positive rate: $\mathrm{fpr} = \frac{FP}{TN+FP} = 1 - \mathrm{Sp}$
- positive predictive value: $\mathrm{ppv} = \frac{TP}{TP+FP}$

For a comparison of classifiers, we have to compare the values of these measures. Often, it is not obvious which classifier is best if the values of these measures are very different, as for instance, for one classifier with a high Sn and a low Sp, and for another classifier with a low Sn and a high Sp. For this reason, one evaluates the classifiers using some constraints on the threshold T, e.g., for each classifier, we evaluate the Sn for a given Sp of 99.9% [Ben-Gal et al., 2005], the fpr for a given Sn of 95% [Castelo and Guigo, 2004], or the ppv for a given Sn of 95%. Nevertheless, comparing classifiers based only on these scalar measures might be problematic since for different values of T the results might differ.

Besides these scalar measures, two characteristic curves have been proposed for the evaluation of classifiers by varying the threshold T: on the one hand, the receiver operating characteristic (ROC) curve [Metz, 1978, Fawcett, 2004], which shows the Sn on the abscissa and the fpr on the ordinate, and on the other hand, the precision recall (PR) curve [Raghavan et al., 1989, Davis and Goadrich, 2006], which depicts the ppv on the abscissa and Sn on the ordinate. We obtain the curves by interpolating between the points obtained from varying the threshold T (Equation (3.2b)). This interpolation is done on the underlying confusion matrices for the points. While in case of the ROC curve, the interpolation between two neighboring points is linear, the interpolation for the PR curve is often non-linear [Davis and Goadrich, 2006].

The ROC curve has a long tradition in bioinformatics, whereas the PR curve becomes more and more interesting for unbalanced classification problems, i. e., if sequences of one class occur much more frequent than sequences of the other class [Sonnenburg et al., 2006, Sonnenburg et al., 2007, Abeel et al., 2009]. The visual comparison of curves is not always manageable, for instance, it might be challenging to compare a large number of classifiers. In this case, it is helpful to aggregate a curve into single scalar measure by computing the *area under the curve* (auc). A perfect classifier has the value 1 for both measures, auc-ROC as well as auc-PR. While it is often desired to have a scalar measure for an easy comparison, the auc might sometimes be misleading, since it integrates over the complete curve using also parts of the curve that are not of high interest [Sonnenburg et al., 2006].

Due to the disadvantages of the different scalar measures, often a combination of measures is used to evaluate the performance of classifiers depending on the problem at hand. For this reason, we will use subsets of these measures in chapters *Comparison of learning principles* (page 50), *Donor splice site recognition in Caenorhabditis elegans* (page 68), and *Discriminative de-novo motif discovery utilizing positional preference* (page 92) in different practical applications for assessing the general quality of the obtained classifiers.

Chapter 4

Probabilistic models for DNA sequences

In this chapter, we introduce several probabilistic models for DNA sequences, which can be used in classifiers and which will be applied in the chapters *Comparison of learning principles* (page 50), *Donor splice site recognition in Caenorhabditis elegans* (page 68), *Recognition of human transcription start sites* (page 72), *Computational reassessment of transcription factor binding site annotations* (page 82), and *Discriminative de-novo motif discovery utilizing positional preference* (page 92). Probabilistic models differ in the features they use to score a sequence. For each model $\mathcal{M}$, we provide the likelihood $P^{\mathcal{M}}\left(c, \underline{x} \middle| \underline{\lambda}\right)$ and a parameterization $\underline{\lambda}$ that can be used easily for any learning principle introduced in section *Learning principles* (page 10) of the previous chapter. In chapter *Priors* (page 34), we provide for each model the prior $Q^{\mathcal{M}}\left(\underline{\lambda} \middle| \underline{\alpha}\right)$, which allows to learn the model parameters in a Bayesian way using MAP, MSP, or the unified generative-discriminative learning principle.

We distinguish two types of models that will be described in more detail later.

- Models that are capable of scoring sequences of arbitrary length such as homogeneous Markov models. These models can be used for sequences that are no BS as for instance flanking sequences of BS.

- Models that are capable of scoring only sequences of fixed length as for instance position weight matrix models, weight array matrix models, and higher order inhomogeneous Markov models. These models are used for BS data.

Additionally, we define models that are capable scoring the length of a sequence $\underline{x}$ instead of the symbols of sequence. On the basis of these simple models, we can build more complex models, which we describe in the section *Composite models* (page 30).

4.1 Models for sequences with arbitrary length

In this section, we briefly introduce models that are capable of scoring sequences of arbitrary length. Specifically, we consider two families of models. First, we consider homogeneous Markov models, which assume that each nucleotide is conditionally statistically independent of all other nucleotides given a number of preceding nucleotides. Second, we consider cyclic Markov models, which assume that each nucleotide is conditionally statistically independent of all other nucleotides given a number of preceding nucleotides and the period in the position of the nucleotides.

For both model families, the preceding nucleotides used to score the current nucleotide are denoted as *context* and the length of the context is denoted as *order* h of the model. Furthermore, we use that the likelihood can be written as

$$P^{\mathcal{M}}\left(c,\underline{x}\middle|\underline{\lambda}\right) = P\left(c\middle|\underline{\lambda}\right) \cdot P^{\mathcal{M}}\left(\underline{x}\middle|c,\underline{\lambda}\right), \tag{4.1}$$

and we specify each model by $P^{\mathcal{M}}\left(\underline{x}\middle|c,\underline{\lambda}\right)$.

4.1.1 Homogeneous Markov models

Homogeneous Markov models (hMMs) assume that each nucleotide is conditionally statistically independent of all other nucleotides given a number of preceding nucleotides. For this reason, hMMs are often used for non-coding sequences such as upstream or downstream sequences of genes, $3'$ and $5'$ untranslated regions, non-coding exons, and introns. In the rest of this work, we denote a hMM of order h by hMM(h), which is defined by the conditional probability of sequence $\underline{x}$ given class label c and the parameter vector $\underline{\lambda}$

$$P^{\mathrm{hMM}(h)}\left(\underline{x}\middle|c,\underline{\lambda}\right) := \prod_{\ell=1}^{L} P^{\mathrm{hMM}(h)}\left(x_\ell\middle|x_{\max\{1,\ell-h\},\ldots,\ell-1},c,\underline{\lambda}\right). \tag{4.2}$$

Following [MacKay, 1998], we define each conditional distribution with context $\underline{a} \in \Sigma^k, k \in [0,h]$ and $b \in \Sigma$ using the parameter vector $\underline{\lambda}_{c,\underline{a}} := (\lambda_{c,\underline{a},A}, \lambda_{c,\underline{a},C}, \lambda_{c,\underline{a},G}, \lambda_{c,\underline{a},T})$

$$P^{\mathrm{hMM}(h)}\left(b\middle|\underline{a},c,\underline{\lambda}\right) := \frac{\exp\left(\lambda_{c,\underline{a},b}\right)}{\sum_{\tilde{b}\in\Sigma}\exp\left(\lambda_{c,\underline{a},\tilde{b}}\right)}. \tag{4.3}$$

The parameter vector $\underline{\lambda}$ of the model consists of all parameters $\underline{\lambda}_{c,\underline{a}}$ with $\underline{a} \in \Sigma^k$ and $k \in [0,h]$.

4.1.2 Cyclic Markov models

In contrast to homogeneous Markov models, cyclic Markov models (cMMs) assume that each nucleotide is conditionally statistically independent of all other nucleotides given a number of preceding nucleotides and a period in the position of the nucleotides. We denote a cMM of order h and period p by cMM(h,p). Often cMMs with period $p=3$ are used for coding DNA sequences due to the genetic code that uses tri-nucleotides, so-called codons.

As the frame of a sequence is often not known, the conditional probability of sequence $\underline{x}$ given class label c is defined as a weighted sum of all possible frames

$$P^{\text{cMM}(h,p)}\left(\underline{x}\middle|c,\underline{\lambda}\right) := \sum_{q=1}^{p} P^{\text{cMM}(h,p)}\left(q\middle|c,\underline{\lambda}\right) \cdot \prod_{\ell=1}^{L} P^{\text{cMM}(h,p)}\left(x_\ell\middle|\pi(\ell,q), x_{\max\{1,\ell-h\},\dots,\ell-1}, c, \underline{\lambda}\right), \tag{4.4}$$

where

$$\pi(\ell,q) := ((\ell+q) \bmod p) + 1, \tag{4.5}$$

denotes the current frame at position ℓ when starting in frame q and where $P^{\text{cMM}(h,p)}\left(q\middle|c,\underline{\lambda}\right)$ denotes the probability of starting in frame q. In close analogy to Equation (4.3), we define the conditional probabilities as

$$P\left(q\middle|c,\underline{\lambda}\right) := \frac{\exp\left(\lambda_{c,q}\right)}{\sum_{\tilde{q}=1}^{p}\exp\left(\lambda_{c,\tilde{q}}\right)} \tag{4.6a}$$

and

$$P\left(b\middle|q,\underline{a},c,\underline{\lambda}\right) := \frac{\exp\left(\lambda_{c,q,\underline{a},b}\right)}{\sum_{\tilde{b}\in\Sigma}\exp\left(\lambda_{c,q,\underline{a},\tilde{b}}\right)} \tag{4.6b}$$

with $q \in [1,p]$, $\underline{a} \in \Sigma^k$ and $k \in [0,h]$, and $b \in \Sigma$. The parameter vector $\underline{\lambda}$ of the model consists of all parameters $\underline{\lambda}_{c,m} := (\lambda_{c,1},\dots,\lambda_{c,p})$, and $\underline{\lambda}_{c,q,\underline{a}} := (\lambda_{c,q,\underline{a},A}, \lambda_{c,q,\underline{a},C}, \lambda_{c,q,\underline{a},G}, \lambda_{c,q,\underline{a},T})$ with $q \in [1,p]$ and $\underline{a} \in \Sigma^k$ with $k \in [0,h]$.

4.2 Models for sequences with fixed length

In this section, we introduce models that are capable of scoring only sequences of a predefined fixed length. All models that we introduce in the following are so-called *graphical models*.

Graphical models, which combine probability theory and graph theory, are statistical models where random variables are represented by nodes of a graph and in which the dependency structure of the joint probability distribution is represented by edges [Jordan, 2004]. The nodes in the graph represent random variables X_ℓ having realizations denoted by x_ℓ. Edges between nodes represent potential statistical dependencies between the corresponding random variables, while missing edges between nodes represent conditional independences of the associated random variables.

Graphical models can be categorized into *directed* acyclic graphical models called Bayesian networks and *undirected* graphical models called Markov random fields (MRFs) with a non-empty intersection called moral Bayesian networks (mBNs) [Castelo, 2002]. Many models currently used in bioinformatics belong to the family of moral Bayesian networks such as the

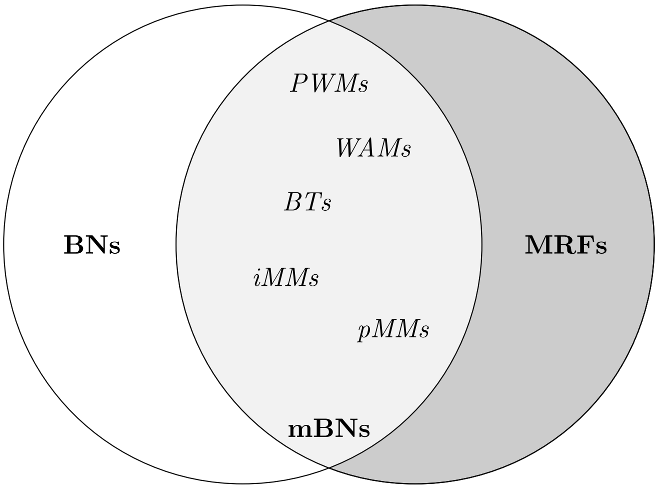


Figure 4.1: Relation of Bayesian networks and Markov random fields. The figure shows the families of Bayesian Networks (BNs) and Markov random fields (MRFs) with the non-empty intersection of moral Bayesian networks (mBNs). The intersection contains, for instance, position weight matrix (PWM) models, weight array matrix (WAM) models, Bayesian trees (BTs), as well as inhomogeneous Markov models (iMMs) and permuted Markov models (pMMs) of higher order.

position weight matrix (PWM) models [Stormo et al., 1982, Staden, 1984, Kel et al., 2003], the weight array matrix (WAM) models [Zhang and Marr, 1993, Salzberg, 1997a, Segal et al., 2006], inhomogeneous Markov models (iMMs) of higher order [Yakhnenko et al., 2005], permuted Markov models [Zhao et al., 2005], Bayesian trees [Cai et al., 2000], and their variable order extensions [Ben-Gal et al., 2005]. Nevertheless, some of the used models belong to the family of MRFs but not to the family of moral Bayesian networks [Yeo and Burge, 2004]. Figure 4.1 visualizes this situation.

For each model $\mathcal{M}$ in this section, we define its likelihood based on some non-negative scoring function $s^{\mathcal{M}}\left(c, \underline{x} \middle| \underline{\lambda}\right)$. The likelihoods are then defined as

$$P^{\mathcal{M}}\left(c, \underline{x} \middle| \underline{\lambda}\right) := \frac{s^{\mathcal{M}}\left(c, \underline{x} \middle| \underline{\lambda}\right)}{Z^{\mathcal{M}}\left(\underline{\lambda}\right)} \tag{4.7}$$

and

$$P^{\mathcal{M}}\left(\underline{x} \middle| c, \underline{\lambda}\right) := \frac{Z_c^{\mathcal{M}}\left(\underline{\lambda}\right)}{Z^{\mathcal{M}}\left(\underline{\lambda}\right)}, \tag{4.8}$$

with the partial class normalization constant

$$Z_c^{\mathcal{M}}\left(\underline{\lambda}\right) := \sum_{\underline{x}} s^{\mathcal{M}}\left(c, \underline{x} \middle| \underline{\lambda}\right), \tag{4.9}$$

where the sum runs over all possible sequences, and with the normalization constant

$$Z^{\mathcal{M}}\left(\underline{\lambda}\right) := \sum_{c \in \mathcal{C}} Z_c^{\mathcal{M}}\left(\underline{\lambda}\right). \tag{4.10}$$

Here, we use a global normalization of the score which has some desirable properties that we describe later, whereas we use a local normalization of the score for models that are capable of handling sequences of arbitrary length. For the latter models a global normalization is impossible since the global normalization constant is infinite for an infinite space of sequences $\underline{x}$. For simplicity of notation, we use $\Sigma = \{1, \ldots, |\Sigma|\}$ and $\mathcal{C} = \{1, \ldots, |\mathcal{C}|\}$ in this section.

4.2.1 Moral Bayesian networks

For Bayesian networks, the underlying structure is a directed acyclic graph (DAG). In this case, the edges are directed from the *parent* nodes to their *children*. We denote by $\underline{\mathrm{Pa}}\left(\ell\right)$ the vector of parents of node ℓ representing random variable X_ℓ, and we denote by $\underline{\mathrm{pa}}\left(\ell, \underline{x}\right)$ the realizations of the parents $\underline{\mathrm{Pa}}\left(\ell\right)$ in sequence $\underline{x}$.

In this work, we consider models with a given graph structure τ, such that all parents of each node are pre-determined, as for instance the PWM model in which each node has no parents, the WAM model in which each node has only its preceding node as parent, and iMMs of order h in which the parents of each node are the h preceding nodes. Inhomogeneous Markov models of order 0 (iMM(0)) or of order 1 (iMM(1)) are therefore a PWM model or an WAM model, respectively. To simplify notation in the following derivations, we assume the same graph structure for the models of all classes. The extension to models with different graph structures as well as to position-dependent alphabets is straightforward.

A Bayesian network is called a *moral* Bayesian network iff its DAG is moral. A DAG is said to be moral iff for each node ℓ each pair of its parents (ρ_1, ρ_2), $\rho_1 \neq \rho_2$, is connected by an edge [Castelo, 2002]. When considering the parents $\underline{\mathrm{Pa}}\left(\ell\right)$ of a node ℓ in a moral Bayesian network, we can order the nodes in $\underline{\mathrm{Pa}}\left(\ell\right)$ uniquely according to the topological ordering within the set $\underline{\mathrm{Pa}}\left(\ell\right)$. We denote a moral Bayesian network with graph structure τ by mBN(τ).

With these prerequisites, the likelihood of a directed graphical model with the commonly used parameters $\underline{\theta}$ is defined by

$$P_{\underline{\theta}}^{\mathrm{mBN}(\tau)}\left(c, \underline{x} \middle| \underline{\theta}\right) := \theta_c \cdot \prod_{\ell=1}^{L} \theta_{c,\ell,x_\ell,\underline{\mathrm{pa}}(\ell,\underline{x})}, \tag{4.11}$$

where θ_c denotes the probability of class c, and $\theta_{c,\ell,x_\ell,\underline{\mathrm{pa}}(\ell,\underline{x})}$ denotes the probability of observing x_ℓ at X_ℓ for class c given the observations $\underline{\mathrm{pa}}\left(\ell, \underline{x}\right)$ at the random variables represented by the nodes $\underline{\mathrm{Pa}}\left(\ell\right)$ [Heckerman et al., 1995]. The parameter vector $\underline{\theta}$ consists of the subvectors $\underline{\theta}_{\mathcal{C}}$ and $\underline{\theta}_{c,\ell,\underline{a}}$ for $\ell \in [1, L]$ and $\underline{a} \in \Sigma^{|\underline{\mathrm{Pa}}(\ell)|}$ where $\underline{\theta}_{\mathcal{C}} := (\theta_1, \ldots, \theta_{|\mathcal{C}|})$ and $\underline{\theta}_{c,\ell,\underline{a}} := (\theta_{c,\ell,1,\underline{a}}, \ldots, \theta_{c,\ell,|\Sigma|,\underline{a}})$. The following constraints together with the non-negativity of the θ-parameters ensure that described subvectors of

$\underline{\theta}$ remain on simplices,

$$\sum_{c \in \mathcal{C}} \theta_c = 1 \Leftrightarrow \theta_{|\mathcal{C}|} = 1 - \sum_{c=1}^{|\mathcal{C}|-1} \theta_c \tag{4.12a}$$

$$\sum_{b \in \Sigma} \theta_{c,\ell,b,\underline{a}} = 1 \Leftrightarrow \theta_{c,\ell,|\Sigma|,\underline{a}} = 1 - \sum_{b=1}^{|\Sigma|-1} \theta_{c,\ell,b,\underline{a}}. \tag{4.12b}$$

It follows from these constraints that not all parameters of $\underline{\theta}$ are free parameters: If the values of $\theta_1, \theta_2, \ldots, \theta_{|\mathcal{C}|-1}$ are given, the value of $\theta_{|\mathcal{C}|}$ is determined, and if the values of $\theta_{c,\ell,1,\underline{a}}, \theta_{c,\ell,2,\underline{a}}, \ldots, \theta_{c,\ell,|\Sigma|-1,\underline{a}}$ are given, the value of $\theta_{c,\ell,|\Sigma|,\underline{a}}$ is determined.

While for generative learning principles the parameters can be determined analytically for many statistical models including moral Bayesian networks, no analytical solution is known for most of the popular models in case of the MCL or the MSP principle, and the unified generative-discriminative learning principle. Hence, we must resort to numerical optimization techniques like conjugate gradients or second-order methods [Wallach, 2002]. Unfortunately, the parameterization of directed graphical models in terms of $\underline{\theta}$ causes two problems in case of numerical optimization:

1. The limited domain, which is $[0,1]$ for probabilities, must be assured, for instance, by barrier methods.

2. Neither the conditional likelihood $P_{\underline{\theta}}^{\mathrm{mBN}(\tau)}\left(c \middle| \underline{x}, \underline{\theta}\right)$ nor its logarithm are concave functions of $\underline{\theta}$, so numerical optimization procedures may get trapped in local maxima or saddle points [Wettig et al., 2002].

Hence, the likelihood of moral Bayesian networks is often defined in terms of alternative parameters $\underline{\lambda}$, which are closely related to the natural parameters of MRFs [Berger et al., 1996, Klein and Manning, 2003] and prevent from these problems. We define the scoring function as

$$s_{\underline{\lambda}}^{\mathrm{mBN}(\tau)}\left(c, \underline{x} \middle| \underline{\lambda}\right) := \exp\left(\lambda_c + \sum_{\ell=1}^{L} \lambda_{c,\ell,x_\ell,\underline{\mathrm{pa}}(\ell,\underline{x})}\right), \tag{4.13}$$

and we define the likelihood using Equation (4.7) as

$$P_{\underline{\lambda}}^{\mathrm{mBN}(\tau)}\left(c, \underline{x} \middle| \underline{\lambda}\right) := \frac{\exp\left(\lambda_c + \sum_{\ell=1}^{L} \lambda_{c,\ell,x_\ell,\underline{\mathrm{pa}}(\ell,\underline{x})}\right)}{Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})}. \tag{4.14}$$

Similar to the $\underline{\theta}$-parameters, we have one parameter $\lambda_c \in \mathbb{R}$ for each class $c \in \mathcal{C}$, and one parameter $\lambda_{c,\ell,b,\underline{a}} \in \mathbb{R}$ for each class c and each symbol b at X_ℓ given the observation $\underline{a}$ at random variables represented by the nodes $\underline{\mathrm{Pa}}(\ell)$. However, in contrast to $\underline{\theta}$, these parameters cannot be interpreted directly as probabilities.

As for the $\underline{\theta}$-parameters, not all parameters of $\underline{\lambda}$ are free. In case of $\underline{\lambda}$-parameters, we may fix one of the parameters in each subset, i. e., one of the λ_c and one of the $\lambda_{c,\ell,b,\underline{a}}$ for each $c \in \underline{C}$, $\ell \in [1, L]$, and $\underline{a} \in \Sigma^{|\underline{\mathrm{Pa}}(\ell)|}$ to a constant value without reducing the codomain of $s_{\underline{\lambda}}^{\mathrm{mBN}(\tau)}\left(c, \underline{x} \middle| \underline{\lambda}\right)$, resulting in the same number of free parameters for $\underline{\theta}$ and $\underline{\lambda}$. We choose to fix the last parameter in each subset arbitrarily to 0, i. e.,

$$\lambda_{|\mathcal{C}|} = 0 \qquad \text{and} \qquad \lambda_{c,\ell,|\Sigma|,\underline{a}} = 0. \tag{4.15}$$

In order to show that the likelihoods in the Equations (4.11) and (4.14) are equivalent, we need a bijective mapping from $\underline{\theta}$ to $\underline{\lambda}$. The mapping from $\underline{\theta}$ to $\underline{\lambda}$ is defined by [Meila-Predoviciu, 1999]

$$\lambda_c = \log\left(\frac{\theta_c}{\theta_{|\mathcal{C}|}}\right) \tag{4.16a}$$

and

$$\lambda_{c,\ell,b,\underline{a}} = \log\left(\frac{\theta_{c,\ell,b,\underline{a}}}{\theta_{c,\ell,|\Sigma|,\underline{a}}}\right). \tag{4.16b}$$

However, the mapping $\underline{t}$ from $\underline{\lambda}$ to $\underline{\theta}$ is non-trivial. We denote by $\theta_c := \left[\underline{t}(\underline{\lambda})\right]_c$ the component of $\underline{t}$ defining θ_c, and we denote by $\theta_{c,\ell,b,\underline{a}} := \left[\underline{t}(\underline{\lambda})\right]_{c,\ell,b,\underline{a}}$ the component of $\underline{t}$ defining $\theta_{c,\ell,b,\underline{a}}$. We obtain $\underline{t}$ by marginalization of Equation (4.14)

$$\left[\underline{t}(\underline{\lambda})\right]_c = \frac{\exp\left(\lambda_c\right) Z_c^{\mathrm{mBN}(\tau)}(\underline{\lambda})}{Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})} = \frac{\exp\left(\lambda_c\right) Z_c^{\mathrm{mBN}(\tau)}(\underline{\lambda})}{\sum_{\tilde{c}\in\mathcal{C}} \exp\left(\lambda_{\tilde{c}}\right) Z_{\tilde{c}}^{\mathrm{mBN}(\tau)}(\underline{\lambda})} \tag{4.17a}$$

and

$$\left[\underline{t}(\underline{\lambda})\right]_{c,\ell,b,\underline{a}} = \frac{\exp\left(\lambda_{c,\ell,b,\underline{a}}\right) Z_{c,\ell,b,\underline{a}}^{\mathrm{mBN}(\tau)}(\underline{\lambda})}{\sum_{\tilde{b}\in\Sigma} \exp\left(\lambda_{c,\ell,\tilde{b},\underline{a}}\right) Z_{c,\ell,\tilde{b},\underline{a}}^{\mathrm{mBN}(\tau)}(\underline{\lambda})}, \tag{4.17b}$$

where $Z_c^{\mathrm{mBN}(\tau)}(\underline{\lambda})$ and $Z_{c,\ell,b,\underline{a}}^{\mathrm{mBN}(\tau)}(\underline{\lambda})$ are two partial normalization constants defined as marginalization of Equation (4.10). While the partial class normalization constant $Z_c^{\mathrm{mBN}(\tau)}(\underline{\lambda})$ is defined in Equation (4.9), the partial node normalization constant $Z_{c,\ell,b,\underline{a}}^{\mathrm{mBN}(\tau)}(\underline{\lambda})$ of class c and node ℓ given the context $\underline{a}$ has to be defined. These partial normalization constants are defined in close analogy to the partial class normalization constant by summing over all possible realizations of the random variables in the subgraph under node ℓ given the observation $X_\ell = b$ and given the observations $\underline{\mathrm{Pa}}(\ell) = \underline{a}$. For shortness of notation, we define the partial transformation constants recursively. If node ℓ of the DAG τ of class c is a leaf, we define

$$Z_{c,\ell,b,\underline{a}}^{\mathrm{mBN}(\tau)}(\underline{\lambda}) := 1. \tag{4.18a}$$

Otherwise, following directly from the definition of a moral graph, node ℓ is parent of at least one other node k whose parents are ℓ and a subset of $\underline{\mathrm{Pa}}(\ell)$. We denote this non-empty set of nodes by

K. For each node $k \in K$ the realization of its parent nodes can be obtained directly from b and $\underline{a}$ following the definition of a moral Bayesian network. We define a specific selection function $\underline{r}_k(b, \underline{a})$ that returns the realizations of $\underline{\mathrm{Pa}}\,(k)$ given b and $\underline{a}$. The partial node normalization constant is then defined by

$$Z^{\mathrm{mBN}(\tau)}_{c,\ell,b,\underline{a}}(\underline{\lambda}) := \sum_{k \in K} \sum_{d \in \Sigma} \exp\left(\lambda_{c,k,d,\underline{r}_k(b,\underline{a})}\right) \cdot Z^{\mathrm{mBN}(\tau)}_{c,k,d,\underline{r}_k(b,\underline{a})}(\underline{\lambda}). \tag{4.18b}$$

4.2.2 Markov random fields

Markov random fields (MRFs) are undirected graphical models, i. e., the underlying structure is an undirected graph. Again, edges between nodes model potential statistical dependencies between the random variables represented by these nodes, while the absence of edges between nodes represents conditional independences of the associated random variables given their neighboring nodes. The undirected graph structure of an MRF is determined by indicator functions $\underline{f} := (\underline{f}_1, \ldots, \underline{f}_{|\mathcal{C}|})$ where $\underline{f}_c := (f_{c,1}, \ldots, f_{c,|\underline{f}_c|})$ denote the indicator functions of the class c. An indicator function $f_{c,i}(\underline{x})$ determines whether the parameter $\lambda_{c,i}$ is used for sequence $\underline{x}$ [Berger et al., 1996, Klein and Manning, 2003]. We denote an MRF with indicator functions $\underline{f}$ by MRF($\underline{f}$). The likelihood of MRF($\underline{f}$) in terms of $\underline{\lambda}$-parameters is defined by

$$P^{\mathrm{MRF}(\underline{f})}\left(c, \underline{x} \middle| \underline{\lambda}\right) = \frac{\exp\left(\lambda_c + \sum_{i=1}^{|\underline{f}_c|} \lambda_{c,i} \cdot f_{c,i}(\underline{x})\right)}{Z^{\mathrm{MRF}(\underline{f})}(\underline{\lambda})}. \tag{4.19}$$

For illustration purposes, we rewrite the likelihood of a mBN in analogy to the MRF likelihood. Hence, we rewrite the likelihood of Equation (4.14) in terms of Kronecker symbols δ,

$$P^{\mathrm{mBN}(\tau)}\left(c, \underline{x} \middle| \underline{\lambda}\right) = \frac{\exp\left(\lambda_c + \sum_{\ell=1}^{L} \sum_{b \in \Sigma} \sum_{\underline{a} \in \Sigma^{|\underline{\mathrm{Pa}}(\ell)|}} \lambda_{c,\ell,b,\underline{a}} \cdot \delta_{x_\ell,b} \delta_{\underline{\mathrm{pa}}(\ell,\underline{x}),\underline{a}}\right)}{Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})}. \tag{4.20}$$

Renaming the parameters in terms of $\lambda_{c,i}$ and defining the indicator functions $f_{c,i}$ as corresponding Kronecker symbols, we obtain the likelihood in form of Equation (4.19).

Proofing that the chosen parameterization is reasonable, we present proofs that the likelihood and the conditional likelihood of one labeled sequence are log-convex function. For the proof of the log-convexity of the likelihood, we make use of

$$Z^{\mathrm{MRF}(\underline{f})}\left(\frac{{}^1\underline{\lambda} + {}^2\underline{\lambda}}{2}\right) \leq \sqrt{\prod_{j=1}^{2} Z^{\mathrm{MRF}(\underline{f})}({}^j\underline{\lambda})}, \tag{4.21}$$

which we obtain using the Cauchy-Schwarz inequality. We obtain the log-convexity of the likelihood by showing the midpoint convexity of the logarithm of the likelihood using the vectors ${}^1\underline{\lambda}$ and ${}^2\underline{\lambda}$,

$$\log P^{\mathrm{MRF}(\underline{f})}\left(c, \underline{x} \middle| \tfrac{{}^1\underline{\lambda}+{}^2\underline{\lambda}}{2}\right) = \frac{1}{2}\sum_{j=1}^{2} {}^j\lambda_c + \sum_{i=1}^{|\underline{f}_c|} {}^j\lambda_{c,i} f_{c,i}(\underline{x}) - \log Z^{\mathrm{MRF}(\underline{f})}\left(\frac{{}^1\underline{\lambda}+{}^2\underline{\lambda}}{2}\right). \tag{4.22a}$$

Using Equation (4.21), we obtain

$$\geq \frac{1}{2}\sum_{j=1}^{2}\left[{}^j\lambda_c + \sum_{i=1}^{|\underline{f}_c|} {}^j\lambda_{c,i} f_{c,i}(\underline{x}) - \log Z^{\mathrm{MRF}(\underline{f})}\left({}^j\underline{\lambda}\right)\right] \tag{4.22b}$$

$$\geq \frac{1}{2}\sum_{j=1}^{2} \log P^{\mathrm{MRF}(\underline{f})}\left(c, \underline{x} \middle| {}^j\underline{\lambda}\right). \tag{4.22c}$$

Similarly, we present a proof for the log-convexity of the conditional likelihood.

$$\log P^{\mathrm{MRF}(\underline{f})}\left(c \middle| \underline{x}, \tfrac{{}^1\underline{\lambda}+{}^2\underline{\lambda}}{2}\right) = \frac{1}{2}\sum_{j=1}^{2}\sum_{i=1}^{|\underline{f}_c|} {}^j\lambda_{c,i} f_{c,i}(\underline{x}) - \log \sum_{\tilde{c}\in\mathcal{C}} \exp\left(\frac{1}{2}\sum_{j=1}^{2} {}^j\lambda_{\tilde{c}} + \sum_{i=1}^{|\underline{f}_{\tilde{c}}|} {}^j\lambda_{\tilde{c},i} \cdot f_{\tilde{c},i}(\underline{x})\right) \tag{4.23a}$$

Using the Cauchy-Schwarz inequality in close analogy to Equation (4.21), we obtain

$$\geq \frac{1}{2}\sum_{j=1}^{2}\left[\sum_{i=1}^{|\underline{f}_c|} {}^j\lambda_{c,i} f_{c,i}(\underline{x}) - \log \sum_{\tilde{c}\in\mathcal{C}} \exp\left({}^j\lambda_{\tilde{c}} + \sum_{i=1}^{|\underline{f}_{\tilde{c}}|} {}^j\lambda_{\tilde{c},i} \cdot f_{\tilde{c},i}(\underline{x})\right)\right] \tag{4.23b}$$

$$\geq \frac{1}{2}\sum_{j=1}^{2} \log P^{\mathrm{MRF}(\underline{f})}\left(c \middle| \underline{x}, {}^j\underline{\lambda}\right) \tag{4.23c}$$

The log-convexity of the likelihood and conditional likelihood of an MRF allow to optimize the parameters using the ML or the MCL principle by any numerical optimization algorithm without getting stuck in saddle points or local optima. In chapter *Priors* (page 34), we consider a convex prior for MRFs allowing to state the same for the MAP and the MSP principle, as well as the generative-discriminative learning principle.

4.3 Models for sequence length

In this section, we introduce models, which are capable of scoring the length of a sequence. That is, instead of using the nucleotides of a sequence only the length of the sequence is used to determine the likelihood. For this reason, we present the conditional probability $P^{\mathcal{M}}\left(\ell \middle| c\right)$ for all models of this section. Models that score the sequence length are often used to assess the distance between two features of the DNA, as for instance a TFBS and the TSS, and are therefore part of many models defined in section *Composite models* (page 30).

The simplest model capable of scoring the sequence length is a uniform distribution for a fixed interval $[L_0, L_1] \subset \mathbb{N}$ with $\Delta := L_1 - L_0 + 1 > 0$, which has been used in many de-novo motif discovery tools [Lawrence and Reilly, 1990, Bailey and Eklan, 1994, Favorov et al., 2005, Redhead and Bailey, 2007, Linhart et al., 2008]. This model is defined by the conditional probability

$$P^{\text{uni}}\left(\ell \middle| c\right) := \frac{1}{\Delta}, \tag{4.24}$$

with $\ell \in [L_0, L_1]$ and has no free parameter. For this reason, we do not give a parameter prior for this model in chapter *Priors* (page 34). Similarly, it is also possible to use any user-specified but fixed model for the sequence length to include some prior knowledge [Thompson et al., 2003].

In contrast to this simple model, it is also possible to use models capable of learning the sequence length distribution from the data, as for instance, a Gaussian distribution [Ao et al., 2004, Kim et al., 2008]. In the next subsection, we introduce a flexible model that includes the Gaussian distribution as a special case.

4.3.1 Skew normal models

The Gaussian distribution is a continuous, symmetric distribution with two parameters, which are related to the mean μ and the standard deviation σ of the distribution. In contrast, the skew normal distribution, which is an extension of the Gaussian distribution, allows to have an asymmetric distribution [Azzalini, 1985]. The density of this continuous distribution is defined as

$$d\left(\ell \middle| \mu, \sigma, \gamma\right) := \frac{1}{\sqrt{2\pi}} \exp\left(-0.5 \cdot \left[\frac{\ell - \mu}{\sigma}\right]^2\right) \cdot \Phi\left(\eta \cdot \frac{\ell - \mu}{\sigma}\right) \tag{4.25}$$

with the parameters $\mu, \eta \in \mathbb{R}$, $\sigma \in \mathbb{R}^+$, and where Φ denotes the cumulative distribution of the standard Gaussian distribution. The parameters μ and σ are related to the mean and standard deviation of the distribution, whereas η is related to the skewness of the distribution, but also affects the mean and the standard deviation. For $\eta = 0$, we obtain the Gaussian distribution with mean μ and standard deviation σ. For $\sigma \to \infty$, we obtain a uniform distribution.

Based on this density, we define a skew normal model as a discrete distribution over a fixed interval $[L_0, L_0 + \Delta - 1]$ based on the scoring function

$$s^{\text{skew}}\left(\ell \middle| \underline{\lambda}\right) := \exp\left(-0.5 \cdot g\left(\ell, \lambda_{c,0}, \exp\left(-0.5\lambda_{c,1}\right)\right)^2\right) \cdot \Phi\left(\lambda_{c,2} \cdot g\left(\ell, \lambda_{c,0}, \exp\left(-0.5\lambda_{c,1}\right)\right)\right) \tag{4.26a}$$

with the auxiliary function

$$g\left(\ell, \lambda_{c,0}, \tilde{\sigma}\right) := \frac{\ell - \left[L_0 + \Delta \cdot \left[0.01\lambda_{c,0} + \frac{\exp(\lambda_{c,0})}{1+\exp(\lambda_{c,0})}\right]\right]}{\tilde{\sigma}}, \tag{4.26b}$$

and the model parameters $\underline{\lambda} = (\lambda_{c,0}, \lambda_{c,1}, \lambda_{c,2}) \in \mathbb{R}^3$, where $\lambda_{c,0}$ is related to μ, $\lambda_{c,1}$ is related to σ, and $\lambda_{c,2}$ is related to η. The conditional probability $P^{\text{skew}}\left(\ell \middle| c\right)$ of the model is defined as

$$P^{\text{skew}}\left(\ell \middle| c, \underline{\lambda}\right) := \frac{s^{\text{skew}}\left(\ell \middle| \underline{\lambda}\right)}{Z_c^{\text{skew}}(\underline{\lambda})} \tag{4.27a}$$

with the normalization constant

$$Z_c^{\text{skew}}(\underline{\lambda}) := \sum_{l=L_0}^{L_1} s^{\text{skew}}\left(\ell \middle| \underline{\lambda}\right). \tag{4.27b}$$

4.4 Composite models

In this section, we introduce composite models, which are based on the models of the previous sections. Composite models can be used for a great variety of applications ranging from *classification of DNA sequences* over *database curation* to *de-novo motif discovery*, which we discuss in the following chapters. For each composite model, we present the conditional probability $P^{\mathcal{M}}\left(\underline{x} \middle| c, \underline{\lambda}\right)$ and the parameters $\underline{\lambda}$. For shortness of notation, we omit class c in this section yielding the more compact form $P^{\mathcal{M}}\left(\underline{x} \middle| \underline{\lambda}\right)$.

The first three models that we introduce, namely mixture model, strand model, and extended ZOOPS model, are quite similar in the way the parameters of the models are composed. For this reason, we keep the specific subsections short.

4.4.1 Mixture models

Mixture models are composite models which assume that the data of one class is generated by a number of processes instead of only one specific process. We denote a mixture model with component models $\underline{\mathcal{M}}$ by $\text{mix}(\underline{\mathcal{M}})$, which is defined by the conditional probability

$$P^{\text{mix}(\underline{\mathcal{M}})}\left(\underline{x} \middle| \underline{\lambda}\right) := \sum_{u=1}^{|\underline{\mathcal{M}}|} P^{\text{mix}(\underline{\mathcal{M}})}\left(u \middle| \underline{\lambda}_m\right) \cdot P^{\mathcal{M}_u}\left(\underline{x} \middle| \underline{\lambda}^{(\mathcal{M}_u)}\right) \tag{4.28}$$

where the probability of process u denoted by $P^{\text{mix}(\underline{\mathcal{M}})}\left(u \middle| \underline{\lambda}_m\right)$ is defined in close analogy to Equation (4.3) as

$$P^{\text{mix}(\underline{\mathcal{M}})}\left(u \middle| \underline{\lambda}_m\right) = \frac{\exp\left(\lambda_{m,u}\right)}{\sum_{i=1}^{|\underline{\mathcal{M}}|} \exp\left(\lambda_{m,i}\right)} \tag{4.29}$$

and $\underline{\lambda}$ consists of the parameters of the component probabilities $\underline{\lambda}_m := (\lambda_{m,1}, \lambda_{m,2}, \ldots, \lambda_{m,|\mathcal{M}|})$ and the parameters of the component models $\underline{\lambda}^{(\mathcal{M}_u)}$ for $u \in [1, |\mathcal{M}|]$.

4.4.2 DNA Strand models

Because TFs bind to double-stranded DNA, the strand annotation of non-palindromic BSs is important for estimating the parameters of a model. In many cases, as for instance de-novo motif discovery, this annotation is unknown and has to be learned from the data. For this reason, we define a strand model based on any motif model $\mathcal{M}$ with parameters $\underline{\lambda}^{(\mathcal{M})}$, which scores BSs on both strands as a mixture of two components. One component models the BS as located on the forward strand, whereas the other component models the BS on the reverse complementary strand, yielding the conditional probability

$$\begin{aligned} P^{\mathrm{strand}(\mathcal{M})}\left(\underline{x}\middle|\underline{\lambda}\right) &:= P^{\mathrm{strand}(\mathcal{M})}\left(u=0\middle|\underline{\lambda}_m\right)\cdot P^{\mathcal{M}}\left(\underline{x}\middle|\underline{\lambda}^{(\mathcal{M})}\right) \\ &\quad + P^{\mathrm{strand}(\mathcal{M})}\left(u=1\middle|\underline{\lambda}_m\right)\cdot P^{\mathcal{M}}\left(\underline{x}^{RC}\middle|\underline{\lambda}^{(\mathcal{M})}\right) \qquad (4.30) \\ &:= \frac{\exp\left(\lambda_{m,0}\right)}{\sum_{i=0}^{1}\exp\left(\lambda_{m,i}\right)} P^{\mathcal{M}}\left(\underline{x}\middle|\underline{\lambda}^{(\mathcal{M})}\right) + \frac{\exp\left(\lambda_{m,1}\right)}{\sum_{i=0}^{1}\exp\left(\lambda_{m,i}\right)} P^{\mathcal{M}}\left(\underline{x}^{RC}\middle|\underline{\lambda}^{(\mathcal{M})}\right), \qquad (4.31) \end{aligned}$$

where $P^{\mathrm{strand}(\mathcal{M})}\left(u=0\middle|\underline{\lambda}_m\right)$ and $P^{\mathrm{strand}(\mathcal{M})}\left(u=1\middle|\underline{\lambda}_m\right)$ are the probabilities of finding the BS on the forward or the reverse complementary strand, respectively. The model parameters $\underline{\lambda}$ are defined as $\underline{\lambda} := (\underline{\lambda}_m, \underline{\lambda}^{(\mathcal{M})})$ with $\underline{\lambda}_m := (\lambda_{m,0}, \lambda_{m,1})$.

4.4.3 Extended ZOOPS models

For de-novo motif discovery, many models and algorithms have been implemented during the last years. One widely used model is the zero or one occurrence per sequence (ZOOPS) model [Bailey and Eklan, 1994, Ao et al., 2004, Redhead and Bailey, 2007, Kim et al., 2008], which allows each sequence to contain at most one BS of one type of motif. Here, we extend the ZOOPS model by allowing different types of motifs for the BSs. We call this model extended zero or one occurrence per sequence (eZOOPS). The eZOOPS model is defined based on two hidden variables:

- The variable u_1 handles the possibility that a sequence does not contain a BS. $u_1 = 0$ denotes the case that the sequence contains no BS, and $u_1 > 0$ denotes the case that the sequence contains exactly one BS of motif type u_1. If the sequence contains one BS, it can be located at any position.
- The variable u_2 handles the start position of a BS in the sequence given that $u_1 > 0$.

For shortness of notation, we define $\underline{u} := (u_1, u_2)$. Based on any vector of *motif models* $\underline{\mathcal{M}}$ with motif lengths $\underline{w}$, any vector of *start position distributions* $\underline{\mathcal{S}}$, and any *flanking sequence model* $\mathcal{F}$ the hidden values of $\underline{u}$ lead to the conditional probability

$$P^{\mathrm{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(\underline{x}\middle|\underline{\lambda}\right) := \sum_{\underline{u}} P^{\mathrm{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(\underline{u}\middle|\underline{\lambda}\right)\cdot P^{\mathrm{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(\underline{x}\middle|\underline{u},\underline{\lambda}\right), \qquad (4.32)$$

where the sum runs over all possible values of $\underline{u}$, and where $\underline{\lambda} := (\underline{\lambda}_m, \underline{\lambda}^{(\underline{\mathcal{M}})}, \underline{\lambda}^{(\underline{\mathcal{S}})}, \underline{\lambda}^{(\mathcal{F})})$. Here, we denote the vector of motif model parameters by $\underline{\lambda}^{(\underline{\mathcal{M}})}$, and we denote the vector of position distribution parameters by $\underline{\lambda}^{(\underline{\mathcal{S}})}$. The probability $P^{\text{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(\underline{u}\middle|\underline{\lambda}\right)$ is defined as

$$P^{\text{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(\underline{u}\middle|\underline{\lambda}\right) := P^{\text{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(u_1\middle|\underline{\lambda}_m\right) \cdot P^{\text{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(u_2\middle|u_1,\underline{\lambda}^{(\underline{\mathcal{S}})}\right) \tag{4.33a}$$

with

$$P^{\text{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(u_2\middle|u_1,\underline{\lambda}^{(\underline{\mathcal{S}})}\right) := \begin{cases} 1 & \text{, if } u_1 = 0 \\ P^{\mathcal{S}_{u_1}}\left(u_2\middle|\underline{\lambda}^{(\mathcal{S}_{u_1})}\right) & \text{, otherwise} \end{cases} . \tag{4.33b}$$

If the sequence $\underline{x}$ contains no BS, i. e., if $u_1 = 0$, it is assumed that $\underline{x}$ is generated by $\mathcal{F}$

$$P^{\text{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(\underline{x}\middle|u_1 = 0,\underline{\lambda}\right) := P^{\mathcal{F}}\left(\underline{x}\middle|\underline{\lambda}^{(\mathcal{F})}\right) . \tag{4.34a}$$

If the sequence $\underline{x}$ contains a BS, i. e., if $u_1 > 0$, then it is assumed that the nucleotides upstream and downstream of the BS are generated by $\mathcal{F}$, while the BS is generated by $\mathcal{M}_{u_1}$. This yields

$$\begin{aligned} P^{\text{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(\underline{x}\middle|u_1,u_2,\underline{\lambda}\right) := {} & P^{\mathcal{F}}\left(\underline{x}_{1,\ldots,u_2-1}\middle|\underline{\lambda}^{(\mathcal{F})}\right) \\ & \cdot P^{\mathcal{M}_{u_1}}\left(\underline{x}_{u_2,\ldots,u_2+w_{u_1}-1}\middle|\underline{\lambda}^{(\mathcal{M}_{u_1})}\right) \\ & \cdot P^{\mathcal{F}}\left(\underline{x}_{u_2+w_{u_1},\ldots,L}\middle|\underline{\lambda}^{(\mathcal{F})}\right) . \end{aligned} \tag{4.34b}$$

If the BSs of motif $u_1 > 0$ can be located on both strands of the DNA, we use a strand model as motif model $\mathcal{M}_{u_1}$. If we restrict the number of motifs to one, we obtain the traditional ZOOPS model.

4.4.4 Independent product models

Another way of using more than one model for the data of one class are independent product models (IPMs). In contrast to mixture models, which assume that each sequence in the data is generated exactly by one but not necessarily the same process, independent product models assume that each sequence is generated from the same processes where each process independently generates a subsequence. For a better understanding, we illustrate independent product models using a small example. For a sequence of length 20, an independent product model can model the first 10 positions using model $\mathcal{M}_1$ and the last 10 positions using model $\mathcal{M}_2$. Of course, other partitions using more subsequences and different lengths are also possible.

An independent product model IPM$(\underline{\mathcal{M}},\underline{w},\underline{\zeta})$ is defined by a vector of models $\underline{\mathcal{M}}$, a vector of lengths $\underline{w}$, and an assignment vector $\underline{\zeta}$. The assignment vector $\underline{\zeta}$ allows to use a model for more than one subsequence. Let

$$o(i) := \sum_{j=1}^{i-1} w_{\zeta_j} = \begin{cases} 0 & \text{, if } i = 1 \\ o(i-1) + w_{\zeta_{i-1}} & \text{, if } i > 1 \end{cases} \tag{4.35}$$

be the offset for the i^{th} subsequence, then we define the conditional probability as

$$P^{\mathrm{IPM}(\underline{\mathcal{M}},\underline{w},\underline{\zeta})}\left(\underline{x}\middle|\underline{\lambda}\right) = \prod_{i=1}^{|\underline{\zeta}|} P^{\mathcal{M}_{\zeta_i}}\left(\underline{x}_{o(i)+1,\ldots,o(i)+w_{\zeta_i}}\middle|\underline{\lambda}^{(\mathcal{M}_{\zeta_i})}\right), \tag{4.36}$$

where $\underline{\lambda} := (\underline{\lambda}^{(\mathcal{M}_1)}, \ldots, \underline{\lambda}^{(\mathcal{M}_{|\mathcal{M}|})})$. Independent product models are well-suited for modeling long sequences, which are aligned to a specific signal as for instance long sequences of splice sites or transcription start sites.

Chapter 5

Priors

In this chapter, we provide for each model $\mathcal{M}$ presented in chapter *Probabilistic models for DNA sequences* (page 20) a prior $Q^{\mathcal{M}}\left(\underline{\lambda}\middle|\underline{\alpha}\right)$ that allows to learn the model parameters in a Bayesian way using the MAP principle, the MSP principle, or the unified generative-discriminative learning principle that have been presented in section *Learning principles* (page 10).

5.1 Dirichlet density

Equation (4.3) as well as the conditional probabilities of moral Bayesian networks are multinomial distributions. For this reason, almost all models of the chapter *Probabilistic models for DNA sequences* (page 20), namely hMM(h), cMM(h,p), mBN(τ), mix($\underline{\mathcal{M}}$), strand($\mathcal{M}$), eZOOPS($\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F}$), and IPM($\underline{\mathcal{M}},\underline{w},\underline{\zeta}$), use multinomial distributions. For a multinomial distribution, the Dirichlet density is a conjugate prior, which is defined by

$$\mathrm{Dir}_{\theta}\left(\underline{\theta}\middle|\underline{\alpha}\right) := \Gamma\left(\alpha.\right)\prod_{b\in\Sigma}\frac{\theta_b^{\alpha_b-1}}{\Gamma\left(\alpha_b\right)}, \tag{5.1}$$

where $\alpha. := \sum_{b\in\Sigma}\alpha_b$, $\alpha_b > 0$ for $b \in \Sigma$, and Γ denotes the gamma function.

As mentioned in subsection *Moral Bayesian networks* (page 24), the θ-parameters have two main disadvantages, namely the constraints on the parameters and the non-concavity of the conditional likelihood or its logarithm [Wettig et al., 2002]. For this reason, alternative parameterizations have been developed. In the next two subsections, we address two alternative parameterizations where we denote the hyper-parameter of each parameter by α using the same indices as for the parameter, for instance we denote the hyper-parameter for the parameter λ_b by α_b.

5.1.1 Dirichlet prior using softmax basis

For avoiding the first disadvantage, namely the constraints on the parameters, one can use the parameterization presented in Equation (4.3). Based on this parametrization and the corresponding transformation $\underline{t}$ from $\underline{\lambda}$ to $\underline{\theta}$, the Dirichlet density in Equation (5.1) is transformed into a Dirichlet density using softmax basis [MacKay, 1998]. Both densities are connected by integration via substitution,

$$Q_{\lambda}\left(\underline{\lambda}\middle|\underline{\alpha}\right) = Q_{\theta}\left(\underline{t}(\underline{\lambda})\middle|\underline{\alpha}\right)\cdot\left|\det\underline{t}'(\underline{\lambda})\right|, \tag{5.2}$$

where $\det\left(\underline{t}'(\underline{\lambda})\right)$ denotes the Jacobian of transformation function $\underline{t}$. Using Equation (5.2) and the parameterization presented in Equation (4.3), the transformed Dirichlet density is obtained as

$$\mathrm{Dir}_{\lambda}\left(\underline{\lambda}\middle|\underline{\alpha}\right) := \frac{\Gamma\left(\alpha_{\cdot}\right)}{\left[\sum_{\tilde{b}\in\Sigma}\exp\left(\lambda_{\tilde{b}}\right)\right]^{\alpha_{\cdot}}}\prod_{b\in\Sigma}\frac{\exp\left(\lambda_b\alpha_b\right)}{\Gamma\left(\alpha_b\right)}. \tag{5.3}$$

Using this transformed density, a conjugate prior of a hMM(h) in class c is composed as product for each multinomial distribution

$$Q^{\mathrm{hMM}(h)}\left(\underline{\lambda}_c\middle|\underline{\alpha}_c\right) := \prod_{k=0}^{h}\prod_{\underline{a}\in\Sigma^k}\mathrm{Dir}_{\lambda}\left(\underline{\lambda}_{c,\underline{a}}\middle|\underline{\alpha}_{c,\underline{a}}\right). \tag{5.4}$$

Analogously, a prior for cMM(h,p) is

$$Q^{\mathrm{cMM}(h,p)}\left(\underline{\lambda}_c\middle|\underline{\alpha}_c\right) := \mathrm{Dir}_{\lambda}\left(\underline{\lambda}_{c,m}\middle|\underline{\alpha}_{c,m}\right)\cdot\prod_{q=1}^{p}\prod_{k=0}^{h}\prod_{\underline{a}\in\Sigma^k}\mathrm{Dir}_{\lambda}\left(\underline{\lambda}_{c,q,\underline{a}}\middle|\underline{\alpha}_{c,q,\underline{a}}\right). \tag{5.5}$$

Furthermore, we use the prior of Equation (5.3), if the class probability is modeled separately, as for instance in hMM(h), cMM(h,p), and composite models.

Revisiting the two main disadvantages of θ-parameters, we find that there are no constraints for this parameterization allowing to use unconstrained optimization procedures. However, using the same example as [Wettig et al., 2002], we can proof that for this parameterization and mBNs still neither the conditional likelihood nor its logarithm are concave functions, and numerical optimization procedures may get trapped in local maxima or saddle points.

5.1.2 Product-Dirichlet prior for Markov random fields

The product-Dirichlet prior for moral Bayesian networks has many desirable properties, namely parameter independence, parameter modularity, likelihood equivalence [Heckerman et al., 1995], and it is intensively used for the MAP principle. Due to the lack of a suitable parameterization and the corresponding transformed product-Dirichlet prior, the product-Dirichlet prior has not yet been used for the MSP principle.

In comparative studies of different models or learning principles, many different priors have been used in the past [Ng and Jordan, 2002, Pernkopf and Bilmes, 2005, Greiner et al., 2005, Grau et al., 2007], and their choice seems arbitrary or motivated by technical aspects. Product-Gaussian and product-Laplace priors are widely used for generatively trained MRFs [Chen and Rosenfeld, 1999] and discriminatively trained MRFs [Klein and Manning, 2003, Goodman, 2003]. For the generative MAP principle applied to Markov Models and Bayesian networks, the most prevalent prior has been the product-Dirichlet prior [Heckerman et al., 1995], whereas for the discriminative MSP either a product-Gaussian or product-Laplace prior has been employed [Grau et al., 2007]. In Table 5.1, we summarize the usage of different priors found in literature showing that there is no common prior that is used for the MAP and the MSP principle as well as for mBNs and MRFs.

		learning principles	
		MAP	MSP
prior	**product-Dirichlet**	mBN	
	product-Gaussian	MRF	mBN, MRF
	product-Laplace	MRF	mBN, MRF

Table 5.1: Established priors for moral Bayesian networks and Markov random fields. While product-Gaussian and product-Laplace prior are used for MRFs for the MAP and the MSP principle, for mBNs these priors are only used for the MSP principle. The product-Dirichlet prior of mBNs, which has many desirable properties, is only used for the MAP principle.

These different priors render any conclusions regarding the superiority of one model or learning principle over the others questionable, because the *differing* influences of these priors are entirely neglected. Hence, when comparing generatively and discriminatively trained Markov models, Bayesian networks, and MRFs, in many occasions apples are compared to oranges by using different priors.

Motivated by this lack of consistency, we aim at establishing a common prior that can be used for moral Bayesian networks as well as for MRFs [Keilwagen et al., 2010b] and that can be used for all Bayesian learning principles presented in section *Learning principles* (page 10) [Keilwagen et al., 2010c]. For deriving the desired prior, we start with moral Bayesian networks using the conjugate product-Dirichlet prior, which we transform and finally generalize for the usage as a prior for MRFs.

Priors for moral Bayesian networks

For the parameter training using the Bayesian learning principles, we need to specify a prior on the parameters of the model. One conjugate prior for the likelihood of directed graphical models and their specializations is the product-Dirichlet prior $Q_\theta^{\mathrm{mBN}(\tau)}(\underline{\theta})$ [Heckerman et al., 1995]. This conjugate prior uses the assumption of parameter independence and amounts to a product of independent Dirichlet densities,

$$Q_\theta^{\mathrm{mBN}(\tau)}\left(\underline{\theta}\middle|\underline{\alpha}\right) = \mathrm{Dir}_\theta\left(\underline{\theta}_{\mathcal{C}}\middle|\underline{\alpha}_{\mathcal{C}}\right)\cdot\prod_{c\in\mathcal{C}}\prod_{\ell=1}^{L}\prod_{\underline{a}\in\Sigma^{|\underline{\mathrm{Pa}}(\ell)|}}\mathrm{Dir}_\theta\left(\underline{\theta}_{c,\ell,\underline{a}}\middle|\underline{\alpha}_{c,\ell,\underline{a}}\right). \tag{5.6}$$

We use hyper-parameters $\underline{\alpha}$ that satisfy the *consistency* condition [Buntine, 1991, Heckerman et al., 1995], which imposes the following constraints on the hyper-parameters $\underline{\alpha}$. Let $\alpha_{c,\underline{x}}$ be *joint* hyper-parameters with $\underline{x}\in\Sigma^L$ and $c\in\underline{C}$ such that for all $\ell\in[1,L]$, for all $b\in\Sigma$, and for all $\underline{a}\in\Sigma^{|\underline{\mathrm{Pa}}(\ell)|}$

$$\alpha_c := \sum_{\underline{x}\in\Sigma^L}\alpha_{c,\underline{x}} \tag{5.7a}$$

and

$$\alpha_{c,\ell,b,\underline{a}} := \sum_{\underline{x} \in \Sigma^L} \alpha_{c,\underline{x}} \cdot \delta_{x_\ell,b} \cdot \delta_{\underline{\mathrm{pa}}(\ell,\underline{x}),\underline{a}} \quad . \tag{5.7b}$$

These constraints ensure that the hyper-parameters $\underline{\alpha}$ of the product-Dirichlet prior can be interpreted as, possibly real-valued, counts stemming from a set of a-priorily observed pseudo-data. The size of the set of pseudo-data is commonly referred to as *equivalent sample size* (ESS) [Buntine, 1991, Heckerman et al., 1995], and we denote the ESS of class c by α_c.

One of our goals is to derive that prior for $\underline{\lambda}$, which is equivalent to the commonly-used product-Dirichlet prior for $\underline{\theta}$ in Equation (5.6) allowing a fair comparison of different Bayesian learning principles for mBNs based on the same prior knowledge. To this end, we use Equation (5.2) and the transformation $\underline{t}$ from $\underline{\lambda}$ to $\underline{\theta}$ given in the Equations (4.17a) and (4.17b) to transform the product-Dirichlet prior $Q_\theta^{\mathrm{mBN}(\tau)}\left(\underline{\theta}\middle|\underline{\alpha}\right)$ to the desired prior. The Jacobian of the transformation function $\underline{t}$ can be derived by exploiting independences between parameters of the model. In the following, we show the essential steps for the computation of the Jacobian.

The order of the parameters in the parameter vector has no influence on the absolute value of the Jacobian, so we choose an ordering that simplifies further computation. That is, the first parameters in the vector are the class parameters followed by the parameters of each class ordered according to the topological ordering of the corresponding nodes. Using this ordering, the transformation from $\underline{\lambda}$ to $\underline{\theta}$ for a parameter at position k of the parameter vector depends almost only on parameters at positions greater than k in the parameter vector (Equations (4.17a), (4.17b), (4.9), and (4.18b)). For this reason, we obtain zero-valued entries for almost all entries of the Jacobian matrix below the diagonal. The only non-zero entries below the diagonal are located in on-diagonal blocks. In the following, we rearrange the Jacobian matrix and especially the on-diagonal blocks to obtain an upper triangular matrix for which the determinant is simply the product of the diagonal elements. Here, we consider the first on-diagonal block for the class parameters $\lambda_1, \ldots, \lambda_{|\mathcal{C}|-1}$ and the fixed parameter $\lambda_{|\mathcal{C}|}$.

We consider the partial derivatives of $|\mathcal{C}| - 1$ parameters that build the first on-diagonal block $B(\underline{\lambda})$ of the Jacobian matrix,

$$\frac{\partial \left[\underline{t}(\underline{\lambda})\right]_c}{\partial \lambda_j} = \frac{\exp\left(\lambda_c\right) Z_c^{\mathrm{mBN}(\tau)}(\underline{\lambda})}{Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})^2} \cdot \begin{cases} Z^{\mathrm{mBN}(\tau)}(\underline{\lambda}) - \exp\left(\lambda_c\right) Z_c^{\mathrm{mBN}(\tau)}(\underline{\lambda}) & , c = j \\ -\exp\left(\lambda_j\right) Z_j^{\mathrm{mBN}(\tau)}(\underline{\lambda}) & , c \neq j \end{cases} . \tag{5.8}$$

We compute the Jacobian in two steps, which are performed on the complete Jacobian matrix. Since the results for all elements in off-diagonal blocks do not influence the determinant, we omit these elements here.

First, we subtract the first row from all other rows, and we obtain

$$
\begin{aligned}
|\det B(\underline{\lambda})| = & \prod_{c=1}^{|\mathcal{C}|-1} \frac{\exp\left(\lambda_c\right) Z_c^{\mathrm{mBN}(\tau)}(\underline{\lambda})}{Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})^2} \\
& \cdot \left| \det \begin{pmatrix} Z^{\mathrm{mBN}(\tau)}(\underline{\lambda}) - \exp(\lambda_1) Z_1^{\mathrm{mBN}(\tau)}(\underline{\lambda}) & -\exp(\lambda_2) Z_2^{\mathrm{mBN}(\tau)}(\underline{\lambda}) & \dots & -\exp\left(\lambda_{|\mathcal{C}|-1}\right) Z_{|\mathcal{C}|-1}^{\mathrm{mBN}(\tau)}(\underline{\lambda}) \\ -Z^{\mathrm{mBN}(\tau)}(\underline{\lambda}) & Z^{\mathrm{mBN}(\tau)}(\underline{\lambda}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -Z^{\mathrm{mBN}(\tau)}(\underline{\lambda}) & 0 & \dots & Z^{\mathrm{mBN}(\tau)}(\underline{\lambda}) \end{pmatrix} \right| .
\end{aligned}
\tag{5.9a}
$$

Second, we add to the first column all the other columns. We obtain an upper triangular block with $\exp\left(\lambda_{|\mathcal{C}|}\right) Z_{|\mathcal{C}|}^{\mathrm{mBN}(\tau)}(\underline{\lambda})$ as first diagonal element and $Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})$ for all $|\mathcal{C}| - 2$ other diagonal elements

$$
\begin{aligned}
|\det B(\underline{\lambda})| = & \prod_{c=1}^{|\mathcal{C}|-1} \frac{\exp\left(\lambda_c\right) Z_c^{\mathrm{mBN}(\tau)}(\underline{\lambda})}{Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})^2} \\
& \cdot \left| \det \begin{pmatrix} \exp\left(\lambda_{|\mathcal{C}|}\right) Z_{|\mathcal{C}|}^{\mathrm{mBN}(\tau)}(\underline{\lambda}) & -\exp(\lambda_2) Z_2^{\mathrm{mBN}(\tau)}(\underline{\lambda}) & \dots & -\exp\left(\lambda_{|\mathcal{C}|-1}\right) Z_{|\mathcal{C}|-1}^{\mathrm{mBN}(\tau)}(\underline{\lambda}) \\ 0 & Z^{\mathrm{mBN}(\tau)}(\underline{\lambda}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Z^{\mathrm{mBN}(\tau)}(\underline{\lambda}) \end{pmatrix} \right| .
\end{aligned}
\tag{5.9b}
$$

The determinant can now be computed as the product of the diagonal elements, and we obtain

$$
|\det B(\underline{\lambda})| = \prod_{c \in \mathcal{C}} \frac{\exp\left(\lambda_c\right) Z_c^{\mathrm{mBN}(\tau)}(\underline{\lambda})}{Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})} . \tag{5.9c}
$$

Applying these steps to all other on-diagonal blocks $B_{c,\ell,\underline{a}}(\underline{\lambda})$, we obtain

$$
|\det B_{c,\ell,\underline{a}}(\underline{\lambda})| = \prod_{b \in \Sigma} \frac{\exp\left(\lambda_{c,\ell,b,\underline{a}}\right) Z_{c,\ell,b,\underline{a}}^{\mathrm{mBN}(\tau)}(\underline{\lambda})}{\sum_{\tilde{b}} \exp\left(\lambda_{c,\ell,\tilde{b},\underline{a}}\right) Z_{c,\ell,\tilde{b},\underline{a}}^{\mathrm{mBN}(\tau)}(\underline{\lambda})} . \tag{5.10}
$$

Using Equations (5.9c) and (5.10), the Jacobian is

$$
|\det \underline{t}'(\underline{\lambda})| = \prod_{c \in \mathcal{C}} \frac{\exp\left(\lambda_c\right) Z_c^{\mathrm{mBN}(\tau)}(\underline{\lambda})}{Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})} \prod_{\ell=1}^{L} \prod_{\underline{a} \in \Sigma^{|\mathrm{Pa}(\ell)|}} \prod_{b \in \Sigma} \frac{\exp\left(\lambda_{c,\ell,b,\underline{a}}\right) Z_{c,\ell,b,\underline{a}}^{\mathrm{mBN}(\tau)}(\underline{\lambda})}{\sum_{\tilde{b}} \exp\left(\lambda_{c,\ell,\tilde{b},\underline{a}}\right) Z_{c,\ell,\tilde{b},\underline{a}}^{\mathrm{mBN}(\tau)}(\underline{\lambda})} . \tag{5.11}
$$

Based on Equation (5.2) and (5.11), and the consistency condition, many normalization constants cancel, and we obtain a simplified expression of the transformed Dirichlet prior

$$Q_{\lambda}^{\mathrm{mBN}(\tau)}\left(\underline{\lambda}\middle|\underline{\alpha}\right) \propto Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})^{-\alpha_{\cdot}} \cdot \prod_{c\in\mathcal{C}} \exp\left(\alpha_c \lambda_c\right) \cdot \prod_{\ell=1}^{L} \prod_{\underline{a}\in\Sigma^{|\underline{\mathrm{Pa}}(\ell)|}} \prod_{b\in\Sigma} \exp\left(\alpha_{c,\ell,b,\underline{a}} \lambda_{c,\ell,b,\underline{a}}\right) \tag{5.12a}$$

$$= Z^{\mathrm{mBN}(\tau)}(\underline{\lambda})^{-\alpha_{\cdot}} \cdot \exp\left(\sum_{c\in\mathcal{C}} \alpha_c\lambda_c + \sum_{\ell=1}^{L}\sum_{b\in\Sigma}\sum_{\underline{a}\in\Sigma^{|\underline{\mathrm{Pa}}(\ell)|}} \alpha_{c,\ell,b,\underline{a}}\lambda_{c,\ell,b,\underline{a}}\right) \tag{5.12b}$$

with

$$\alpha_{\cdot} = \sum_{c\in\mathcal{C}} \alpha_c \; . \tag{5.13}$$

Since the commonly-used product-Dirichlet prior for $\underline{\theta}$ defined in Equation (5.6) is conjugate to the likelihood defined in Equation (4.11), the transformed prior of Equation (5.12b) is conjugate to the likelihood defined in Equation (4.14). While in earlier comparisons of different learning principles for the same mBN, different priors have been employed, we can now use the same prior defined by Equation (5.12b) for the MAP and the MSP principle, as well as for the unified generative-discriminative learning principle. Employing this prior, we can compare the performance of two classifiers based on the same model but trained either by the MAP or the MSP principle using the identical prior, avoiding the potential bias induced by differing priors.

Choice of hyper-parameters

In contrast to the comparison of the MAP and the MSP principle for the same model, the derived prior can not be used for the unbiased comparison of different models without further premises, since these models may differ in the structure or in the number of parameters. Building on the consistency condition for the product-Dirichlet prior, we define specific joint hyper-parameters for the priors of different models representing identical sets of pseudo-data. However, even if we use identical pseudo-data as basis of these hyper-parameters, a comparison might be biased if these data contain information that can not be exploited by all models. For example, dinucleotide dependencies can be captured by a WAM model but not by a PWM model. In this case, the corresponding hyper-parameters bias the results of all models that are able to use these information. In the remainder of this paragraph, we show how to appropriately choose the hyper-parameters $\underline{\alpha}$ for an unbiased comparison of different models.

To this end, we choose hyper-parameters that represent the a-priori information that all possible sequences $\underline{x} \in \Sigma^L$ occur with equal probability in the set of pseudo-data [Buntine, 1991]. Despite the general assumption of uniform pseudo-data, the equivalent sample size may differ between the different classes $c \in \mathcal{C}$, representing a-priori class-probabilities. Using the concept of joint hyper-parameters introduced for the consistency condition earlier, this a-priori information implies that for each class c the joint hyper-parameters $\alpha_{c,\underline{x}}$ are identical for each $\underline{x}$. For this reason, we derive from

Equation (5.7a)

$$\alpha_{c,\underline{x}} = \frac{\alpha_c}{|\Sigma|^L},$$

which implies the following values of the hyper-parameters $\alpha_{c,\ell,b,\underline{a}}$ for the model parameters $\lambda_{c,\ell,b,\underline{a}}$

$$\alpha_{c,\ell,b,\underline{a}} = \frac{\alpha_c}{|\Sigma|^{1+|\underline{\mathrm{Pa}}(\ell)|}},$$

where $|\underline{\mathrm{Pa}}(\ell)|$ is the number of parents $\underline{\mathrm{Pa}}(\ell)$ of node ℓ, $c \in \underline{C}, \ell \in [1, L], b \in \Sigma$, and $\underline{a} \in \Sigma^{|\underline{\mathrm{Pa}}(\ell)|}$.

As an example, assume that we decided for equivalent sample size $\alpha_c = 32$ for class c, and we want to model the likelihood of that class either by a PWM model or by a WAM model. The PWM model has parameters $\lambda^{(\mathrm{PWM})}_{c,\ell,b}, \ell \in [1, L], b \in \Sigma$, whereas the WAM model has parameters $\lambda^{(\mathrm{WAM})}_{c,1,b}, b \in \Sigma$ and $\lambda^{(\mathrm{WAM})}_{c,\ell,b,a}, \ell \in [2, L], b, a \in \Sigma$. In case of the DNA alphabet, the hyper-parameters for the PWM model are then set to $\alpha^{(\mathrm{PWM})}_{c,\ell,b} = 8$, whereas the hyper-parameters for the WAM model are set to $\alpha^{(\mathrm{WAM})}_{c,1,b} = 8$ and $\alpha^{(\mathrm{WAM})}_{c,\ell,b,a} = 2$. With this choice of hyper-parameters, both product-Dirichlet priors represent the same amount of pseudo-data. The hyper-parameters $\alpha^{(\mathrm{PWM})}_{c,\ell,b}$ of the PWM model correspond to pseudo-counts of mono-nucleotides b, whereas the hyper-parameters $\alpha^{(\mathrm{WAM})}_{c,\ell,b,a}$ of the WAM model correspond to conditional pseudo-counts of nucleotides b observed at the current position ℓ given nucleotide a observed at the previous position $\ell - 1$. Both represent the identical a-priori information of uniform pseudo-data, since the value of $\alpha^{(\mathrm{WAM})}_{c,\ell,b,a}$ does not depend on a and all hyper-parameters fulfill the consistency condition. This result does equally hold for all specializations of Markov random fields considered in this work, and we choose the hyper-parameters accordingly throughout the case studies.

Prior for Markov random fields

Often product-Gaussian or product-Laplace priors are used for MRF$(\underline{f})$ and the Bayesian learning principles, whereas the product-Dirichlet prior is often used for moral Bayesian networks and the MAP principle. The prior of Equation (5.12b) allows an unbiased comparison of different learning principles and different models from the family of moral Bayesian networks including PWM models, WAM models, Markov models of higher order, or Bayesian trees. However, several important models proposed for the recognition of short signal sequences do not belong to this family. Hence, we now focus on generalizing this prior for the family of MRFs, which contains the family of moral Bayesian networks as special case.

Using the conformity of the parameterization of Equation (4.14) and (4.19), we suggest a prior for MRF$(\underline{f})$ in analogy to Equation (5.12b),

$$Q^{\mathrm{MRF}(\underline{f})}\left(\underline{\lambda}\middle|\underline{\alpha}\right) \propto Z^{\mathrm{MRF}(\underline{f})}(\underline{\lambda})^{-\alpha_\cdot} \cdot \exp\left(\sum_{c\in\mathcal{C}} \alpha_c\lambda_c + \sum_{i=1}^{|\underline{f}_c|} \alpha_{c,i}\lambda_{c,i}\right) \tag{5.14}$$

that contains the transformed product-Dirichlet prior of Equation (5.12b) as special case if the MRF of each class belongs to the family of moral Bayesian networks. We denote the prior of Equation (5.14) by *generalized transformed product-Dirichlet (GTPD) prior*.

Comparison of different priors

We illustrate the GTPD prior for one and two free parameters and for different values of the hyper-parameters α_i in Figures 5.1(a) and 5.1(b). Figure 5.1(a) compares the GTPD prior to the Gaussian prior and the Laplace prior for one free parameter λ_1. For illustration purposes, we choose the hyper-parameters of the Gaussian and Laplace prior such that their maxima are identical to that of the GTPD prior. We find that the GTPD prior provides an interesting interpolation between a Gaussian prior and a Laplace prior. In the vicinity of the maximum, the logarithm of the GTPD prior shows a quadratic dependence on λ_1, whereas it shows a linear dependence on λ_1 in the far tails. That is, the GTPD prior is similar to a Gaussian prior in the vicinity of the maximum and similar to a Laplace prior in the far tails.

Figure 5.1(b) shows the GTPD prior for two free parameters λ_1 and λ_2. Interestingly, the GTPD prior exhibits a mirror symmetry about the plane of $\lambda_1 = \lambda_2$ which can be explained by the choice of equal hyper-parameters $\alpha_1 = \alpha_2$. In contrast to the product-Gaussian and the product-Laplace prior, we do not find a radial symmetry for the GTPD that can be explained by the fixed parameter $\lambda_3 = 0$.

Properties

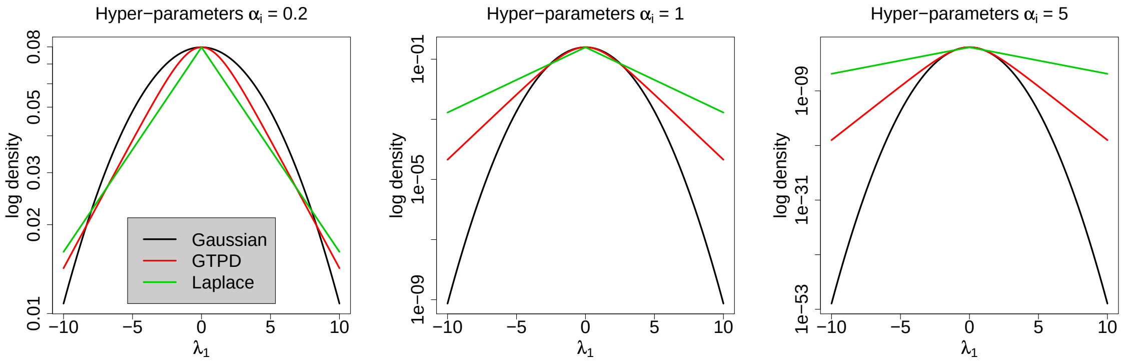
After the visual inspection of the GTPD prior for one and two free parameters, we consider some interesting properties of the prior. First, we consider whether the prior is conjugate to the likelihood of MRFs by using the i.i.d. assumption,

$$P^{\mathrm{MRF}(\underline{f})}\left(\underline{C},\underline{D}\middle|\underline{\lambda}\right)\cdot Q^{\mathrm{MRF}(\underline{f})}\left(\underline{\lambda}\middle|\underline{\alpha}\right) \tag{5.15a}$$

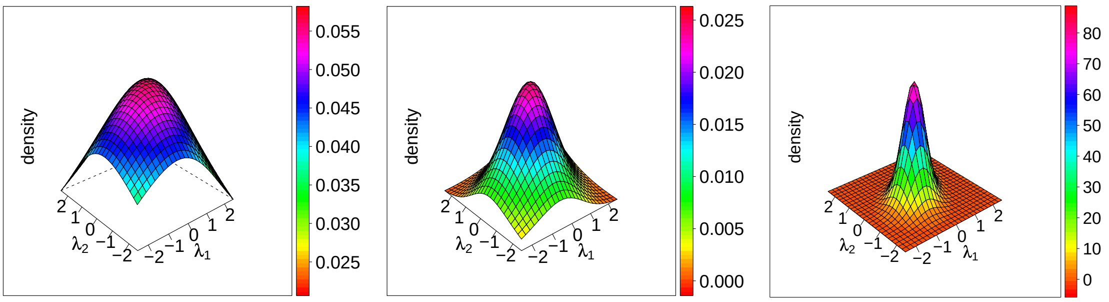
$$\propto \left[\left[Z^{\mathrm{MRF}(\underline{f})}(\underline{\lambda})\right]^{-N}\cdot\exp\left(\sum_{n=1}^{N}\lambda_{c_n}+\sum_{i=1}^{|\underline{f}_{c_n}|}\lambda_{c_n,i}\cdot f_{c_n,i}(\underline{x}_n)\right)\right]$$
$$\cdot\left[\left[Z^{\mathrm{MRF}(\underline{f})}(\underline{\lambda})\right]^{-\alpha_{\cdot}}\cdot\exp\left(\sum_{c\in\mathcal{C}}\alpha_c\lambda_c+\sum_{i=1}^{|\underline{f}_c|}\alpha_{c,i}\lambda_{c,i}\right)\right]. \tag{5.15b}$$

Rewriting the likelihood in terms of parameters instead of sequences, we obtain

$$= \left[\left[Z^{\mathrm{MRF}(\underline{f})}(\underline{\lambda})\right]^{-N}\cdot\exp\left(\sum_{c\in\mathcal{C}}N_c\lambda_{c_n}+\sum_{i=1}^{|\underline{f}_c|}N_{n,i}\lambda_{c_n,i}\right)\right]$$
$$\cdot\left[\left[Z^{\mathrm{MRF}(\underline{f})}(\underline{\lambda})\right]^{-\alpha_{\cdot}}\cdot\exp\left(\sum_{c\in\mathcal{C}}\alpha_c\lambda_c+\sum_{i=1}^{|\underline{f}_c|}\alpha_{c,i}\lambda_{c,i}\right)\right] \tag{5.15c}$$



(a) The GTPD prior (red line) for one free parameter λ_1 and $\alpha_i \in \{0.2, 1, 5\}$ on a logarithmic scale.



(b) The GTPD prior for two free parameter λ_1, λ_2 and $\alpha_i \in \{0.2, 1, 5\}$.

Figure 5.1: Illustration of the GTPD prior for one and two free parameters and 3 different hyper-parameter vectors. Figure a) shows a comparison for one free parameter λ_1 using GTPD prior (red line), Gaussian (black line), and Laplace prior (green line), and $\alpha_i \in \{0.2, 1, 5\}$, whereas figure b) shows the GTPD prior for two free parameter λ_1, λ_2 and $\alpha_i \in \{0.2, 1, 5\}$.

with $N_c := \sum_{n=1}^{N} \delta_{c_n,c}$ and $N_{c,i} := \sum_{n=1}^{N} \delta_{c_n,c} f_{c_n,i}(\underline{x}_n)$. Since both terms are now structural identical, we merge them, and we obtain

$$= \left[Z^{\mathrm{MRF}(\underline{f})}(\underline{\lambda})\right]^{-\tilde{\alpha}_{.}} \cdot \exp\left(\sum_{c\in\mathcal{C}} \tilde{\alpha}_c \lambda_c + \sum_{i=1}^{|\underline{f}_c|} \tilde{\alpha}_{c,i} \lambda_{c,i}\right) \tag{5.15d}$$

with $\tilde{\alpha}_{.} := N + \alpha_{.}$, $\tilde{\alpha}_c := N_c + \alpha_c$, and $\tilde{\alpha}_{c,i} := N_{c,i} + \alpha_{c,i}$, which is proportional to

$$\propto Q^{\mathrm{MRF}(\underline{f})}\left(\underline{\lambda} \middle| \underline{\tilde{\alpha}}\right). \tag{5.15e}$$

With this result at hand, we can state the following:

1. The GTPD prior is conjugate to the likelihood of MRFs.
2. While for moral Bayesian networks using $\underline{\theta}$ and $Q_{\theta}^{\mathrm{mBN}(\tau)}\left(\underline{\theta}\middle|\underline{\alpha}\right)$ the optimal parameters for the MAP principle with $\alpha_{.} \to 0$ are not equal to the parameters obtained from the ML principle, for $\underline{\lambda}$ and the GTPD prior this equality holds.
3. The likelihood equivalence for mBNs states "that data should not help to discriminate network structures that represent the same assertions of conditional independence" [Heckerman et al., 1995]. That is, each two mBNs with DAGs encoding the same conditional independences return the same likelihood for ML parameters. Using parameters $\underline{\lambda}$ and the GTPD prior with the consistency condition, i. e., the prior represents a pseudo-data set $\underline{D}_1$ with class labels $\underline{C}_1$, the MAP solution for a mBN and a data set $\underline{D}_2$ with class labels $\underline{C}_2$ is equivalent to the ML solution for the combined data set and class labels. For this reason, the likelihood equivalence holds for MAP parameters using $\underline{\lambda}$ and $Q_{\lambda}^{\mathrm{mBN}(\tau)}\left(\underline{\lambda}\middle|\underline{\alpha}\right)$, which is again not true for $\underline{\theta}$.

Second, the prior of Equation (5.14) obviously fulfills the condition of Equation (3.12) which allows to interpret the unified generative-discriminative learning principle using this prior for MRFs as illustrated in Figure 3.2. For the GTPD prior, we can interpret the condition of Equation (3.12) as multiplication of the initially chosen ESS by a factor which results in a *virtual* ESS. This allows to interpret the axes as MSP or MAP principle using different ESS but using the same ratio between the hyper-parameters. In the interpretation of the complete simplex $\underline{\beta}$, the hyper-parameters $\underline{\tilde{\alpha}}$ are

$$\tilde{\alpha}_c := N_c + \frac{\beta_2}{\beta_1}\alpha_c \tag{5.16a}$$

and

$$\tilde{\alpha}_{c,i} := N_{c,i} + \frac{\beta_2}{\beta_1}\alpha_{c,i}. \tag{5.16b}$$

Third, we show that the prior is a log-convex function by showing the midpoint convexity of the logarithm of the prior using the vectors ${}^{1}\underline{\lambda}$ and ${}^{2}\underline{\lambda}$ and the proportionality constant $\mathcal{B}(\underline{\alpha})$ of Equation (5.14), which in case of a moral Bayesian network is a quotient of gamma functions,

$$\log Q^{\mathrm{MRF}(\underline{f})}\left(\frac{{}^{1}\underline{\lambda}+{}^{2}\underline{\lambda}}{2}\middle|\underline{\tilde{\alpha}}\right) = \log\mathcal{B}(\underline{\alpha}) - \alpha_{.}\log Z^{\mathrm{MRF}(\underline{f})}\left(\frac{{}^{1}\underline{\lambda}+{}^{2}\underline{\lambda}}{2}\right) + \frac{1}{2}\left[\sum_{j=1}^{2}\sum_{c\in\mathcal{C}}\alpha_c{}^{j}\lambda_c + \sum_{i=1}^{|\underline{f}_c|}\alpha_{c,i}{}^{j}\lambda_{c,i}\right] \quad . \tag{5.17a}$$

Using Equation (4.21), we obtain

$$\geq \frac{1}{2}\left[\sum_{j=1}^{2} \log \mathcal{B}(\underline{\alpha}) - \alpha_{.} \log Z^{\mathrm{MRF}(\underline{f})}({}^{j}\underline{\lambda}) + \sum_{c\in\mathcal{C}} \alpha_c {}^{j}\lambda_c + \sum_{i=1}^{|\underline{f}_c|} \alpha_{c,i} {}^{j}\lambda_{c,i}\right] \tag{5.17b}$$

$$\geq \frac{1}{2}\sum_{j=1}^{2} \log Q^{\mathrm{MRF}(\underline{f})}\left({}^{j}\underline{\lambda}\middle|\underline{\tilde{\alpha}}\right), \tag{5.17c}$$

stating that the prior is a log-convex function. The log-convexity of the GTPD prior in combination with the log-convexity of the likelihood and the conditional likelihood presented in the Equations (4.22) and (4.23), respectively, allow to optimize the parameters of MRFs using the unified generative-discriminative learning principle by any numerical optimization algorithm without getting stuck in saddle points or local optima. Since the MAP and the MSP principle are special cases of the unified generative-discriminative learning principle, the same holds true for these two learning principles.

Finally, the GTPD prior is equivalent to the conjugate prior of the exponential family [Bishop, 2006] for the studied family of models.

5.2 Prior of skew normal models

As prior for the model parameters of the skew normal models, we choose a product of three independent parts.

- For the first parameter λ_0, which is related to the mean of the distribution, we choose a transformed Gaussian distribution,

$$Q_0^{\mathrm{skew}}\left(\lambda_0\middle|\underline{\alpha}_0\right) \propto \exp\left(-0.5\cdot g\left(\alpha_{0,0}, \lambda_0, \alpha_{0,1}\right)^2\right)\cdot\left[0.01 + \frac{\exp\left(\lambda_0\right)}{\left[1+\exp\left(\lambda_0\right)\right]^2}\right] \quad . \tag{5.18a}$$

- For the second parameter λ_1, which is related to the standard deviation of the distribution, we choose a transformed Gamma distribution,

$$Q_1^{\mathrm{skew}}\left(\lambda_1\middle|\underline{\alpha}_1\right) \propto \exp\left(\alpha_{1,0}\lambda_1 - \alpha_{1,1}\exp\left(\alpha_1\right)\right) \quad . \tag{5.18b}$$

- For the third parameter λ_2, which is related to the skew of the distribution, we choose a Gaussian distribution,

$$Q_2^{\mathrm{skew}}\left(\lambda_2\middle|\underline{\alpha}_2\right) \propto \exp\left(-0.5\cdot\left(\frac{\lambda_2-\alpha_{2,0}}{\alpha_{2,1}}\right)^2\right) \quad . \tag{5.18c}$$

The complete prior of the model composes as

$$Q^{\text{skew}}\left(\underline{\lambda}\middle|\underline{\alpha}\right) \propto \prod_{i=0}^{2} Q_i^{\text{skew}}\left(\lambda_i\middle|\underline{\alpha}_i\right), \tag{5.19}$$

with $\underline{\alpha} = (\underline{\alpha}_0, \underline{\alpha}_1, \underline{\alpha}_2)$ and $\underline{\alpha}_i = (\alpha_{i,0}, \alpha_{i,1})$

5.3 Prior of composite models

For composite models, we use composite priors which consist of the prior of each component model and the prior for the remaining parameters of the model. Since the proposed priors are very similar, we keep this section short.

- For a mixture model mix($\underline{\mathcal{M}}$), we compose the prior of a transformed Dirichlet prior (Equation (5.3)) for $\underline{\lambda}_m$ and the priors for the component models. The composed prior is

  $$Q^{\text{mix}(\underline{\mathcal{M}})}\left(\underline{\lambda}\middle|\underline{\alpha}\right) = \text{Dir}_\lambda\left(\underline{\lambda}_m\middle|\underline{\alpha}_m\right) \cdot \prod_{u=1}^{|\underline{\mathcal{M}}|} Q^{\mathcal{M}_u}\left(\underline{\lambda}^{(\mathcal{M}_u)}\middle|\underline{\alpha}^{(\mathcal{M}_u)}\right), \tag{5.20}$$

  where $\underline{\alpha}_m := (\alpha_\cdot^{(\mathcal{M}_1)}, \ldots, \alpha_\cdot^{(\mathcal{M}_{|\mathcal{M}|})})$ and $\underline{\alpha}$ consists of the hyper-parameters for the component probabilities $\underline{\alpha}_m$ and the hyper-parameters of the component models $\underline{\alpha}^{(\mathcal{M}_u)}$. We denote the ESS of the mixture model by $\alpha_\cdot = \sum_{m=1}^{|\underline{\mathcal{M}}|} \alpha_\cdot^{(\mathcal{M}_m)}$.

- For a strand model strand($\mathcal{M}$), we compose the prior of a transformed Dirichlet prior (Equation (5.3)) for $\underline{\lambda}_m$ and the prior for the component model. The composed prior is

  $$Q^{\text{strand}(\mathcal{M})}\left(\underline{\lambda}\middle|\underline{\alpha}\right) = \text{Dir}_\lambda\left(\underline{\lambda}_m\middle|\underline{\alpha}_m\right) \cdot Q^{\mathcal{M}}\left(\underline{\lambda}^{(\mathcal{M})}\middle|\underline{\alpha}^{(\mathcal{M})}\right), \tag{5.21}$$

  where $\underline{\alpha} := (\underline{\alpha}_m, \underline{\alpha}^{(\mathcal{M})})$, $\underline{\alpha}_m := (\alpha_{m,0}, \alpha_{m,1})$, and the ESS of the strand model $\alpha_\cdot = \alpha_{m,0} + \alpha_{m,1}$ has to be equal to the ESS of the component model $\alpha_\cdot^{(\mathcal{M})}$.

- For an extended zero or one occurrence per sequence model eZOOPS($\underline{\mathcal{M}}, \underline{\mathcal{S}}, \mathcal{F}$), we compose the prior of a transformed Dirichlet prior (Equation (5.3)) for $\underline{\lambda}_m$ and the priors for the component models. The composed prior is

  $$\begin{aligned} Q^{\text{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(\underline{\lambda}\middle|\underline{\alpha}\right) = {} & \text{Dir}_\lambda\left(\underline{\lambda}_m\middle|\underline{\alpha}_m\right) \cdot Q^{\mathcal{F}}\left(\underline{\lambda}^{(\mathcal{F})}\middle|\underline{\alpha}^{(\mathcal{F})}\right) \\ & \cdot \prod_{u=1}^{|\underline{\mathcal{M}}|} Q^{\mathcal{M}_u}\left(\underline{\lambda}^{(\mathcal{M}_u)}\middle|\underline{\alpha}^{(\mathcal{M}_u)}\right) Q^{\mathcal{S}_u}\left(\underline{\lambda}^{(\mathcal{S}_u)}\middle|\underline{\alpha}^{(\mathcal{S}_u)}\right), \end{aligned} \tag{5.22}$$

  where $\underline{\alpha}$ consists of the hyper-parameters for the component probabilities $\underline{\alpha}_m$ and the hyper-parameters of the component models $\underline{\alpha}^{(\mathcal{M}_u)}$, $\underline{\alpha}^{(\mathcal{S}_u)}$, and $\underline{\alpha}^{(\mathcal{F})}$.

- For an independent product model IPM$(\underline{\mathcal{M}}, \underline{w}, \underline{\zeta})$, we compose the prior of the priors for the component models. The composed prior is

  $$Q^{\mathrm{IPM}(\underline{\mathcal{M}},\underline{w},\underline{\zeta})}\left(\underline{\lambda}\middle|\underline{\alpha}\right) = \prod_{i=1}^{|\underline{\mathcal{M}}|} Q^{\mathcal{M}_i}\left(\underline{\lambda}^{(\mathcal{M}_i)}\middle|\underline{\alpha}^{(\mathcal{M}_i)}\right), \tag{5.23}$$

  where $\underline{\alpha}$ consists of the hyper-parameters of the component models $\underline{\alpha}^{(\mathcal{M}_i)}$.

Chapter 6

Implementation

After considering learning principles, probabilistic models, and priors in the theoretical part of this work, we now turn to the practical part of this work. We implement an open-source Java framework called Jstacs that provides implementation for statistical analysis and classification of biological sequences. Jstacs is strictly object-oriented and structured in several packages. Furthermore, Jstacs includes all implementations that are used for this work including for instance implementations for data handling, classifiers, learning principles, and probabilistic models.

Jstacs comprises an efficient representation and convenient handling of sequence data, provides ready-to-use implementations of many statistical models for sequence data, methods for evaluating the performance measures described in section *Classification measures* (page 17), and includes possibilities for saving and loading objects in XML-format. Jstacs is capable of handling a great variety of data and is not only restricted to DNA sequences. Each sequence is represented by an instance of the abstract class `Sequence`. Data sets are called `Samples` in Jstacs and consist of a number of `Sequences`. Models can be combined to constitute classifiers, which can be trained and assessed using Jstacs. For comparing different classifiers on test data sets or by hold-out experiments, Jstacs comes with a number of performance measures, which can be selected by the user. Hence, Jstacs allows to solve complex problems efficiently with only few lines of code. We provide some examples in the documentation section of the Jstacs homepage `www.jstacs.de`. In addition, we present a simplified part of the Jstacs class hierarchy in Figure 6.1 and a list of classes frequently used in this work in Table 6.1.

Jstacs provides two possibilities for implementing probabilistic models. On the one hand, the interface `Model` and its abstract class `AbstractModel`, which implements `Model` providing the implementation of several methods declared in `Model`, can be used for any probabilistic model that should be trained only in a generative way. On the other hand, Jstacs provides the interface `NormalizableScoringFunction` for any probabilistic model that might be learned using the unified generative-discriminative learning principle. Similar to `Model` and `AbstractModel`, `AbstractNormalizableScoringFunction` implements `NormalizableScoringFunction` and provides the implementation of several methods declared in `NormalizableScoringFunction`.

In Jstacs, we distinguish these two possibilities of implementing probabilistic models, since generative parameter learning often can be done analytically, while we have to use numerical methods for discriminative and hybrid parameter learning. However, the interfaces `Model` and `NormalizableScoringFunction` enable the user to implement any probabilistic models for both possibilities at once. In addition, the class `NormalizableScoringFunctionModel` allows to use each `NormalizableScoringFunction` as a `Model`.

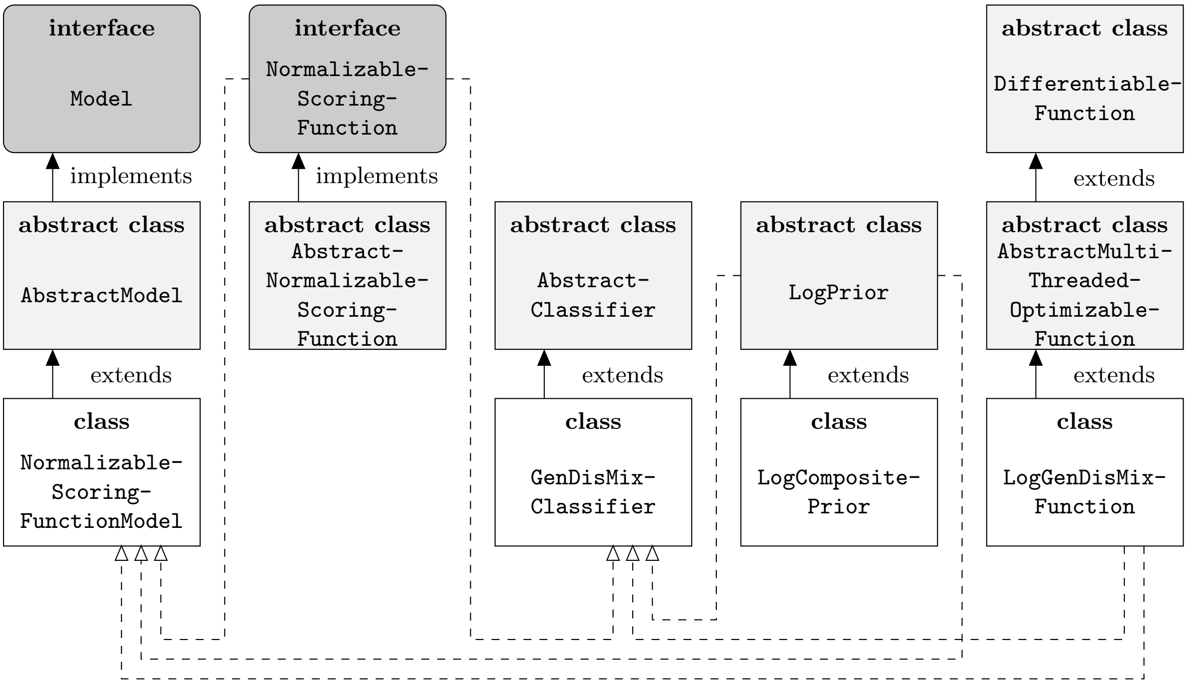


Figure 6.1: Simplified part of the Jstacs class hierarchy. Interfaces are visualized by dark gray, rounded rectangles, abstract classes by light gray rectangles, and classes by white rectangles. Solid lines indicate the hierarchy of inheritance, i. e., which class implements which interface or extends which abstract class. The dashed lines indicate that instances of these classes are used by instances of other classes. For reason of clarity, we do not show any implementations of `AbstractNormalizableScoringFunction` and refer to Table 6.1 in this case.

Here, we consider only the latter possibility of implementing probabilistic models using `NormalizableScoringFunction`. In contrast to generative parameter learning, we have to specify the complete classifier for learning the parameters using discriminative or hybrid learning principles. In Jstacs, the class `GenDisMixClassifier`, which extends the abstract class `AbstractClassifier`, is used to specify a classifier using the unified generative-discriminative learning principle for parameter learning. Instantiating an object of this class, we have to specify a `NormalizableScoringFunction` for each class, the prior that is an instance of the abstract class `LogPrior`, and the weights $\underline{\beta}$ for the unified generative-discriminative learning principle besides several parameters for the numerical optimization. The GTPD prior for MRFs is implemented by the class `CompositeLogPrior`, which extends `LogPrior`.

A `GenDisMixClassifier` can be trained on some `Samples` using one of the `train`-methods declared in `AbstractClassifier`. However, internally the `GenDisMixClassifier` creates an instance of `LogGenDisMixFunction`, which is used in the numerical optimization accomplished by the class `Optimizer`. `LogGenDisMixFunction` is an extension of the abstract class `AbstractMultiThreadedOptimizableFunction`, which itself is an extension of the abstract class

Description	**Jstacs class**
DNA data set	`DNASample`
classifier using the unified generative-discriminative learning principle (Equation (3.10a))	`GenDisMixClassifier`
homogeneous Markov model	`HMMScoringFunction`
cyclic Markov model	`CMMScoringFunction`
moral Bayesian network	`BayesianNetworkScoringFunction`
Markov random field	`MRFScoringFunction`
mixture model	`MixtureScoringFunction` `VariableLengthMixtureScoringFunction`
strand model	`StrandScoringFunction`
eZOOPS model	`HiddenMotifsMixture`
independent product model	`IndependentProductScoringFunction`

Table 6.1: Main classes of Jstacs used in this work.

`DifferentiableFunction`. The abstract class `DifferentiableFunction` declares a method for evaluating the gradient of the function that is used during numerical optimization. The abstract class `AbstractMultiThreadedOptimizableFunction` provides the possibility to evaluate a function and its gradients using a user-specified number of threads using the i.i.d. assumption for the data. This allows to exploit the compute power of modern multi-core computers.

Due to its strictly object-oriented design, which provides many interfaces and abstract classes, Jstacs is readily extensible. For instance, implementing a new probabilistic model by extending `NormalizableScoringFunction` enables us to use instances of this class in combination with other existing models in more complex models, as for instance, `MixtureScoringFunction` and `IndependentProductScoringFunction` as well as in classifiers, as for instance, the `GenDisMixClassifier`. Similarly, implementing a new learning principle, enables us to use it on any `NormalizableScoringFunction`. In addition, the class `AbstractMultiThreadedOptimizableFunction` enables to implement new learning principles in parallel with almost no implementation overhead.

Furthermore, Jstacs provides classes for using BioJava [Holland et al., 2008] as well as R [R Development Core Team, 2009]. Jstacs is easy to use and is publicly available at the Jstacs homepage `www.jstacs.de` and at the machine-learning open source software (mloss) repository `www.mloss.org`, and requires Java Runtime Environment (JRE)[1] of at least version 5. A complete API, several code examples, as well as a forum are also available at the Jstacs homepage.

---

[1] `www.sun.com/java`

Chapter 7

Comparison of learning principles

In this chapter, we use the probabilistic models presented in chapter *Probabilistic models for DNA sequences* (page 20) and the learning principles presented in section *Learning principles* (page 10) with the priors presented in chapter *Priors* (page 34) using the Jstacs library. We investigate whether the generative, discriminative, or hybrid learning principles are well suited for two important tasks of DNA sequence analysis. On one hand, we consider the classification of TFBSs, and on the one hand, we consider the classification of splices sites. As mentioned earlier, both tasks are very important for our understanding of gene regulation and gene function.

This chapter is threepart. First, we compare two non-Bayesian learning principles, namely the generative ML and the discriminative MCL principle, for the recognition of splice sites. Second, we use a prior with specific hyper-parameters, and compare two Bayesian learning principles, the MAP and the MSP principle, for splice sites as well as for TFBSs. Finally, we use the unified generative-discriminative learning principle, which enables us to investigate different aspects of the learning principles. Here, we focus on two aspects. On the one hand, we investigate whether the hybrid learning principles might help to improve the performance of classifiers by estimating parameters that might be better suited for classification. On the other hand, we investigate the MAP and the MSP principle when varying the strength of the prior. This can be systematically achieved by comparing two edges of the simplex spanned by the weights $\underline{\beta}$ of the unified generative-discriminative learning principle (Equations (3.11b) and (3.11d), and Figure 3.2).

7.1 Comparison of the ML and the MCL principle

For the comparison of the ML and the MCL principle [Keilwagen et al., 2007], we choose the human splice site data sets that are already split into training and test data set [Yeo and Burge, 2004]. All donor splice sites in this data set contain a consensus `GT`, while all acceptor splice sites contain a consensus `AG`. These canonical dinucleotides can be found in more than 98% of mammalian introns [Burset et al., 2000], thus constituting the most important class of splice sites. After removing the consensus dinucleotide for both kinds of splice sites, we obtain sequences of lengths 7 bp and 21 bp for donors and acceptors, respectively. In Table 7.1, we summarize the number of sequences for the data sets [Yeo and Burge, 2004].

Based on the performance measures used in [Yeo and Burge, 2004], an MRF that captures all pairwise dependencies between nucleotides that are at most 5 bp apart has been proposed for the recognition of splice sites. Following [Yeo and Burge, 2004], we denote this MRF as MRF(me2x5).

data set		**donor splice sites**	**acceptor splice sites**
train	**real**	8,415	8,465
	decoy	179,438	180,957
test	**real**	4,208	4,233
	decoy	89,717	90,494

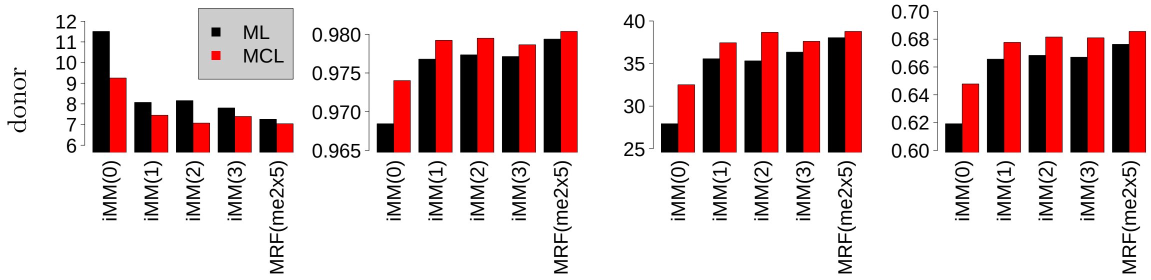
Table 7.1: Size of splice site data sets of [Yeo and Burge, 2004]. Each entry shows the number of sequences for the specific data set.

Based on the presented models in [Yeo and Burge, 2004], we compare iMMs of different order ranging from order 0 to order 3 with MRF(me2x5) for the ML and the MCL principle. Based on the predefined splits [Yeo and Burge, 2004], we train the classifiers on about two thirds of the data and evaluate the performance of the classifiers on the remaining third of the data (Table 7.1). For the evaluation of the classifiers, we choose the four performance measures fpr for a fixed Sn of 95%, the auc-ROC, the ppv for a fixed Sn of 95%, and the auc-PR (section *Classification measures* (page 17)).

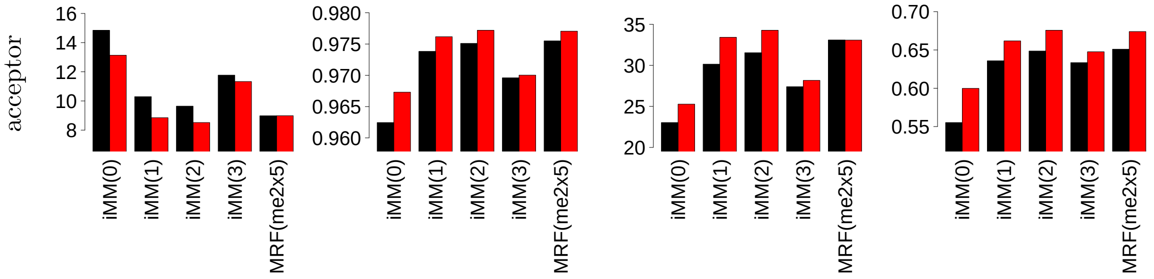
In Figure 7.1, we visualize the results obtained from ML trained and MCL trained classifiers. First, we consider the results for donor splice sites in subfigures 7.1a-d. For generatively trained classifiers visualized by black bars, we find that the results of an iMM(0) can be strongly improved by an iMM(1) for all four performance measures, while the performance is comparable for generatively trained classifiers based on iMM(1), iMM(2), and iMM(3). We obtain the best results for donor splice sites and a generatively trained classifier based on an MRF(me2x5), which yields a fpr of 7.3%, an auc-ROC of 0.979, a ppv of 38.0%, and an auc-PR of 0.676. Turning to discriminatively trained classifiers visualized by gray bars, we find a similar behaviour. The classifier based on iMM(0) performs worst, while the classifiers based on iMM(1), iMM(2), and iMM(3) perform comparable based on the four performance measures. The best results are again obtained for a classifier based on MRF(me2x5), which yields a fpr of 7.0%, an auc-ROC of 0.980, a ppv of 38.8%, and an auc-PR of 0.686.

Comparing generatively and discriminatively trained classifiers based on the same models, we find that in all cases the discriminatively trained classifier outperforms its generative counterpart. Consequently, we find as best classifier a discriminatively trained classifier based on MRF(me2x5). Scrutinizing the results for classifiers based on iMM(1), iMM(2), iMM(3), and MRF(me2x5), we find that choosing the discriminative MCL principle instead of the generative ML principle is of same importance as choosing the best of these four models.

Second, we consider the results for acceptor splice sites in subfigures 7.1e-h. We find an increasing performance for all four performance for the generatively trained classifiers based on iMM(0) to iMM(2), while we find less good results for the generatively trained classifier based on iMM(3). This decrease might be caused by overfitting, i. e., overadaption of the classifier to the training data. Similar to donor splice sites, we find that the classifier based on MRF(me2x5) performs best among the generatively trained classifiers yielding a fpr of 9.0%, an auc-ROC of 0.976, a ppv of 33.1%, and an auc-PR of 0.651. Considering the discriminatively trained classifiers, we find a similar behaviour. For iMMs with order 0 to 2 the performance is increasing, while it is decreasing for order 3. Considering



(a) False positive rate (b) auc-ROC (c) Positive predictive value (d) auc-PR



(e) False positive rate (f) auc-ROC (g) Positive predictive value (h) auc-PR

Figure 7.1: Comparison of classification performance for donor and acceptor splice sites based on the ML and the MCL principle. In each subfigure, we plot the performance measure on the ordinate and the models on the abscissa. Black bars indicate the results of ML-trained classifiers, while red bars indicate the performance of MCL-trained classifiers. In the subfigures a-d, we show the results for donor splice sites, while we show the results for acceptor splice sites in subfigures e-h.

the classifier based on MRF(me2x5), we find that it performs slightly worse than the discriminatively trained classifier based on iMM(2), which yields a fpr of 8.5%, an auc-ROC of 0.977, a ppv of 34.3%, and an auc-PR of 0.676.

In close analogy to the comparison of discriminatively and generatively trained classifiers for donor splice sites, we compare classifiers for acceptor splice sites based on the same models but trained either generatively or discriminatively. We find that for all iMMs the performance of the discriminatively trained classifier is better than for its generative counterpart. For the MRF(me2x5), we find that the discriminatively trained classifier performs at least as good as the generatively trained classifier. Hence, we find as best classifier a discriminatively trained classifier based on iMM(2). Scrutinizing the results for iMM(1), iMM(2), and MRF(me2x5), we find that choosing the discriminative MCL principle instead of generative ML principle is of same importance as choosing the best of these three models.

Concluding, we find for splice site data that classifiers trained using the MCL principle outperform classifiers trained by the ML principle. This is in accordance with previous findings [Ng and Jordan, 2002] that state that discriminatively trained classifiers have a smaller asymptotic error and therefore perform better than their generative counterparts if the training data set is large. For splice sites, typical data sets are large containing thousands of sequences (Table 7.1) enabling to use the power of the MCL principle. In addition, we find that choosing an appropriate learning principle often is similar important as choosing an appropriate model.

7.2 Comparison of the MAP and the MSP principle

After investigating the benefits of the ML and the MCL principle, we now investigate the Bayesian learning principles MAP and MSP [Keilwagen et al., 2010b]. These two learning principles incorporate prior knowledge on the parameters of the classifier by using some prior density. In this section, we compare classifiers trained either using the generative MAP or the discriminative MSP principle employing the same prior and the same hyper-parameters. In case study 1, we continue the case study of the previous section investigating donor splice sites, while in case study 2, we consider the recognition of TFBSs for different sizes of training data sets.

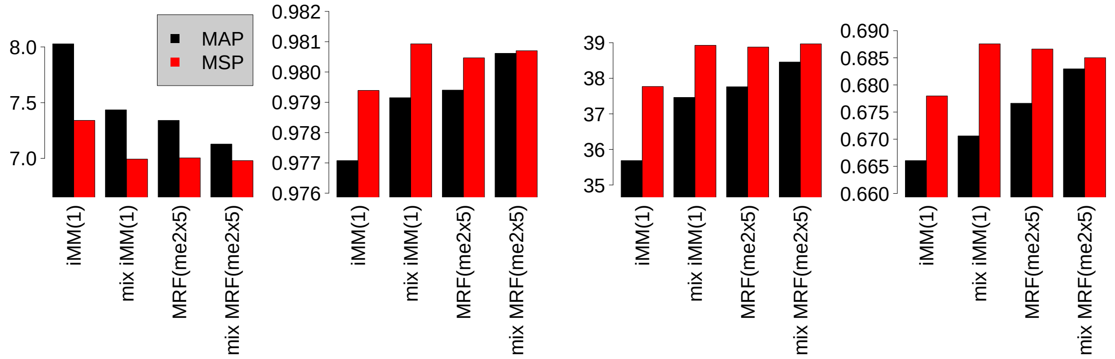
7.2.1 Case study 1: Mixture models for donor splice sites

In this case study, we compare classifiers based on different models from the family of MRFs trained either by the MAP or by the MSP principle. For both learning principles, we need a prior on the parameters of the models, which we choose to be the GTPD prior. To avoid any bias in the hyper-parameters of the prior, we choose a prior that represents uniform pseudo-data with an ESS of 32 for the foreground data set and an ESS of 96 for the background data set. We again choose the standard data set compiled by Yeo & Burge [Yeo and Burge, 2004] but restrict ourselves to donor splice sites.

Following [Yeo and Burge, 2004] and the case study in the previous section, we choose the models iMM(1) and MRF(me2x5). Based on these basic models, we also use two-component mixture models of these models. On one hand, we use a mixture of two iMM(1) denoted as mix iMM(1), and on the other hand, we use a mixture of two MRF(me2x5) denoted as mix MRF(me2x5). We compare these four classifiers and both learning principles based on the same performance measures as in the previous section.

We present the results of this comparison in Figure 7.2, which shows bar plots of each of the four performance measures for each of the four classifiers and both learning principles. In close analogy to Figure 7.1, we show the results for the generative MAP principle as black bars, while we show the results for the discriminative MSP principle as gray bars.

Considering the results for the generative MAP principle, we find that the two classifiers based on mixture models outperform the two corresponding classifiers based on single models with respect to all four performance measures. We also find that the two classifiers based on MRFs and mix MRFs yield a higher classification performance than the two corresponding classifiers based on iMM(1) and mix iMM(1). The classifier based on a mix MRF(me2x5) yields the lowest fpr (7.1%), the highest auc-ROC (0.9806), the highest ppv (38.5%), and the highest auc-PR (0.6830), stating that, among the four models tested, it is the most appropriate model for classifying human donor splice sites and decoy sites using the MAP principle.



(a) False positive rate (b) auc-ROC (c) Positive predictive value (d) auc-PR

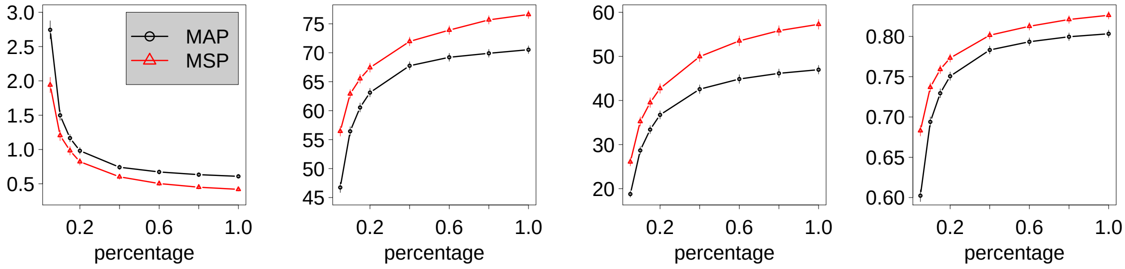
Figure 7.2: Comparison of classification performance for donor splice sites based on the MAP and the MSP principle. In each subfigure, we plot the performance measure on the ordinate and the models on the abscissa. Black bars indicate the results of MAP-trained classifiers, while red bars indicate the results of MSP-trained classifiers.

In analogy to the generative MAP principle, we now consider the results for the discriminative MSP principle visualized by gray bars in Figure 7.2. We find that the discriminatively trained classifier based on mix iMM(1) outperforms the corresponding classifier based on a single iMM(1). In contrast to this improvement of performance, we only find comparable results for the classifiers based on mix MRF(me2x5) and on single MRF(me2x5). Interestingly, we find that a discriminatively trained classifier based on mix iMM(1) performs best based on all performance measures. It yields an fpr of 7.0%, a ppv of 39.0%, a auc-ROC of 0.981, and the best auc-PR of 0.6876.

Comparing for each classifier the black and gray bar in Figure 7.2, we find that the four MSP-trained classifiers outperform the corresponding MAP-trained classifiers. For instance, the iMM(1) classifier yields an ppv of 37.8% for the MSP principle and only 35.7% for the MAP principle, and the mix MRF(me2x5)) classifier yields a ppv of 39.0% for the MSP principle and only 38.5% for the MAP principle. Interestingly, classifiers based on simple models (iMM(1) and mix iMM(1)) show the greatest improvement when replacing the MAP principle by the MSP principle. This observation is in accordance with previous findings that discriminative learning seems to be advantageous over generative learning if the model assumptions are wrong [Greiner et al., 2005].

7.2.2 Case study 2: Influence of the size of the training data set

In case study 2, we illustrate a comparison of Markov models trained on different amounts of the training data set using the GTPD prior. We choose the data set of [Wingender et al., 1996] containing 257 aligned BSs, each of length 16 bp, of the mammalian TF Sp1 as foreground data set and 267 second exons of human genes with a total size of approximately 68 kb as background data set. We use a PWM model as foreground model and an iMM(3) as background model. Results for all other combinations of a Markov model of orders 0 or 1 as foreground model and Markov models of orders 0 to 3 as



(a) False positive rate (b) Sensitivity (c) Positive predictive value (d) auc-PR

Figure 7.3: Comparison of classification performance for different sizes of the training data sets using the MAP and the MSP principle. We compare the classification performance of classifiers using the MAP principle (black line) and the MSP principle (red line) with the GTPD prior on differently sized training data sets for BSs of the TF Sp1 using a 1,000-fold hold-out sampling. For both classifiers, we use a PWM model in the foreground and an iMM(3) in the background. We plot the four performance measures, false positive rate, sensitivity, positive predictive value, and area under the PR curve (auc-PR) against the percentage of the preliminary training data set used to estimate the parameters. Whiskers indicate two-fold standard error. We find, that the classification performance increases with increasing size of the training data set. For the false positive rate this corresponds to a decreasing curve. For all four measures and all sizes of the data set, we find that the discriminatively trained Markov models yield a consistently higher classification performance than the generatively trained Markov models.

background model are available in Additional File 2 of article [Keilwagen et al., 2010b].

We use a *stratified hold-out sampling* procedure for the comparison of the classification performance of the resulting classifiers. Before the hold-out sampling, we chunk the background data set into non-overlapping sequences of length of at most 100 bp to avoid artificial class ratios during the hold-out sampling. In each iteration of the stratified hold-out sampling procedure, we randomly partition both the foreground data set and the background data set into a preliminary training data set comprising 90% of the sequences and a test data set comprising the remaining 10% of the sequences. In order to vary the size of the training data set, we use an additional sampling step, where we randomly draw a given fraction of the preliminary training data sets ranging from 5% to 100% yielding the final training data sets. We train all classifiers corresponding to different learning principles on the same subsets of the preliminary training data sets, and we evaluate the resulting classifiers on the same sequences in the test data sets.

We evaluate the classification performance on the test data sets using as performance measures the fpr for a fixed Sn of 95%, the Sn for a fixed Sp of 99.9%, the ppv for a fixed Sn of 95%, and the auc-PR. We repeat the stratified hold-out sampling procedure several times, and report the means and standard errors of the four performance measures fpr, Sn, ppv, and auc-PR for each classifier as the final result of the comparison.

We perform a $1,000$-fold stratified hold-out sampling with these models trained either by the MAP principle or by the MSP principle using the same priors for both cases. We choose for both cases an

ESS of 4 for the foreground model and an ESS of 1024 for the background model. We present the results of this comparison in Figure 7.3, which shows the four performance measures fpr, Sn, ppv, and auc-PR as functions of the relative size of the training data sets. The classification performance increases rapidly with increasing size of the training data set and achieves its optimal value for the largest training data sets. For the largest training data set, the discriminatively trained classifier yields an fpr of 0.4%, an Sn of 76.6%, a ppv of 57.3%, and an auc-PR of 0.826, whereas the generatively trained classifier yields only an fpr of 0.6%, an Sn of 70.5%, a ppv of 47.0%, and an auc-PR of 0.803.

Ng & Jordan compare the classification performance of a classifier based on PWMs trained by either the MAP principle or the MCL principle on a number of data sets from the UCI machine-learning repository [Ng and Jordan, 2002]. They find that for large data sets the discriminative MCL principle has a lower asymptotic error, corresponding to a higher classification performance, but that the generative MAP principle yields a higher classification performance for small data sets.

Based on the GTPD prior, it is now possible to compare the two Bayesian learning principles, MAP and MSP, directly using exactly the same priors in both cases. Based on the chosen model combination and prior, we find a superior classification performance of the discriminatively compared to the generatively trained classifiers irrespective of the size of the training data set. This result gives a first hint that it might be problematic to compare results obtained by the MAP and the MCL principle, since the prior could possibly bias the results.

7.3 Applications of the unified generative-discriminative learning principle

In the last two sections, we compare the performance of classifiers using two non-Bayesian learning principles and two Bayesian learning principles. These four learning principles are special cases of the learning principle presented in subsection *Unified generative-discriminative learning principle* (page 13). In this section, we consider the complete simplex $\underline{\beta}$ that includes the ML, the MCL, the MAP, and the MSP principle, as well as the hybrid learning principles GDT and PGDT. In the first case study, we use the unified generative-discriminative learning principle for the recognition of TFBSs. In the second case study, we investigate the influence of the size of the training data set and of the prior. Finally, we return to the first case study of the previous section using the unified generative-discriminative learning principle for donor splice sites.

7.3.1 Case study 1: Transcription factor binding site recognition

In this case study, we investigate whether the unified generative-discriminative learning principle might possibly allow an improvement of the recognition of TFBSs [Keilwagen et al., 2010c]. We consider four data sets of vertebrate TFBSs of length $L = 16$ bp collected from the TRANSFAC database [Wingender et al., 1996], namely AR/GR/PR, GATA, NF-κB, and Thyroid containing 104, 110, 72, and 127 BSs, respectively. For each of these foreground data sets, we use the same background data set as described in the previous case study.

With the goal of classifying, for each family of TFs separately, any 16-mer as a BS or as subsequence of a second exon, we build a naïve Bayes classifier consisting of two PWM models using the GTPD prior with an ESS of 4 and 1024 for the foreground and the background class, respectively. In analogy to the previous case study, we perform a 1,000-fold stratified hold-out sampling with 90% of the data for training and 10% of the data for assessing the same performance measures for the evaluation of the unified generative-discriminative learning principle.

In Figure 7.4, we visualize the results for the four data sets and the four performance measures. Initially, we restrict ourselves to the BSs of a family of TFs called AR/GR/PR and the Sn as performance measure, which is depicted in the upper left panel, and later we also consider the other data sets and performance measures.

Considering the ML principle located at $(\beta_0, \beta_1) = (0, 1)$ and the MCL principle located at $(\beta_0, \beta_1) = (1, 0)$, we find a Sn of 54.7% and 55.2%, respectively. Interestingly, the MCL principle achieves a higher Sn for a given Sp of 99.9% than the ML principle for this small data set. Using the GTPD prior with hyper-parameters corresponding to uniform pseudo-data, the sensitivities can be increased. Considering the MAP principle located at $(\beta_0, \beta_1) = (0, 0.5)$ and the MSP principle located at $(\beta_0, \beta_1) = (0.5, 0)$, we obtain a Sn of 54.9% and 55.6%, respectively. This shows that the MSP principle yields an increase of Sn of 0.7% compared to the MAP principle, consistent with the general observation that discriminatively learned classifiers often outperform their generatively-learned counterparts. This increase of Sn is achieved using the same prior and the same hyper-parameters for both learning principles, but it is possible that the particular choice of the hyper-parameters may favour one of the learning principles.

Following Equations (3.11b) and (3.11d), each point on the β_0- and β_1-axis corresponds to the MSP and the MAP principle, respectively, with specific hyper-parameters $\underline{\alpha}$. The location on the axis indicates the strength of the prior reflected by the virtual ESS. Next, we investigate for both learning principles the influence of the strength of the prior on the sensitivity results using a with step width of 0.05 along the axes. For the MAP principle, the Sn ranges from 54.7% for $\underline{\beta} = (0, 0.05, 0.95)$ to 54.8% for $\underline{\beta} = (0, 0.95, 0.05)$, achieving a maximum of 55.1% for $\underline{\beta} = (0, 0.1, 0.9)$. For the MSP principle, the Sn ranges from the maximum value 56.7% for $\underline{\beta} = (0.05, 0, 0.95)$ to 55.3% for $\underline{\beta} = (0.95, 0, 0.05)$. Comparing the maximum sensitivities for both learning principles and different virtual ESSs, we find that the MSP principle with a maximum Sn of 56.7% clearly outperforms the MAP principle by 1.6%, whereas the difference of sensitivities is only 0.7% for the original ESS.

Investigating this increase in the difference of sensitivities between the results for the MAP and the MSP principle, we find that the Sn increases for decreasing β_0 on the β_0-axis, which corresponds to the MSP principle with an increasing virtual ESS of the prior. In contrast to this observation, the sensitivity for the MAP principle increases less strongly with an increasing virtual ESS. This finding gives a first hint that a prior with a large ESS might be beneficial for the MSP principle, while we cannot observe a similar effect for the MAP principle in this case.

Next, we consider the lines $\beta_1 = \nu - \beta_0$ for $\nu \in [0, 1]$, which correspond to the hybrid learning

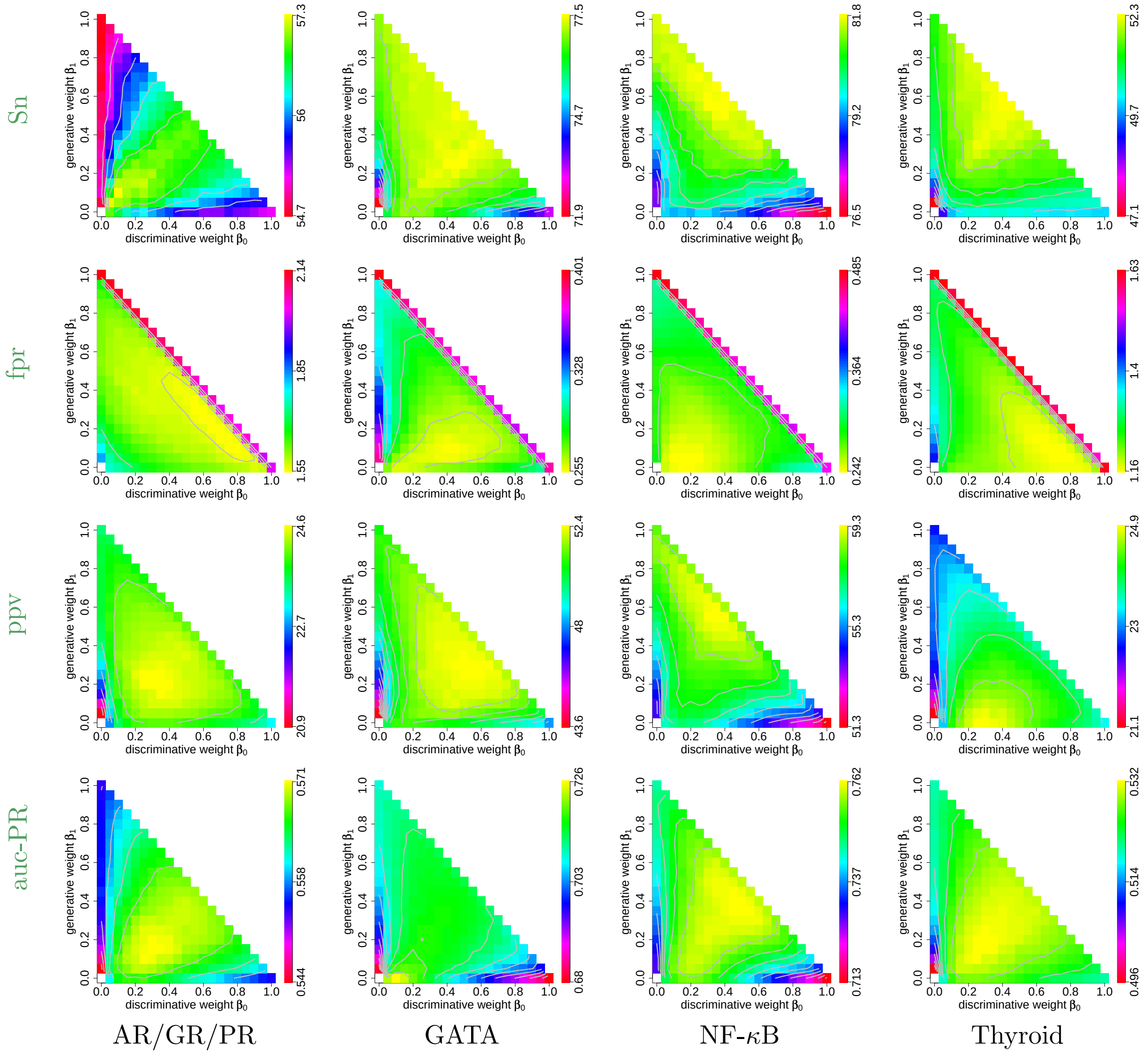


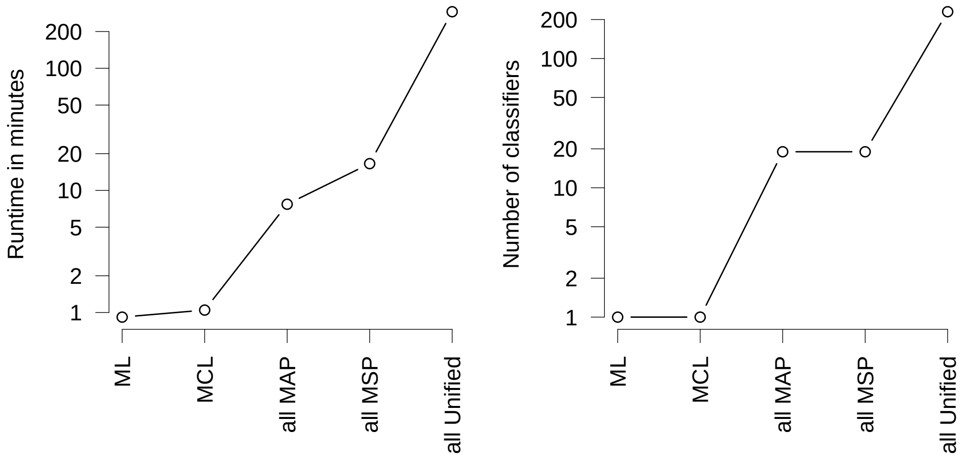
Figure 7.4: Results of the 1,000-fold stratified hold-out sampling procedure for four data sets of TFBSs and the unified generative-discriminative learning principle. Each column contains the subfigures for one data set corresponding to a family of TFs, and each row contains the subfigures for a specific performance measure. In each subfigure, we plot the values of the specific performance measure as a function of $\underline{\beta}$. Yellow indicates the best results, red indicates the worst results, and the gray contour lines in each subfigure indicate multiples of the standard error of the best result. We find the best results in the interior of the simplex $\underline{\beta}$ in 13 out of 16 cases.

		AR/GR/PR	GATA	NF-κB	Thyroid
Sn in %	ML	54.7	77.0	81.6	51.3
	MCL	55.2	73.2	76.5	50.0
	MAP	55.1	77.0	81.6	51.3
	MSP	56.9	77.0	79.6	50.3
	Unified	**57.3**	**77.5**	**81.8**	**52.3**
fpr in %	ML	2.14	0.401	0.485	1.63
	MCL	2.01	0.384	0.437	1.63
	MAP	1.61	0.309	0.285	1.28
	MSP	1.57	0.260	0.243	**1.16**
	Unified	**1.55**	**0.255**	**0.242**	**1.16**
ppv in %	ML	23.5	50.6	58.2	22.6
	MCL	23.1	48.1	51.3	23.4
	MAP	23.7	50.9	58.6	23.0
	MSP	24.1	51.1	57.1	**24.9**
	Unified	**24.6**	**52.4**	**59.3**	**24.9**
auc-PR	ML	0.554	0.709	0.746	0.520
	MCL	0.554	0.680	0.713	0.520
	MAP	0.555	0.711	0.747	0.520
	MSP	0.567	**0.727**	0.756	0.528
	Unified	**0.571**	**0.727**	**0.762**	**0.532**

Table 7.2: Summary of the results of Figure 7.4. For each of the four data sets and each of the four performance measures, we present the results for the ML, the MCL, the MAP, the MSP, and the unified generative-discriminative learning principle. For the MAP, the MSP, and the unified generative-discriminative learning principle, we present the best results from each of the 16 simplices (Figure 7.4). We find that the best results, displayed in bold face, are obtained by the unified generative-discriminative learning principle. Results that are at least one standard error greater than the corresponding results of the other learning principles are highlighted by gray cells.

principles GDT and PGDT for $\nu = 1$ and $\nu = 0.5$, respectively. For the GDT principle, the Sn ranges from 54.7% for $\underline{\beta} = (0, 1, 0)$ to 55.2% for $\underline{\beta} = (1, 0, 0)$, reaching a maximum of 56.9% for $\underline{\beta} = (0.55, 0.45, 0)$. For the PGDT principle, the Sn ranges from 54.9% for $\underline{\beta} = (0, 0.5, 0.5)$ to 55.6% for $\underline{\beta} = (0.5, 0, 0.5)$, reaching a maximum of 57.1% for $\underline{\beta} = (0.3, 0.2, 0.5)$. For both learning principles, we find that the Sn is initially increasing and finally decreasing. This observation indicates that neither the MAP nor the MSP principle with a GTPD prior representing uniform pseudo-data is optimal for estimating the parameter vector $\underline{\lambda}$.

Next, we investigate the interior of the simplex. We vary both β_0 and β_1 along a grid with stepwidth 0.05, and we find the highest Sn of 57.3% for $\underline{\beta} = (0.1, 0.1, 0.8)$. We find the region of highest Sn clearly inside the simplex near the angle bisector. This region corresponds to the MSP principle with an informative prior based on weighted likelihood and weighted original prior. Comparing the highest Sn for the GDT, the PGDT, and the unified generative-discriminative learning principle, we find that it increases from 56.9% over 57.1% to 57.3%, confirming that the prior can have a positive



(a) Runtime for one step of cross-validation.

(b) Number of trained classifiers.

Figure 7.5: Computational costs for different learning principles. The figure shows the runtime in minutes and the number of trained classifiers on a logarithmic scale plotted against groups of classifier utilizing the same learning principles. The results for generative learning principles of the comparison are obtained by numerical optimization in this case, but could be found analytically, in principle. We find that the behaviour of both curves is qualitatively similar indicating that there is a strong relation between the runtime and the number of trained classifiers.

influence on the performance.

Turning to the Sn of the other three TFs GATA, NF-κB, and Thyroid, we find qualitatively similar results. The highest sensitivities are located inside the simplex, while the lowest sensitivities are located on the axes. For BSs of the TF GATA, we obtain a Sn of 77.5% for $\underline{\beta} = (0.45, 0.25, 0.3)$; for the BSs of the TF NF-κB, we obtain a Sn of 81.8% for $\underline{\beta} = (0.4, 0.55, 0.05)$; and for the BSs of the TF Thyroid, we obtain 52.3% for $\underline{\beta} = (0.4, 0.55, 0.05)$. Similar to the data set of AR/GR/PR, we find a small region with high Sn for the BSs of the TFs NF-κB and Thyroid, while we find a broad region with high Sn for the BSs of the TF GATA.

We find that for all four data sets of TFBSs the unified generative-discriminative learning principle yields the highest sensitivities. Regarding the β_1-axis, which corresponds to the MAP principle using the GTPD prior representing uniform pseudo-data with different ESSs, we find that increasing the prior weight β_2, which is equivalent to decreasing the generative weight β_1, often reduces the Sn. We obtain the lowest Sn for the MAP principle for the largest prior weights β_2 in almost all cases. In contrast to this observation, we find on the β_0-axis, which corresponds to the MSP principle with the GTPD prior representing uniform pseudo-data with different ESSs, that increasing the prior weight β_2 improves the Sn at least initially.

Interestingly, we obtain qualitatively similar results when using performance measures alternative to Sn (Figure 7.4). These observations suggest that the same classifier trained either by generative or by discriminative learning principles may prefer different ESSs despite of using hyper-parameters that correspond to uniform pseudo-data. Hence, the strength of the prior has a decisive influence on comparisons of the results from generative and discriminative learning principles as well as the results of the Bayesian hybrid learning principles as for instance PGDT. Most importantly, we find that the unified generative-discriminative learning principle leads to an improvement for almost all of the used data sets and performance measures. We summarize the results for the ML, the MCL, the MAP, the MSP, and the unified generative-discriminative learning principle in Table 7.2.

Although the runtime depends on many factors, as for instance, the hardware, the implementation, the models, the prior, and the data set, we consider the runtime for one step in the cross-validation for the Thyroid data set to get an impression of the qualitative behaviour. In Figure 7.5, we plot the runtime and the number of trained classifiers utilizing the same learning principles. For the ML and the MCL principle, we need to train only one classifier, which takes approximately one minute. For all classifiers corresponding to the β_0- and β_1-axis denoted by “all MSP” and “all MAP”, respectively, we already need to train 19 classifiers, which takes between 5 and 20 minutes. Finally, we have to train 230 classifiers for the group “all unified”, which takes approximately 5 hours. Summarizing these numbers, we find the same qualitative behaviour for the runtime and the number of trained classifiers indicating that the runtime is mainly caused by the number of classifier that is trained.

7.3.2 Case study 2: Influence of the size of the training data set and of the prior

In a second study, we perform stratified hold-out sampling using the unified generative-discriminative learning principle and the same models, prior, and hyper-parameters as in the second study of the previous section. For computational reasons, we restrict the hold-out sampling to 100 iterations and the fraction of the preliminary training data sets to 5%, 20%, and 100%. We present the results of this comparison in Figure 7.6, which shows each of the four performance measures fpr, Sn, ppv, and auc-PR in one row, each containing a plot for 5%, 20%, and 100% of the preliminary training data sets. For numerical reasons, we do not compute the results for the ML and the MCL principle, i. e., the corners of the simplex, since some parameters tend to go to infinity if no prior is used. However, when using a weak prior, which can be achieved by a small prior weight β_2, we obtain similar results.

For fpr the results vary between 0.4% to 3.0% with a standard error ranging from 0.03% to 0.28%, for Sn the results vary between 40.4% to 77.1% with a standard error ranging from 1.0% to 1.6%, for ppv the results vary between 18.3% to 58.8% with a standard error ranging from 0.9% to 2.0%, and for auc-PR the results vary between 0.539 to 0.837 with a standard error ranging from 0.007 to 0.014. For all measures, we observe the worst results for the smallest fraction of the preliminary training data and the best results for the complete preliminary training data set. For each measure, the best value of a specific fraction of the preliminary training data set is about the lowest value of the next bigger fraction of the preliminary training data set. Since the classification performance increases very rapidly by increasing the size of the training data sets, we use independent scales for the plots in a row. Next, we compare the results for each fraction of the preliminary training data set separately.

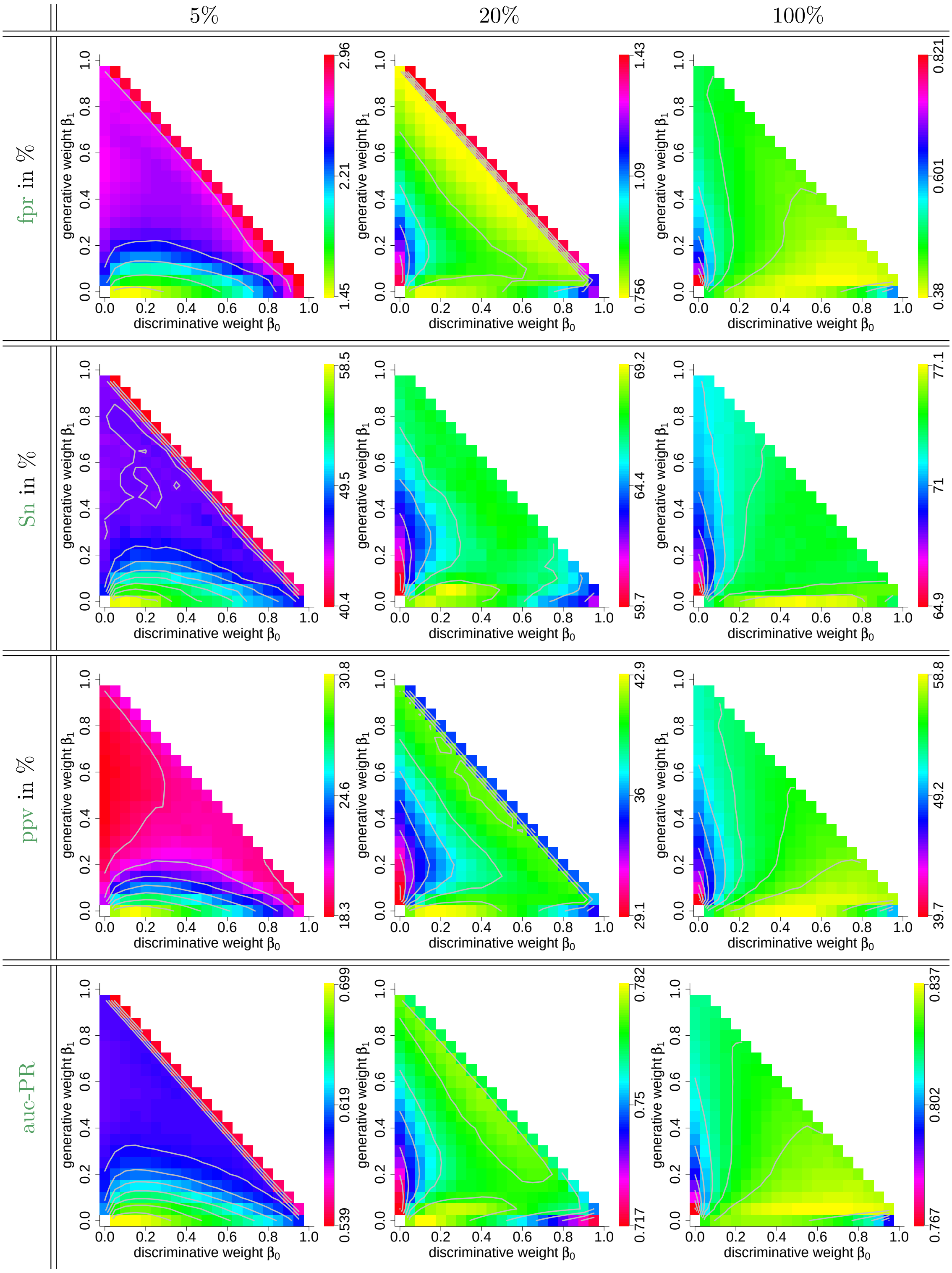


Figure 7.6: Results for unified generative-discriminative learning principle for different sizes of the training data sets. The plots are organized in a table where rows stand for performance measures and columns stand for the used percentage of the preliminary training data sets.

For the smallest fraction of the preliminary training data set, i. e., 5% which corresponds to only 12 sequences from the foreground class, we observe the best classification performance right of the lower left corner, which corresponds to the MSP principle with a strong prior. For all four performance measures, we observe the best results using weights of about $\underline{\beta} = (0.15, 0, 0.85)$ surrounded by a smooth fade-out in all directions with a skew on the β_0-axes. From this observation, we conclude that using the MSP principle with strong prior, namely a GTPD prior with foreground ESS of approximately 23 and background ESS of approximately 5803, yields the best results. Additionally, we observe inferior results for fpr, Sn, and auc-PR at the diagonal $\beta_1 = 1 - \beta_0$, which corresponds to the GDT principle.

For a medium fraction of the preliminary training data set, i. e., 20% which corresponds to only 47 sequences from the foreground class, we observe the worst results above the lower left corner, which corresponds to the MAP principle with a strong prior, and left of the right lower corner, which corresponds to the MSP principle with a weak prior. Again, also the GDT principle provides inferior results. The rest of the simplex shows a comparable performance with again good results for the MSP principle with a strong prior.

For the complete preliminary training data set, we observe the worst results for very small or very high discriminative weights β_0, while for moderate values of β_0, we observe the best result. Interestingly, we find that the GDT principle performs well for this size of training data set. Concerning the performance for small and medium fractions of the preliminary data set, we find that the GDT principle is not the optimal choice as learning principle. One reason for this suboptimal performance might be the accidental non-occurrence of some nucleotides or oligonucleotides at specific positions due to the size of the training data set. The prior, which helps to handle this problem for the rest of the simplex, is not used in the GDT principle.

Since the prior fulfills the condition of Equation (3.12), we can compare the results for the MAP and the MSP principle using the GTPD prior representing uniform pseudo-data but different ESSs. We compare the classification performance on the β_0-axes and on the β_1-axes, which we show in Figure 7.7 as function of the prior weight β_2 for a better comparability. The figure has the same structure as Figure 7.6 showing the performance measures in rows and the fraction of the preliminary training data sets in columns. As described earlier, the virtual ESS, which corresponds to a specific prior weight β_2, can be computed by multiplying the initially chosen ESS by $\frac{\beta_2}{1-\beta_2}$. The prior weight β_2 ranges from 0.05 to 0.95 yielding a virtual ESSs ranging from 76 for the foreground class and 19456 for the background class to approximately 0.2 for the foreground class and approximately 54 for the background class.

We find that the results of the MSP principle have the same qualitative behavior for all fractions of the preliminary training data set, which is an initially, strongly increasing performance with increasing prior weight β_2 and finally a decreasing performance for the highest prior weights β_2. For the complete preliminary training data set, we find a saturation for moderate prior weights β_2. In contrast to these findings, we find that the results for the MAP principle have a different characteristics for 5%, 20%, or 100% of the preliminary training data set. For 5% of the preliminary training data set, the MAP principle results are mainly increasing with an increasing prior weight β_2, while we find the opposite

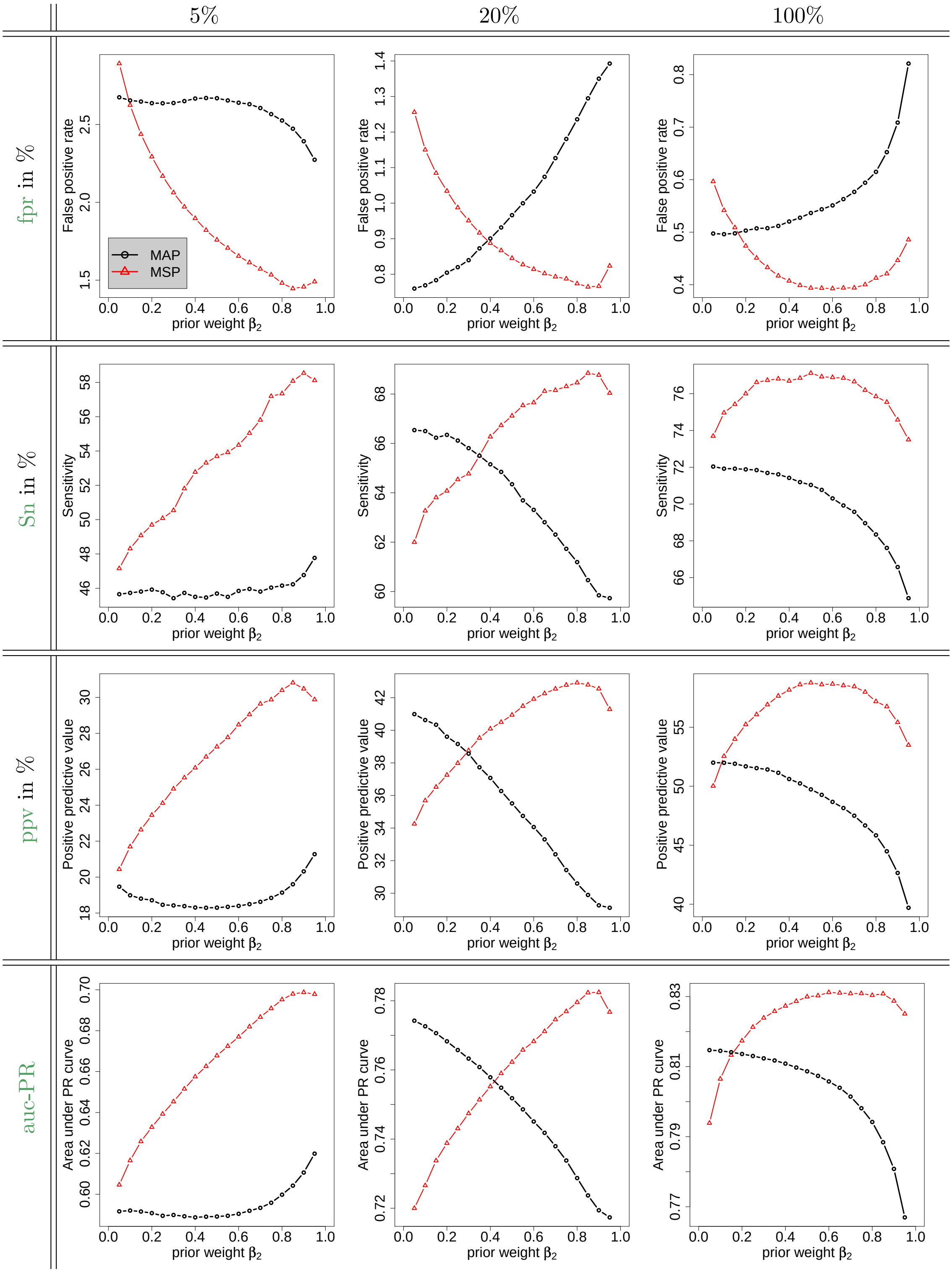


Figure 7.7: Comparison of the MAP and the MSP principle for different ESSs and different sizes of the training data sets. In each panel, the black line shows the results for the MAP principle, whereas the red line shows the results for the MSP principle.

behavior, namely a decreasing performance for increasing prior weights β_2, for 20% and 100%.

Using the complete preliminary training data set, we find that the MSP principle reaches the best performance for all four performance measures. For the smallest fraction of the preliminary training data set, we make a similar observation. Yet, it is not obvious from Figure 7.7 how the results of the MAP principle behave if the prior weight β_2 converges to 1, which corresponds to a further increase of the ESS. We test whether a further increase might improve the results for the MAP principle and might give even better results than for the MSP principle (data not shown). We find that a further increase of the prior weight β_2 slightly improves all measures but still remains worse than the results of the MSP principle. For the medium fraction of the preliminary training data set, the behavior is less clear. Still, we find the best results for Sn, ppv, and auc-PR for the MSP principle, but for fpr both learning principles reach approximately the same values.

Figure 7.6 and 7.7 show that the performance is dramatically influenced by the choice of the hyper-parameters. In the comparison of the MAP and the MSP principles with uniform pseudo-data, we find that both learning principles prefer different prior-weights β_2, which correspond to different ESSs. In addition, we find different optimal ESSs for different sizes of data sets. While for the MSP principle the optimal ESSs differ only marginally, we find large differences for the MAP principle. Nevertheless, for this data set, the MSP principle gives the best performance for all sizes of the training data set. In contrast to [Ng and Jordan, 2002], we find that even for small data sets the MSP principle performs better than the MAP principle using the GTPD prior.

7.3.3 Case study 3: Donor splice site recognition

This case study is based on the first case study of the previous section, where we compare classifiers for donor splice site prediction based on different models from the family of MRF trained either by the MAP or by the MSP principle. Here, we use the unified generative-discriminative learning principle that allows to interpolate between the MAP and the MSP principle. For reason of comparability, we choose the same model combinations, the same prior and hyper-parameters, the same performance measures, and the same data sets. The only difference between both case studies is the learning principle employed to estimate the parameters of the classifier.

In Table 7.3, we show the results of these case studies in close analogy to Table 7.2. For numerical reasons, we do not use the ML and MCL principle. For each learning principle and each model combination, Table 7.3 contains best results with respect to the interpretation of the simplex of the unified generative-discriminative learning principle illustrated in Figure 3.2. We find in all cases that classifiers trained using the MSP principle outperform their corresponding counterparts trained using the MAP principle. The best classifier based on the MAP and the MSP principle is the classifier using for both classes a mixture of two MRF(me2x5). This classifier yields a fpr of 6.84%, an auc-ROC of 0.9810, a ppv of 39.5%, and an auc-PR of 0.687 for the MAP principle. These results can be improved using the MSP principle obtaining a fpr of 6.76%, a auc-ROC of 0.9812, a ppv of 39.7%, and an auc-PR of 0.690.

Comparing the results for the MAP and the MSP principle with those obtained for the MAP and

		iMM(1)	mix iMM(1)	MRF(me2x5)	mix MRF(me2x5)
fpr in %	MAP	7.82	7.44	7.24	6.84
	MSP	7.29	6.97	6.95	6.76
	Unified	7.29	6.97	6.95	**6.66**
auc-ROC	MAP	0.9778	0.9792	0.9794	0.9810
	MSP	0.9795	0.9809	0.9805	0.9812
	Unified	0.9795	0.9809	0.9805	**0.9814**
ppv in %	MAP	36.3	37.4	38.1	39.5
	MSP	37.9	39.0	39.1	39.7
	Unified	37.9	39.0	39.1	**40.1**
auc-PR	MAP	0.669	0.671	0.678	0.687
	MSP	0.679	0.688	0.688	0.690
	Unified	0.679	0.688	0.688	**0.692**

Table 7.3: Results of the unified generative-discriminative learning principle for donor splice sites. The table shows the four performance measures Sn, auc-ROC, ppv, and auc-PR for the four model combinations, namely iMM(1), mix iMM(1), MRF(me2x5), and mix MRF(me2x5). For each learning principle and each model combination, we show the best results with respect to the interpretation of the simplex of the unified generative-discriminative learning principle (Figure 3.2). The best results of all model combinations are display in bold face for each performance measure.

the MSP principle with fixed hyper-parameters illustrated in Figure 7.2, we find that varying the strength of the prior, which corresponds to the ESS, has a positive influence on the performance. For instance, the classifier based on the mixture of two MRF(me2x5) yields a ppv of 39.0% and 39.7% using the MSP principle with fixed ESS and variable ESS, respectively.

Turning to the results of the unified generative-discriminative learning principle, we find that for iMM(1), mix iMM(1), and MRF(me2x5) the results do not differ from those obtained for the MSP principle. This indicates that the best results for the unified generative-discriminative learning principle are located on the β_0-axis. Using the likelihood even with small weight β_1 does not seem to be beneficial in this case. Interestingly, we find that the unified generative-discriminative learning principle helps to improve the performance of the most complex model, i. e., the mix MRF(me2x5). We obtain the best results for all performance measures using this learning principle. Specifically, we obtain a fpr of 6.66%, a auc-ROC of 0.9814, a ppv of 40.1%, and a auc-PR of 0.692. We find that using the unified generative-discriminative learning principle instead of the MSP principle yields a similar improvement of the performance measures as using the MSP principle instead of the MAP principle.

7.4 Conclusions

In this chapter, we investigate the influence of the learning principle on the performance of the classifier. We find that the learning principle has a decisive influence on the performance, and that choosing the learning principle carefully is very important. Specifically, we find that choosing an appropriate learning principle can be at least as important as choosing an appropriate model.

Comparing discriminative and generative learning principles, we find that it is often beneficial to use discriminative instead of generative learning principles. Specifically, we find for splice site data with thousands of sequences that classifiers trained using discriminative learning principles outperform classifiers trained using generative learning principles irrespective of comparing the MCL and the ML principle, or the MSP and the MAP principle. In addition, we find that MSP can also improve the performance of TFBS recognition. Varying the size of the training data set, we show that using the GTPD prior for both learning principles, MSP and MAP, classifiers trained using the discriminative learning principle can outperform their generative counterparts.

Turning to the unified generative-discriminative learning principle, we find that classifiers trained using hybrid learning principles can outperform their purely generative or discriminative counterparts. We find for four data sets of TFBSs and donor splice site data that classifiers trained using the unified generative-discriminative learning principle with weights $\underline{\beta}$, which correspond to the interior of the simplex that is neither purely generative nor purely discriminative, yield the best results. Considering two edges of the simplex, which correspond to purely generative and purely discriminative learning principles, we find that influence of the prior differs for generative and discriminative learning principle. Assessing different strengths of the prior, we still find that classifiers trained using discriminative learning principles outperform their generative counterparts.

Nevertheless, generative learning principles still have advantages over discriminative learning principles. First, generative learning principles often allow to infer the parameters of the model analytically, while for the discriminative counterpart, we have to use numerical optimization methods. Hence, it is often faster to estimate the parameters of classifiers using generative learning principles than estimating the parameters of classifiers using discriminative learning principles. Second, generative learning principles allow to estimate parameters if the class labels are unknown. In this case using a mixture model leads to a so-called *unsupervised* learning, which infers the class labels from the data.

However, we find that for labeled data discriminative learning principles often lead to better results than obtained by generative learning principles. Using hybrid learning principles, we can often improve these results. We obtain this improvement by optimizing several classifiers with different $\underline{\beta}$ which, in total, requires a much longer total runtime. In the following chapters, we use the presented probabilistic models and learning principles to solve concrete biological questions. In next chapter, we aim at improving the classification of DNA sequences as donor splice sites of *Caenorhabditis elegans* using probabilistic models, while we aim at improving the recognition of human TSSs in the chapter *Recognition of human transcription start sites* (page 72). In chapter *Computational reassessment of transcription factor binding site annotations* (page 82), we aim at identifying potential annotation errors in gene-regulatory databases containing TFBS annotations utilizing a generatively trained probabilistic model. Finally in chapter *Discriminative de-novo motif discovery utilizing positional preference* (page 92), we aim at finding de-novo motifs of TFBSs using a discriminative learning principle.

Chapter 8

Donor splice site recognition in Caenorhabditis elegans

After successfully applying the unified generative-discriminative learning principle and the GTPD prior to learning probabilistic models on benchmark data sets, we consider in this and in the following chapters more biologically relevant data sets. In this chapter, we investigate the recognition of donor splice sites of the model organism *Caenorhabditis elegans*, which has been investigated in [Sonnenburg et al., 2007] using support vector machines. As mentioned earlier, splicing has a decisive influence on the pre-mRNA and therefore on all downstream processes as for instance translation of mRNA to the corresponding amino acid sequence. Hence, the recognition of splice sites is of great importance for the assessment of coding DNA regions and thus for many biological applications. For instance, tools for splice site recognition can be used in the computation of spliced alignments for improving the mapping of protein sequences to the genome.

During the last years, a plethora of tools for splice site recognition has been developed. Many of these tools score candidate splice sites using only the nucleotides in the proximity of the candidate splice site [Zhang and Marr, 1993, Castelo and Guigo, 2004, Yeo and Burge, 2004, Zhao et al., 2005]. In this case, sequences have a length of about 10 bp. However, the recognition of splice sites can be improved using longer sequences [Degroeve et al., 2005, Sonnenburg et al., 2007]. For this reason, we investigate whether probabilistic models, which also model longer sequences, can be applied to improve the recognition of splice sites using probabilistic models assessed in the previous chapter as core components.

In the case studies performed in the previous chapter, we find that discriminatively trained mixtures of iMM(1) perform very well for the main splice site. Based on this finding, we build an IPM with four component models for the foreground class where each component model is a mixture. The first component models the first 75 bp of the sequence. This part of the sequence corresponds to the upstream region of the donor splice site and is at least partially an exon. Since exons can either be coding or non-coding, we use a mixture of a cMM(3,2) and a hMM(3) for this component. The second component models 11 bp of the sequence, which corresponds to the donor splice site. Based on the case studies of the previous chapter, we use a mixture model of two iMM(1) for this part of the sequence. The remaining sequence of 55 bp corresponds to the downstream region of the donor splice site, which is at least partially an intron. We model this part by two components, one for the proximal and one for the remaining downstream region. For both parts, we use a mixture of two hMM(3). In

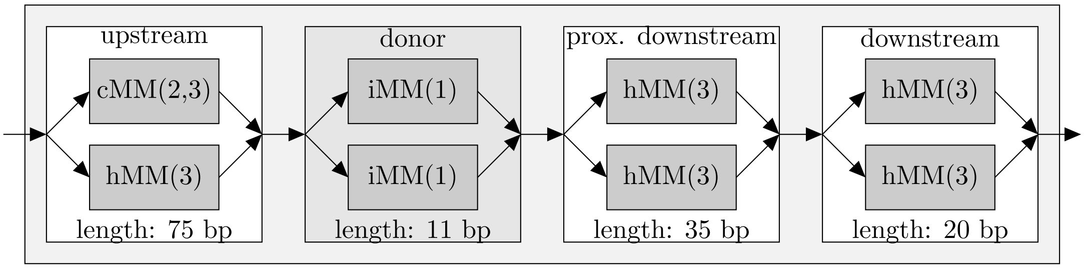


Figure 8.1: Foreground model for donor splice sites.

contrast to the first two components, we do not fix the length of the last two components a-priorily, but learn it during training. In Figure 8.1, we show the IPM for a subsequence length of 35 and 20 bp for the last two components.

For this reason, we vary the length of the last two component models on a grid of 5 bp, i. e., we build ten IPMs where the last two components model subsequence of length 5 bp and 50 bp, respectively, for the first IPM, and 50 bp and 5 bp, respectively, for the last IPM. For each IPM, we use the same background model, which is a mixture of an cMM(2,3) and a hMM(3), and the same prior for building a classifier. We use the same learning principle for all these classifiers, which is the MSP principle. For a given training data set, we estimate the parameters of each IPM by starting a numerical optimization procedure ten times from different randomly chosen initializations, and we use the parameters from the run with the highest supervised posterior. We obtain ten trained IPMs that we assess with an independent test data set. We select the IPM with the highest auc-PR as optimal model.

Based on the partitioned data of [Sonnenburg et al., 2007] with about 65,000 real and about 2,845,000 decoy donor splice sites of length 141 bp, we perform a 5-fold cross-validation to compute the mean classification performance and the standard error for each performance measure. In each iteration of the cross-validation, we train the ten IPMs on three parts of the data, and we validate the models on one part of the data. We pick the model with the highest auc-PR and train it on four parts, namely the tree parts that have been used for training and the one part that has been used for validation. We use a GTPD with foreground ESS of 32 and background ESS of 1280 ($= 40 \cdot 32$) for training, which reflects the approximate class ratio in the data. For testing the classification performance, we use the five performance measures Sn, fpr, ppv, auc-ROC, and auc-PR for the remaining part of the data that has not been used for training or validation before.

In Table 8.1, we present the results of the performance measures and their standard errors for the validation and the test data sets. Comparing the results for the IPM on the validation and the test data sets, we find similar results for both data sets. The results for the test data set are in certain cases even better than those obtained for the validation data set indicating that the model is not overfitted to the validation data set, and that increasing the training data set may further improve the classification performance slightly.

Scrutinizing the optimal model of each iteration of the cross-validation, we find that the same IPM

	validation performance	**test performance**
Sn in %	$80.68 \pm 2.4 \times 10^{-1}$	$81.27 \pm 2.1 \times 10^{-1}$
fpr in %	$0.54 \pm 1.1 \times 10^{-2}$	$0.52 \pm 1.1 \times 10^{-2}$
ppv in %	$80.00 \pm 4.1 \times 10^{-1}$	$80.55 \pm 3.6 \times 10^{-1}$
auc-ROC	$0.9983 \pm 7.4 \times 10^{-5}$	$0.9983 \pm 7.5 \times 10^{-5}$
auc-PR	$0.9552 \pm 5.2 \times 10^{-4}$	$0.9563 \pm 6.0 \times 10^{-4}$

Table 8.1: Classification performance for donor splice sites of *Caenorhabditis elegans* using a classifier based on an IPM visualized in Figure 8.1.

is selected in each iteration of the cross-validation. In Figure 8.1, we show the selected IPM, in which the last two components model subsequences of length 35 bp and 20 bp, respectively. This finding indicates that there is a strong difference between the proximal 35 bp and the distal 20 bp in the foreground sets compared to the background data sets, and that this difference is present in all parts of the data sets.

Comparing the results of the five performance measures for the test data set with those presented in the previous chapter, we find a significant improvement. This increase can be attributed to two possible aspects. On the one hand, the number of sequences in the training data set is much higher than in the studies of the previous chapter. On the other hand, the sequence length has increased from 7 bp to 141 bp. In the previous case studies, we only model the main donor splice site, whereas in the current case we also include the upstream and the downstream region of the donor splice site.

We investigate this difference, and we train a classifier on sequences truncated to the main donor splice site of length 11 bp. This classifier, which we denote as truncated classifier, uses the same background model as the classifier presented above, whereas it uses as foreground model only the second component of the IPM, which models the main donor splice site. We train this classifier on four parts of the data, and test its performance on the remaining fifth part. Using this classifier, we find significantly less good results for all performance measures depicted in Table 8.2. The results of this classifier are more comparable to the results presented in the previous chapter. This finding indicates that the first, third, and forth component of the IPM contribute substantially to the performance of the classifier, which emphasizes that modeling upstream and downstream regions of the splice sites is reasonable.

Besides investigating the upstream and downstream components, we examine the influence of adding hidden variables in the classifier. Each component of the IPM used as foreground model as well as the background model utilizes a mixture model or an cyclic Markov models. These two types of models utilize hidden variables that allow a higher flexibility of the model. Removing this flexibility, we build an classifier that utilizes an IPM with four component models as foreground model and a hMM(3) as background model. The IPM is composed of a hMM(3) for the initial 75 bp, a iMM(1) for the main donor splice site, ahMM(3) for the proximal 35 bp downstream, and another hMM(3) for the remaining 20 bp. Performing a 5-fold cross-validation with the given splits, we train this classifier on four splits and assess its performance on the remaining split. We obtain an auc-ROC of $0.9979 \pm 6.5 \times 10^{-5}$ and an auc-PR of $0.9435 \pm 4.6 \times 10^{-4}$. Comparing the values for auc-PR, which is a

	auc-ROC	**auc-PR**
optimal classifier	$0.9983 \pm 7.5 \times 10^{-5}$	$0.9563 \pm 6.0 \times 10^{-4}$
truncated classifier	$0.9893 \pm 2.1 \times 10^{-4}$	$0.7475 \pm 3.1 \times 10^{-3}$
linear classifier	$0.9979 \pm 6.5 \times 10^{-5}$	$0.9435 \pm 4.6 \times 10^{-4}$
SVM in [Sonnenburg et al., 2007]	$0.9982 \pm 1 \times 10^{-4}$	$0.9534 \pm 1.0 \times 10^{-3}$

Table 8.2: Comparison of area under the curve for donor splice sites of *Caenorhabditis elegans*. The table shows the auc-ROC and auc-PR of four classifiers. The first classifier denoted by optimal classifier is selected during the cross-validation, the second classifier denoted by truncated classifier only uses 11 bp close to the donor site, the third classifier denoted by linear classifier does not use hidden variables, and the fourth classifier is the support vector machine presented in [Sonnenburg et al., 2007]. For this support vector machine, we take the values for auc-ROC and auc-PR from [Sonnenburg et al., 2007].

good performance measure in the presence of skewed classes, we find that the optimal classifier yields a value that is about 0.013 higher than the value for the linear classifier. This significant difference indicates that utilizing hidden variables improves the performance of the classifier.

However, the features of linear classifiers like IPMs without hidden variables can be more easily transferred into the features of a support vector machine using, for instance, a spectrum kernel [Leslie et al., 2002]. Interestingly, we find that for this data set the classifier based on the IPM with hidden variables outperforms the support vector machine presented in [Sonnenburg et al., 2007], which yields an auc-PR of $0.9534 \pm 1.0 \times 10^{-3}$ for the test data set as indicated in Table 8.2. This indicates that an appropriate combination of probabilistic models, prior, and learning principle yields results comparable to state-of-the-art classifiers based on support vector machines. Nevertheless, utilizing hidden variables complicates the training of the classifier as the landscape of the objective function becomes more and more bumpy, while we can always obtain global optimum for support vector machines due to convex optimization. However, one important advantage of probabilistic model over support vector machines is that further hidden variables can be incorporated easily for including further biological knowledge or hypotheses on demand.

Chapter 9

Recognition of human transcription start sites

Following the prediction of donor splice sites in the previous chapter, we turn to the prediction of human transcription start sites in this chapter. Genes are the most important feature of genomic DNA, since they encode proteins and RNAs that are responsible for the development and maintenance of all organisms. The approximate location of genes can be obtained, for instance, by blasting expressed sequence tags (ESTs) against the genome. However, blasting ESTs often does not elucidate the TSSs of the genes, which is essential for subsequent promoter analysis including, for instance, the prediction of TFBSs using databases with known binding motifs of TFs or de-novo motif discovery. Several experimental techniques including rapid amplification of cDNA ends (RACE) [Frohman et al., 1988] and cap-analysis of gene expression (CAGE) [Carninci et al., 2006] have been developed for determining TSSs of genes. Data coming from these methods show that in many cases genes have multiple TSSs, which often cluster building so-called transcription start regions[1]. However, lowly expressed or tissue-specific genes might be missed by these techniques. One feasible way for determining TSSs of such genes is the computational prediction.

A large number of different tools for the recognition of TSSs has been proposed during the last years. Recently, 17 state-of-the-art tools have been compared using the same test data and four different evaluation protocols [Abeel et al., 2009] in the first large scale comparison of promoter prediction programs (PPPs).

The remainder of this chapter is structured as followed: First, we discuss the established protocols of [Abeel et al., 2009]. Second, we describe PACT, a probabilistic approach to CAGE tags, including the training data and the parameter learning. Finally, we compare PACT with the four best performing tools discussed in [Abeel et al., 2009].

9.1 Established protocols

These protocols can be categorized by two criteria. On one hand, the protocols differ by the way predictions are declared to be correct, which is either based on the binned genome with a bin size of 500 bp denoted as protocol 1 or based on distances with a maximal distance of 500 bp denoted as protocol 2. On the other hand, the protocols differ by the data that are used for defining the

---

[1] For simplicity, we use TSS synonymous for transcription start site and transcription start regions.

ground truth, which is either based on CAGE tags [Carninci et al., 2006] denoted as protocol A or based on RefSeq genes downloaded from the UCSC table browser including $23,799$ unique gene models [Kent et al., 2002], denoted as protocol B. Additionally, protocol B discards all intergenic predictions from the evaluation to avoid doubtful false positives due to intergenic regions that are related to unknown genes or other types of transcription [Bajic et al., 2004].

In detail, there are four protocols, namely, protocol

- $1A$, which uses CAGE tags as ground truth and the binned genome with a bin size of 500 bp,
- $1B$, which uses the gene set as ground truth and the binned genome with a bin size of 500 bp,
- $2A$, which uses CAGE tags as ground truth and the distance to the closest real positive, and
- $2B$, which uses the gene set as ground truth and the distance to the closest real positive.

For each protocol $i \in \{1A, 1B, 2A, 2B\}$, the auc-PR is measured and denoted by auc-PR$_i$. For comparing a scalar value in the end, the PPP score,

$$\text{PPP} := \frac{4}{\sum_{i \in \{1A,1B,2A,2B\}} \frac{1}{\text{auc-PR}_i}}, \tag{9.1}$$

which is the harmonic mean of the auc-PRs for all protocols, is used. The PPP score favours tools with a stable performance over all protocols, since the harmonic mean reduces the effect of high outliers, while at the same time increases the effect of low scores. For more details regarding all tested tools, the protocols, and the PPP score, we refer the reader to [Abeel et al., 2009]. Based on the PPP score, the tools ARTS [Sonnenburg et al., 2006], Eponine [Down and Hubbard, 2002], ProSOM [Abeel et al., 2008b], and EP3 [Abeel et al., 2008a] have been reported as best performing tools, where ARTS clearly outperforms all other tools.

However, the tested tools are trained on different training data sets, which at least partially overlap with the test data. For instance, protocols B are based on TSSs of approximately $24,000$ RefSeq genes [Abeel et al., 2009], but ARTS is trained on TSSs of approximately $8,500$ RefSeq genes, reaching an auc-PR of 0.9991 for this training data set [Sonnenburg et al., 2006]. Similarly, the training data sets of other tools overlap with the data used for assessing the performance in [Abeel et al., 2009]. Hence, the performance of some tools might be overestimated.

For further analysis, it has been proposed to do a chromosomewise cross-validation by the authors of [Abeel et al., 2009]. Since genes and CAGE tags are not uniformly distributed over all chromosomes, the class ratio between foreground and background class in the test data sets used in each step of the cross-validation would strongly vary between $1 : 7,414$ and $1 : 221,352$. However, the class ratio in the data sets has a decisive influence on the results of the auc-PR [Davis and Goadrich, 2006]. Computing the mean auc-PR from these values during a chromosomewise cross-validation is therefore not optimal. Hence, we propose to use a k-fold cross validation where each split provides approximately the same class ratio.

Considering the proposed protocols, we find that these are based on the ideas of [Sonnenburg et al., 2006, Abeel et al., 2008b] to allow a detailed comparison. However, the current protocols might possibly lead to some problems that complicate an unbiased comparison of different tools.

On one hand, the protocols are strictly separated by the underlying experimental data, but on the other hand, we find that the TSSs proposed by these experiments partially overlap. For instance, more than 50% of the TSSs proposed by CAGE are located inside of RefSeq genes 500 bp downstream of the corresponding gene start. Hence, the labeling sequence of genes downstream of the gene start as background class might not be optimal. For this reason, the results of protocols B might be too pessimistic.

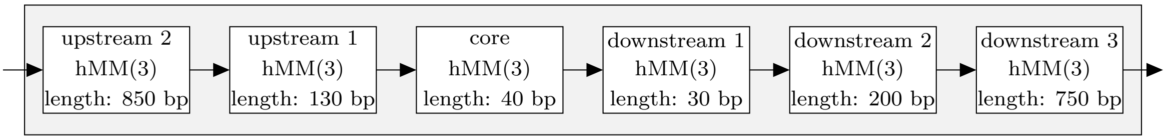
Conversely, looking at the RefSeq genes, we find that more than 25% of the gene starts have a distance of more than 500 bp to the nearest TSS proposed by CAGE. Hence, labeling all nucleotides that do not overlap with a TSS proposed by CAGE as background class might be problematic. For this reason, the results of protocols A might be too pessimistic.

In addition, we find TSSs for both protocols that are not labeled as positive examples in the other protocol. For this reason, it is impossible to reach an auc-PR of 1 for all four protocols at the same time.

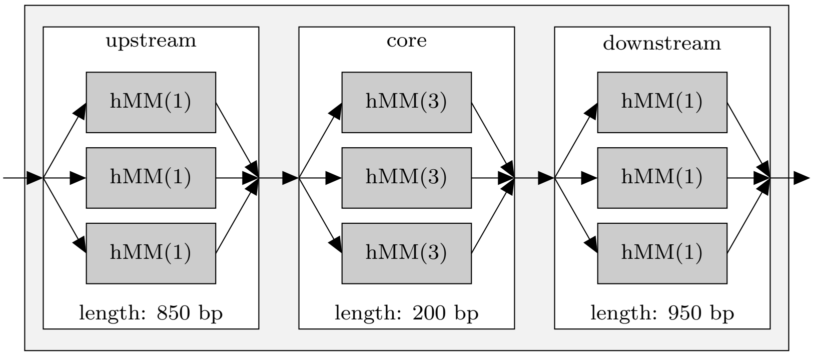
Furthermore, the bins in protocols 1 might obtain positive labels in a too conservative way. Originally, it is proposed to add 20 bp upstream and downstream of the annotated gene start [Sonnenburg et al., 2006] to enlarge the gene start, and subsequently to label the corresponding bins. However, the current version of protocols A does not enlarge the experimental evidences. Hence, a TSS can start or end in close proximity to the border of a bin leading to a false positive prediction if the prediction is shifted by only few positions. Furthermore, the current implementation [Abeel et al., 2009] always labels only one bin as positive, even if the experimental evidence overlaps with more than one bin.

Finally, protocols 2 depend on the resolution of the predictions, since they count the true positive for computing the precision as number of predictions that overlap with experimental evidence. However, they do not take into account that several predictions can overlap with the same experimental evidence. Conversely, they count false positives as number of predictions that do not overlap with experimental evidence. Hence, it is hard to compare PPPs that predict at different resolutions. For instance, PPPs that predict at a very high resolution, such as predictions with a resolution of individual base pairs, could possibly suffer from this convention.

These difficulties show that it is hard to define protocols that allow a fair assessment of PPPs. However, the proposed protocols are the first attempt to enable a large scale comparative study of PPPs. For this reason, we use these protocols also here, which are implemented in the tool pppBenchmark [Abeel et al., 2009], to allow a direct comparison with 17 state-of-the-art tools.



(a) Foreground model.



(b) Background model.

Figure 9.1: Probabilistic models used in PACT. In Figure 9.1(a), we show the foreground model of PACT that is an IPM with six component models. In Figure 9.1(b), we show the background model of PACT that is also an IPM but utilizes three mixture models of homogeneous Markov models.

9.2 Approach

In the previous chapter, we find that modeling the upstream and the downstream region of donor splice sites using an appropriate IPM yields a good classification performance. Typically, long sequences are also used for the prediction of TSSs. With the goal of predicting TSSs accurately, we build a classifier based on two IPMs for sequences of length 2000 bp, where the start of the TSS, if any, is located at position 1001.

In Figure 9.1, we visualize the foreground and background model used for building the classifier. As foreground model, we choose an IPM with six components. The first two components model the sequence upstream of the TSS, the third component models the start of the TSS, and the last three components model the sequence downstream of the TSS. For each of these components, we use a hMM(3). For the background model, we also use an IPM but only with three components. For modeling the sequence heterogeneity of the background class, we use mixture models for each component of the IPM. Specifically, we choose a mixture of three hMM(1) for the first and the last component, which model the upstream and the downstream sequence, respectively, and a mixture of three hMM(3) for the second component, which models the start of the TSS.

For training and assessing the classifier, we download the human genome, version hg18, from UCSC Genome Browser and the TSSs based on CAGE tags provided by [Carninci et al., 2006, Abeel et al., 2009]. We split the genome in 3 parts, where each split has a class ratio of about 1:17,000 bp covered by TSSs versus all bp. In Table 9.1, we show some statistics of these splits

Part	Chromosome	TSSs	bp	bp covered by TSSs	Class ratio
0	1, 5, 10, 11, 13, 14, 20, 21	60,559	1.03×10^9	35.5×10^6	1:16,972
1	2, 3, 6, 9, 12, 15, 22	60,468	1.04×10^9	35.5×10^6	1:17,133
2	4, 7, 8, 16, 17, 18, 19, X, Y	59,386	1.02×10^9	34.1×10^6	1:17,118

Table 9.1: Statistics of splits for human CAGE data based on [Carninci et al., 2006].

indicating that the TSSs seem to be uniformly distributed over all three splits. Using these splits, we extract foreground and background data sets with a class ratio of 1:25 and a sequence length of 2000 bp, where we align the start of a TSS at position 1001. Based on these data sets, we perform a 3-fold cross-validation. For learning the parameters of the classifier, we use the MSP principle and an ESS of 4 and 120 ($= 4 \times 25$) for foreground and background model, respectively. For each step in the cross-validation, we estimate the parameters of the classifier by starting a numerical optimization procedure ten times from different randomly chosen initializations, and choosing the parameters from the run with the highest supervised posterior. Using the corresponding classifier, we subsequently predict the probability of being a TSS for each nucleotide of the split that is not used during training. We assess these predictions using pppBenchmark, version 1.3.

9.3 Results and conclusions

In Figure 9.2, we show the precision recall curves of PACT for each split of the cross-validation in comparison with ARTS, Eponine, ProSOM, and EP3. In Table 9.2, we show the corresponding auc-PRs and PPP scores form the cross-validation in comparison with the results of ARTS, Eponine, ProSOM, and EP3.[2]

Considering individual protocols, we find that for protocol $1A$ the auc-PR of PACT varies between 0.217 and 0.237 with a mean of 0.224. Comparing these values with results of the other tools, we find a slightly better performance than for ARTS (auc-PR = 0.191), Eponine (auc-PR = 0.164), ProSOM (auc-PR = 0.182), and EP3 (auc-PR = 0.176). However, protocol $1A$ is based on CAGE tags that have not been used by ARTS, Eponine, and ProSOM for training. On the other hand, we perform a 3-fold cross-validation for PACT to avoid an overestimation of the performance, which might be the case for some of the other tools since training and test data sets are not strictly disjoint.

Turning to protocol $1B$, we find for PACT an auc-PR that varies between 0.312 and 0.340 with a mean of 0.321. In contrast to the results for protocol $1A$, we find that ARTS performs slightly better than PACT especially for low recall rates yielding an auc-PR of 0.367. However, the current implementation of pppBenchmark uses a fixed number of bins for the predicted probabilities to determine the precision recall curves. This may possibly lead to artefacts in the curves if the probabilities are not equally distributed between 0 and 1. Nevertheless, PACT performs second best yielding a higher auc-PR than Eponine (auc-PR = 0.294), ProSOM (auc-PR = 0.247), and EP3 (auc-PR = 0.230).

---

[2] The results for ARTS, Eponine, ProSOM, and EP3 are slightly different from those reported in [Abeel et al., 2009] due to the update from pppBenchmark 1.0 to pppBenchmark 1.3 (private communication with Thomas Abeel).

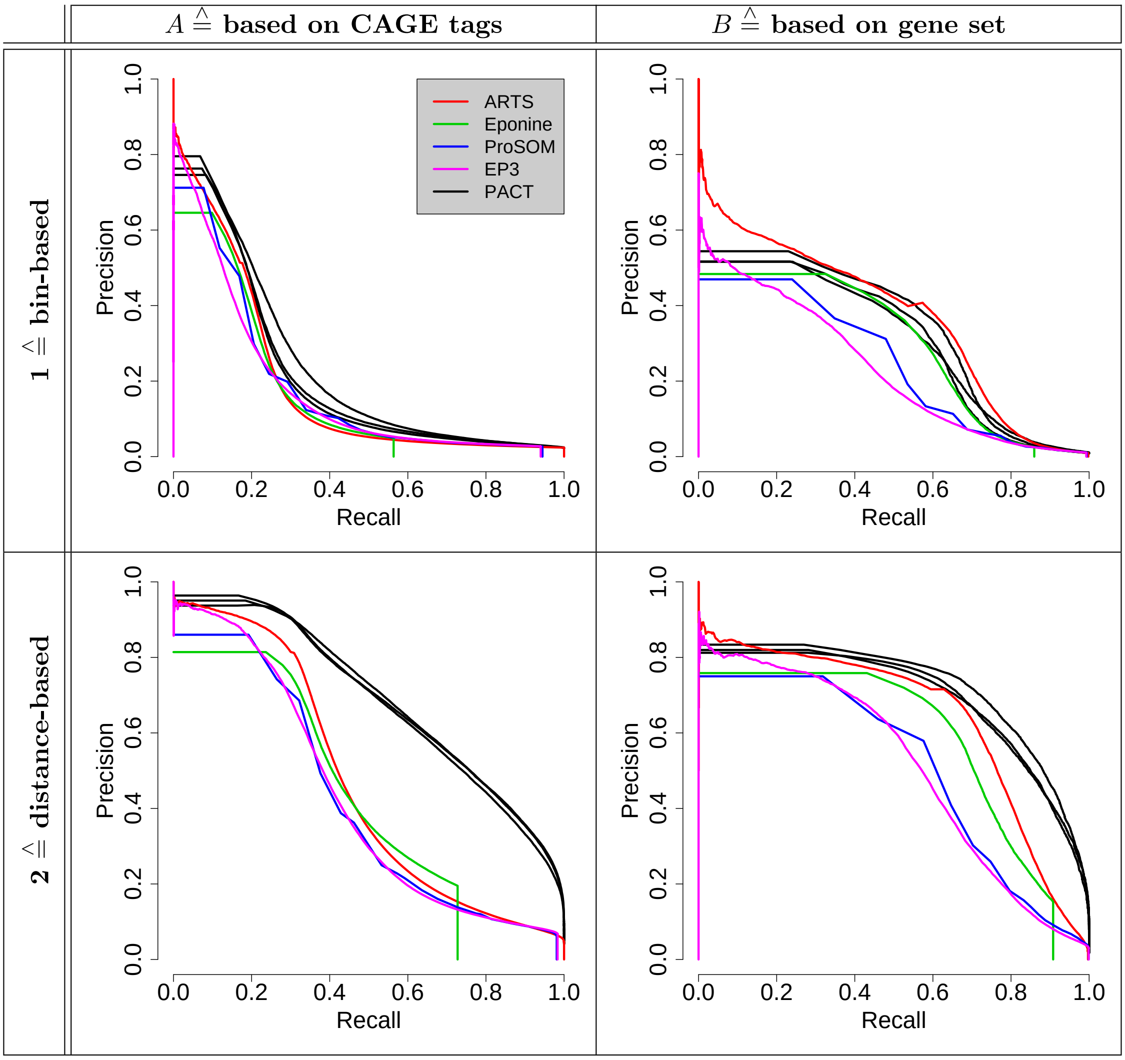


Figure 9.2: Comparison of precision recall curves for promoter prediction programs and the four protocols proposed in [Abeel et al., 2009]. The plot is organized as table, where the first row contains the results for protocol 1 and the second row contains the results for protocol 2. Similarly, the first column contains the results of protocol A, whereas the second column contains the results of protocol B. Each panel shows the results of ARTS (red line), Eponine (green line), ProSOM (blue line), EP3 (magenta line), and PACT (3 black lines). For PACT, we obtain 3 curves due to the 3-fold cross-validation. For the auc-PR of all tools, we refer to Table 9.2.

Tool	**auc-PR for protocol**				**PPP score**
	1A	**1B**	**2A**	**2B**	
ARTS	0.191	0.367	0.466	0.639	0.343
Eponine	0.164	0.294	0.410	0.573	0.292
ProSOM	0.182	0.247	0.416	0.507	0.287
EP3	0.176	0.230	0.424	0.507	0.279
PACT	0.219	0.312	0.692	0.692	0.375
	0.217	0.340	0.680	0.721	0.384
	0.237	0.312	0.695	0.694	0.388
(mean) PACT	0.224	0.321	0.689	0.702	0.383

Table 9.2: Performance of different promoter prediction programs. In the first four rows, we show the results of the four best performing tools of [Abeel et al., 2009] sorted by their PPP score. In rows five to seven, we show the results of PACT for each step of the cross-validation. In row eight, we show the mean performance of PACT with respect to the 3-fold cross-validation.

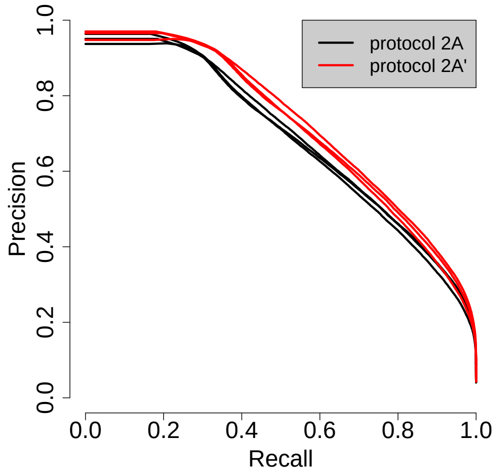
Scrutinizing the difference of the results for protocol $1A$ and $1B$, we find that ARTS performs slightly better for protocol $1B$ that is based on RefSeq genes, while PACT performs slightly better for protocol $1A$ that is based on CAGE tags. This finding could possibly be explained by the data sets used for the training of these tools. ARTS is trained on TSSs of approximately $8,500$ RefSeq genes [Sonnenburg et al., 2006], while we train PACT using CAGE data. Hence, both tools perform best on test data that are similar to the data used for training.

Considering the results for protocol $2A$, we find that PACT clearly outperforms all other tools. PACT yields an auc-PRs between 0.680 and 0.695 with a mean of 0.689, whereas ARTS yields an auc-PR of 0.466, Eponine yields an auc-PR of 0.410, ProSOM yields an auc-PR of 0.416, and EP3 yields an auc-PR of 0.424. Similar to protocol $1A$, protocol $2A$ is based on CAGE tags, which might be beneficial for PACT that is trained on data sets based on CAGE tags.

Finally, considering protocol $2B$, we find that the auc-PR varies between 0.692 and 0.721 with a mean of 0.702. Comparing these results with the results of the other tools, we find that ARTS yields a comparable auc-PR of 0.639, while PACT clearly outperforms Eponine with an auc-PR of 0.573, ProSOM with an auc-PR of 0.507, and EP3 with an auc-PR of 0.507.

Considering the overall performance of PACT, we find that in three out of four protocols PACT yields comparable results to those of ARTS, Eponine, EP3, and ProSOM, which perform best in [Abeel et al., 2009]. These three protocols include both protocol that are bin-based, protocol 1A and 1B, and both protocols that are based on RefSeq genes, protocol 1B and 2B. For the remaining protocol $2A$, which is based on CAGE tags, we find a significant better performance of PACT compared to the performances of ARTS, Eponine, EP3, and ProSOM. However, we note that PACT is trained on data sets based CAGE tags. Hence, the training data as well as PACT might be in better accordance with protocol $2A$ than the other tools.

Considering the PPP scores for PACT, we find that the values vary between 0.375 and 0.388 with a mean of 0.383. Comparing these values with the values obtained for the other tools, we find that



(a) Precision recall curves.

split	auc-PR for protocol	
	2A	2A'
1	0.692	0.722
2	0.680	0.712
3	0.695	0.731
mean	0.689	0.722

(b) Area under precision recall curves.

Figure 9.3: Results of PACT based on precision recall curves and auc-PRs for protocol $2A$ and $2A'$. Figure 9.3(a) visualizes the results for protocol $2A$ (black lines) and the results of protocol $2A'$ (red lines). We obtain three curves for both protocols due to the 3-fold cross-validation. Figure 9.3(b) show the auc-PRs and the mean auc-PRs for the curves visualized in Figure 9.3(a).

PACT performs slightly better than ARTS (PPP score of 0.343), Eponine (PPP = 0.292), ProSOM (PPP = 0.287), and EP3 (PPP = 0.279). However, these results are not directly comparable due to different training data sets and possibly due to different resolutions for the predictions. Yet, these results show that PACT yields comparable results to state-of-the-art PPPs.

Investigating the difference between ARTS, which performed best in [Abeel et al., 2009], and PACT, we find that the performance of ARTS is evaluated using predictions for non-overlapping bins of size 50 bp (private communication with Sören Sonnenburg), while PACT used predictions to single base pairs. To allow a more direct comparison to ARTS, we compute for each bin of size 50 bp the probability of containing a TSS as maximum of the probabilities of each nucleotide in this bin of being a TSS. Using these prediction, we again evaluate PACT, and we find that the results for protocols 1. However, for protocols 2 we find a decisive influence of the resolution of the predictions. For protocol $2A$, we obtain a mean auc-PR of 0.528 compared to 0.689 for base pair resolution, and for protocol $2B$, we obtain a mean auc-PR of 0.611 compared to 0.702 for base pair resolution. Interestingly, both performances decrease when decreasing the resolution of the predictions.

Considering the PPP score, we find a mean of 0.360 compared to 0.383 for base pair resolution. Comparing these values to the value obtained for ARTS, we find that PACT still performs slightly better. However, comparing the auc-PRs for individual protocols, we now find that ARTS performs slightly better for protocols based on RefSeq genes, while PACT performs slightly better for protocols based on CAGE data. This finding is in accordance with the training data used for the tools. Additionally, it indicates that the performance of ARTS might possibly improve when using predictions with base pair resolution. Possibly, other tools evaluated in [Abeel et al., 2009] might also improve when changing their resolution for the prediction. Hence, we recommend trying to use the same prediction resolution for all PPPs in future comparisons.

Investigating the protocols proposed in [Abeel et al., 2009], we find that either CAGE data or RefSeq genes are used in one protocol. As mentioned earlier, the data coming from these experiments only overlaps partially. Hence, some predictions are labeled as true positives in one protocol and as false positives in the other protocol. Aiming at solving this contradiction, we propose a modification of protocol $2A$, which we denote by $2A'$. This protocol is also based on distance as protocol $2A$, but in contrast to protocol $2A$ it used CAGE tags and RefSeq genes as ground truth.

In Figure 9.3, we show the results of this protocol in comparison with the results obtained from protocol $2A$. We obtain a mean auc-PR of 0.722 for this protocol compared to a mean auc-PR of 0.689 for protocol $2A$. This finding indicates that it might be beneficial to use experimental data from different sources to define the ground truth for an evaluation protocol. Following this proposal, it might also be beneficial to use data that are based on different sources of experiments for training the PPPs.

9.4 Conclusions

The annotation of TSSs is of fundamental importance for many areas of life science, as for instance, promoter analyses and subsequent prediction of TFBSs. Based on new experimental techniques as for instance CAGE, many previously unknown TSSs could be identified. However, TSSs of tissue-specific or lowly expressed genes and RNAs might be missed by experimental techniques, and one way out is the computational prediction of their TSSs.

During the last years a wealth of PPPs has been developed. However, it is hard to compare the performance of these tools. Often the tools are trained on different training data sets, and their performance is assessed using different evaluation strategies and different test data sets. Recently, the first large scale comparative study has been published that defines four evaluation strategies called protocols and compares 17 state-of-the-art PPPs on the human genome.

We present PACT, a probabilistic approach to CAGE tags, which is a discriminatively trained classifier based on two IPMs, and compare it to ARTS, Eponine, EP3, and ProSOM the four best performing tools of a recent study. Using a 3-fold cross-validation, we find that PACT performs well based on all four protocols. For three protocols, we find that PACT performs comparable to the four best tools of [Abeel et al., 2009] including ARTS, Eponine, EP3, and ProSOM. For protocol $2A$, which is distance-based and uses CAGE tags to define the TSSs, PACT clearly outperforms ARTS, Eponine, EP3, and ProSOM. However, since PACT is trained using data from CAGE experiments that are not used by ARTS, Eponine, and ProSOM, this is not surprising. Interestingly, in [Abeel et al., 2009] it has been pointed out that protocol $2A$ is preferable as it is the combination the more biologically inspired protocol A and the more accurate protocol 2.

We also find that there are several shortcomings of the currently used protocols as they might be too conservative when defining TSSs. Additionally, the performance of some tools might be overestimated due to an overlap of the training and test data sets. Furthermore, the PPPs are not directly comparable due to differences in the training data and possibly in the evaluation based on the resolution of their predictions. For this reason, this study as well as the study performed in [Abeel et al., 2009] only give a rough overview.

For an unbiased comparison of different PPPs one should unify the experimental evidence from different sources, should refine the evaluation protocols, and should perform a cross-validation for all PPPs of this study. However, such a comparison can only be made in a joint project of the developer of PPPs to avoid any biases, since handling the individual PPPs is a challenging task.

Chapter 10

Computational reassessment of transcription factor binding site annotations

After investigating splice sites and TSSs in the last chapters, we now turn to the regulation of gene expression, which involves a complex system of interacting components in all living organisms [Babu and Teichmann, 2003] and which is of fundamental interest, for instance, for cell maintenance and development. This complex process is controlled by many influential components such as the binding of proteins to DNA, the binding of miRNAs to mRNA, RNA editing, splicing of pre-mRNA, mRNA degradation, or post-translational modification. One of the fundamental regulatory steps is the binding of TFs to the promoters of their target genes [Pabo and Sauer, 1992]. These TFs influence the initiation of transcription as their binding to the target promoter can induce (activator) or inhibit (repressor) the transcription, and, hence, affect many subsequent regulatory processes. The general ability to control a target gene may depend on the BS itself, its strand orientation, and its position with respect to the TSS. If other BSs are present, the ability of a TF to bind the DNA may additionally depend on strand orientations and positions of these BSs.

One important prerequisite for research on gene regulation is the reliable annotation of BSs. The approximate regions on the double-stranded DNA sequence bound by TFs can be determined by wet-lab experiments such as electrophoretic mobility shift assays (EMSA) [Hellman and Fried, 2007], DNAse footprinting [Galas and Schmitz, 1978], enzyme-linked immunosorbent assay (ELISA) [Benotmane et al., 1997, Mönke et al., 2004], ChIP-chip [Sun et al., 2003, Wu et al., 2006], ChIP-seq [Johnson et al., 2007], or mutations of the putative BS and subsequent expression studies. Because TFs bind to double-stranded DNA, the strand annotations of non-palindromic BSs in the databases are either missing or added based on manual inspection or predictions from bioinformatics tools such as MEME [Bailey and Eklan, 1994], Gibbs Sampler [Lawrence et al., 1993, Thompson et al., 2003], Improbizer [Ao et al., 2004], SeSiMCMC [Favorov et al., 2005], or A-GLAM [Kim et al., 2008].

After wet-lab identification, data about transcriptional gene-regulatory interactions including the annotated BSs are published in the scientific literature. Subsequently, these data are extracted by curation teams and manually entered into databases on transcriptional gene regulation such as CoryneRegNet [Baumbach et al., 2009], PRODORIC [Münch et al., 2003], or

RegulonDB [Gama-Castro et al., 2008] for prokaryotes, and AGRIS [Palaniswamy et al., 2006], AthaMap [Bülow et al., 2009], CTCFBSDB [Bao et al., 2008], JASPAR [Bryne et al., 2008], ORegAnno [Montgomery et al., 2006], SCPD [Zhu and Zhang, 1999], TRANSFAC [Matys et al., 2006], TRED [Jiang et al., 2007], or TRRD [Kolchanov et al., 2002] for eukaryotes. Three typical problems may occur during the process of transferring these data.

1. Erroneously annotated BS: This error may occur in the original study or during the transfer process from the scientific literature to the databases. A sequence is declared to contain a BS although, in reality, it does not.

2. Shift of the BS: The BS may be erroneously shifted by one or a few base pairs. This typically happens during the transfer process from the scientific literature to the databases.

3. Missing or wrong strand orientation of the BS: The strand orientation of a BS is often not or incorrectly annotated. For example, all BS orientations are arbitrarily declared to be in $5' \rightarrow 3'$ direction relative to the target gene in CoryneRegNet and in RegulonDB [Baumbach et al., 2009, Gama-Castro et al., 2008].

These problems can strongly affect any of the subsequent analysis steps such as the inference of sequence motifs from “experimentally verified” data, the calculation of p-values for the occurrence of BSs, the detection of putative BSs in genome-wide scans and their experimental validation, or the reconstruction of transcriptional gene-regulatory networks.

In this chapter, which is based on the article [Keilwagen et al., 2009], we introduce MotifAdjuster, a software tool for detecting potential BS annotation errors and for proposing possible corrections. Existing bioinformatics tools [Bailey and Eklan, 1994, Lawrence et al., 1993, Thompson et al., 2003, Ao et al., 2004, Favorov et al., 2005, Kim et al., 2008] are not optimized for this task, because they do not allow shifting the BS using a nonuniform distribution and considering both strands with unequal weights. In contrast, MotifAdjuster allows the user to incorporate prior knowledge about 1) the probability of erroneously annotated BSs, 2) the distribution of possible shifts, and 3) the strand preference.

One widely-used model for the representation of BSs is the PWM model [Kel et al., 2003, Bailey and Eklan, 1994, Tompa et al., 2005, Lawrence et al., 1993, Thompson et al., 2003, Ao et al., 2004, Favorov et al., 2005, Kim et al., 2008], and many software tools for genome-wide scans of sequence motifs are based on PWM models [Münch et al., 2005, Kel et al., 2003]. MotifAdjuster is based on a ZOOPS model using a PWM model on both strands for the motif sequences and a homogeneous Markov model of order 0 for the flanking sequences similar to MEME, Gibbs Sampler, Improbizer, SeSiMCMC, or A-GLAM. For a given set of BSs, MotifAdjuster tests whether each sequence contains a BS, and it refines the annotations of position and strand for each BS, if necessary, by maximizing the posterior of the mixture model.

To test the efficacy of MotifAdjuster, we apply it to seven data sets from CoryneRegNet, and we record for each of them the set of potential annotation errors. For one example, the nitrate regulator

NarL, we compare the proposed adjustments to the original literature, to a manual strand reannotation of the BS strands, and to an independent and hand-curated reannotation provided by PRODORIC. Finally, we test if the PWM estimated from the adjusted NarL BSs can help to detect unknown BSs in those promoter regions that are known to be bound by NarL, but for which no BS could be predicted in the past.

10.1 Approach

For the detection of sequences erroneously annotated as containing BSs, with shifted BSs, or with missing or wrong strand annotations, we use the annotated BSs enlarged by 5 bp upstream and downstream. Using these sequences, we learn the parameters of a special ZOOPS model composed of a strand model of a PWM model as motif model, a skew normal model for the start position of the BSs, and homogeneous Markov model of order 0 for the flanking sequence with the MAP learning principle. For shortness of notation, we use subscript ZOOPS and PWM in this chapter.

We expect there are some sequences annotated to contain a BS despite they do not contain a BS in reality, but we believe that the fraction of such incorrectly annotated sequences is small. Hence, we choose $P^{\text{ZOOPS}}\left(u_1 = 0 \middle| \underline{\lambda}^{(\text{ZOOPS})}\right) = 0.2$ for the studies presented here, i. e., we assume that only 20% of the sequences annotated to contain a BS do not contain a BS in reality. We further expect that the annotated position of the BS might be shifted accidentally by a few base pairs, so we choose maximal shift to be 5 (i. e. $\Delta = 10$) and a skew normal model with fixed parameters $\underline{\lambda}^{(\text{skew})} = (0, 0, 0)$ which encodes a discrete normal distribution with mean 5 and standard deviation 1. This choice results in a conditional probability of approximately 40% that the BS is not shifted, of approximately 25% that it is shifted 1 bp, and of approximately 5% that it is shifted by more than 1 bp upstream or downstream of the annotated start position, respectively, given that a BS is present in sequence $\underline{x}$.

We denote the ESS of the ZOOPS model chosen prior to inspecting any database by ε, and we set the ESS of the PWM model to $P^{\text{ZOOPS}}\left(u_1 = 1 \middle| \underline{\lambda}_m^{(\text{ZOOPS})}\right) \cdot \varepsilon$, the positive hyper-parameters of the strand parameters to $\alpha_{m,0}^{\text{strand}} = \alpha_{m,1}^{\text{strand}} = P^{\text{ZOOPS}}\left(u_1 = 1 \middle| \underline{\lambda}_m^{(\text{ZOOPS})}\right) \cdot \frac{\varepsilon}{2}$, and the ESS of the homogeneous Markov model of order 0 to $\left[L - P^{\text{ZOOPS}}\left(u_1 = 1 \middle| \underline{\lambda}_m^{(\text{ZOOPS})}\right) \cdot w\right] \cdot \varepsilon$.

For the reannotation of BSs presented in this section, we choose an ESS of $\varepsilon = 5$, yielding an ESS of 4 for the PWM model, $\alpha_0^{\text{strand}} = \alpha_1^{\text{strand}} = 2$, and an ESS of 57 for the homogeneous Markov model of order 0. This choice yields $\alpha_{\ell,b}^{\text{PWM}} = 1$ for every $b \in \Sigma$ and every $\ell \in [1, w]$, stating that the chosen prior of the PWM model can be understood as a special case of the BDeu prior [Buntine, 1994, Heckerman et al., 1995], which in turn is a special case of the BD prior [Cooper and Herskovits, 1992].

Using this parameters and hyper-parameters, we learn the remaining parameters of the ZOOPS model by a numerical algorithm. We stop the algorithm if the logarithmic increase of the posterior between two subsequent iterations becomes smaller than 10^{-6}, restart the algorithm 10 times with randomly-chosen initial values, and choose the parameters of that start with the highest posterior, similar to [Lawrence and Reilly, 1990, Bailey and Eklan, 1994].

If we restrict the positional distribution of the ZOOPS model to be a uniform distribution over all possible start positions, if we set $P^{\text{strand}}\left(u = 1\middle|\underline{\lambda}_m^{(\text{strand})}\right) = 0.5$, and if we restrict the background model to be strand symmetric, then we obtain the probabilistic model which is the basis of [Lawrence and Reilly, 1990, Bailey and Eklan, 1994]. The flexibility allowed by MotifAdjuster is important for its practical applicability. Typically for the task of BS reannotation, the user has prior knowledge about the expected motif occurrence and the shift distribution, but no or only limited prior knowledge about the distribution of the BS strand orientation. Hence, we allow the user to specify the probability that a sequence contains a BS $P^{\text{ZOOPS}}\left(u_1 = 1\middle|\underline{\lambda}_m^{(\text{ZOOPS})}\right)$, a non-uniform positional distribution to incorporate the prior knowledge of the shift distribution, and we estimate the probability that the BS is located on the forward strand $P^{\text{strand}}\left(u = 0\middle|\underline{\lambda}_m^{(\text{strand})}\right)$ from the data. This setting allows MotifAdjuster to work, without additional intervention, also in the two extreme cases that the BSs lie predominantly either on the forward or on the reverse complementary strand.

Due to the object oriented implementation of MotifAdjuster, similar ZOOPS models can be implemented easily, for instance, by using other background and motif models such as Markov models of higher order [Zhang and Marr, 1993, Salzberg, 1997b, Thijs et al., 2001], permuted Markov models [Ellrott et al., 2002, Zhao et al., 2005], Bayesian networks [Barash et al., 2003, Castelo and Guigo, 2004], or their extensions to variable order [Rissanen, 1983, Ron et al., 1996, Boutilier et al., 1996, Bühlmann, 1997, Ben-Gal et al., 2005].

10.2 Results and Discussion

In this subsection we present the results of MotifAdjuster applied to seven data sets of *Escherichia coli*, the validation of MotifAdjuster results for NarL BSs, and the prediction of a novel NarL BS.

10.2.1 Results for seven data sets of *Escherichia coli*

For testing the efficacy of MotifAdjuster and improving the annotation of BSs of *Escherichia coli*, we extract all data sets with at least 30 BSs of length of at most 25 bp from the bacterial gene-regulatory reference database CoryneRegNet 4.0. The choice of at least 30 BSs of length of at most 25 bp is arbitrary, but motivated by the intention that the results of the following study should not be influenced by TFs with an insufficient number of BSs or by TFs with an atypical BS length. Seven data sets of BSs corresponding to the TFs CpxR, Crp, Fis, Fnr, Fur, Lrp, and NarL satisfy these requirements, and we apply MotifAdjuster to each of these seven data sets. We summarize the results obtained by MotifAdjuster in Table 10.1.

We find that all of the data sets are considered questionable by MotifAdjuster and, more surprisingly, that 34.5% of the 536 BS annotations are proposed for removal or shifts. The percentage of questionably annotated BSs ranges from 9.3% for Fnr to 95.7% for Fur. MotifAdjuster proposes to remove 51 of the 536 BSs and to shift 134 of the remaining 485 BSs by at least one bp indicating that, in these seven data sets, erroneous shifts of the annotated BSs are the most frequent annotation error. In particular, the percentage of proposed deletions ranges from 2.2% (1 out of 46) for Fur to 27.3% (9

ID	**name**	**#BSs**	**BS length**	**#Removed BSs**	**#Shifted BSs**	**Percentage**
b3357	*crp*	218	22	20	31	23.4%
b1221	*narL*	74	7	2	11	17.6%
b3261	*fis*	68	21	13	17	44.1%
b1334	*fnr*	54	14	2	3	9.3%
b0683	*fur*	46	15	1	43	95.7%
b0889	*lrp*	43	12	4	23	62.8%
b3912	*cpxR*	33	15	9	6	45.5%
Total		536		51	134	34.5%

Table 10.1: Summary of the results of the application of MotifAdjuster to all data sets of CoryneRegNet 4.0 from *Escherichia coli* with at least 30 BSs and of at most 25 bp length. Columns 1 and 2 show the gene ID and gene name of the TF, columns 3 and 4 show the number of BSs stored in the database and their lengths, columns 5 and 6 show the number of BSs proposed to be removed and to be shifted, and column 7 shows the percentage of BSs to be removed or shifted. Interestingly, the percentage of proposed adjustments varies strongly from TF to TF ranging from 9.3% for Fnr to 95.7% for Fur. In summary, we find in the complete data set of 536 BSs that 51 BSs are proposed to be removed and 134 BSs are proposed to be shifted, resulting in 34.5% of the data set being proposed for adjustments.

out of 33) for CpxR, while the percentage of proposed shifts ranges from 5.6% (3 out of 54) for Fnr to 93.5% (43 out of 46) for Fur. In more detail, we observe a broad range of shift lengths ranging from one shift 4 bp upstream to two shifts 4 bp downstream with a sharp peak about 0.

For each of the seven TFs, we analyze if the adjustments proposed by MotifAdjuster result in an improved motif of the BSs (Figure 10.1). We compute the sequence logos [Schneider and Stephens, 1990, Crooks et al., 2004] of the original BSs obtained from CoryneRegNet and those of the BSs proposed by MotifAdjuster, which we call original sequence logos and adjusted sequence logos, respectively. Comparing these sequence logos, we find that the adjusted sequence logos show a higher conservation than the original sequence logos in all seven cases. We also compare the sequence logos to consensus sequences obtained from the literature [De Wulf et al., 2002, Körner et al., 2003, Pan et al., 1996, Baichoo and Helmann, 2002, Cui et al., 1995, Maris et al., 2005], and we find that the adjusted sequence logos are more similar to the consensus sequences than the original sequence logos. In addition, we find for the TFs *CpxR*, *Fur*, and *NarL* that the adjusted sequence logos allow to recognize clear motifs that could not be recognized in the original sequence logos obtained from CoryneRegNet.

We investigate if there is any systematic dependence of the observed rate of proposed adjustments on the number of BSs, the BS length, and the GC-content of the BSs. We find no obvious dependence of the error rate on the number of BSs and on the BS length. Comparing the GC-content of the BSs, we find that the GC-content of the BSs of all but one TF ranges from 30% to 40%. However, the GC-content of the Fur BSs is only 20%. This low GC-content might be the reason for the unexpectedly high percentage of shifts in this data set, since it is more likely to accidentally shift a BS in a sequence composed of a virtually binary alphabet.

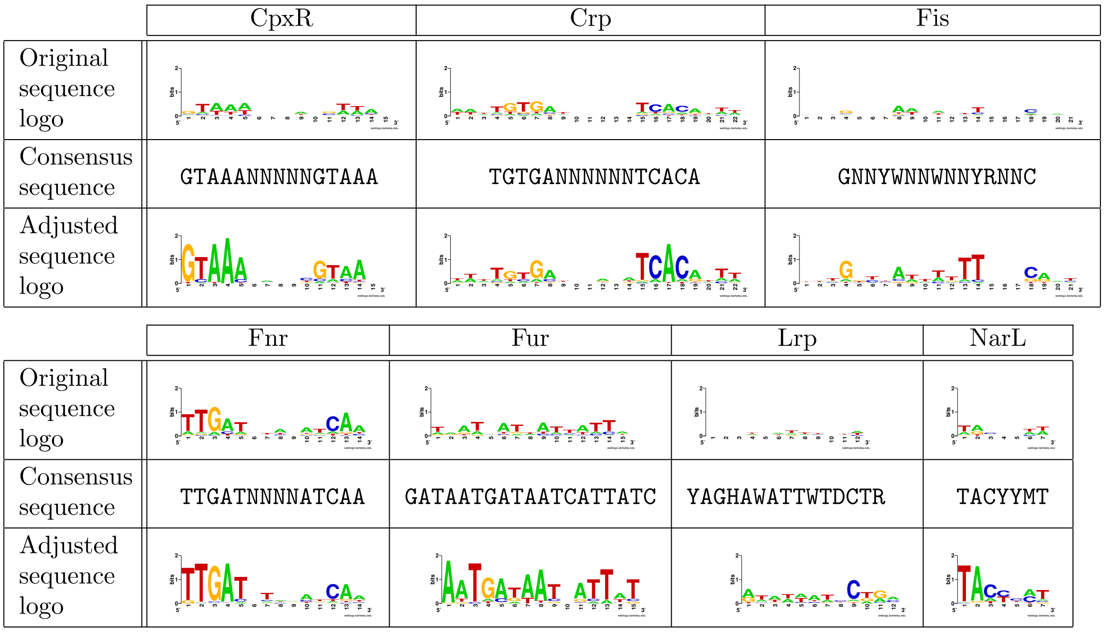


Figure 10.1: Comparison of BS conservation showing the original sequence logos, the consensus sequences for the TFs obtained from the literature [De Wulf et al., 2002, Körner et al., 2003, Pan et al., 1996, Baichoo and Helmann, 2002, Cui et al., 1995, Maris et al., 2005], and the adjusted sequence logos for the data sets of the TFs CpxR, Crp, Fis, Fnr, Fur, Lrp, and NarL. We find in all seven cases that 1) the adjusted sequence logos show a higher conservation than the original sequence logos, 2) the adjusted sequence logos are more similar to the consensus sequences than the original sequence logos, and 3) clear motifs can be recognized in the adjusted sequence logos of the TFs CpxR, Fur, and NarL that could not be recognized in the original sequence logos.

10.2.2 Validation of MotifAdjuster results for NarL

To evaluate the previous results, we choose NarL as example and examine the proposed reannotations of MotifAdjuster for this case. The nitrate regulator NarL of *Escherichia coli* is one of the key factors controlling the upregulation of the nitrate respiratory pathway and the downregulation of other respiratory chains. In the absence of oxygen, the energetically most efficient anaerobic respiratory chain uses nitrate and nitrite as electron acceptors [Unden and Bongaerts, 1997]. Detection and adaption to extracellular nitrate levels are accomplished by complex interactions of a double two-component regulatory system, which consists of the homologous sensory proteins NarQ and NarX, and the homologous TFs NarL and NarP. Depending on the BS arrangement and localization relative to the TSS, NarL and NarP act as activators or repressors, thereby enabling a flexible control of the expression of nearly 100 genes.

CoryneRegNet stores 74 NarL BSs each of length 7 bp (Table 10.1). Out of these 74 BSs, only 36 are considered as accurate by MotifAdjuster, whereas 38 are considered to be questionable. In 25 cases, MotifAdjuster proposes to switch the strand orientation of the BS, in 5 cases it proposes to

	No strand switch	**Strand switch**
No position shift	36	25
Position shift	5	6
removed	2	

Table 10.2: Application of MotifAdjuster to the set of 74 NarL BSs results in adjustments suggested for 38 of these BSs. Two BSs are proposed to be removed from the data set. Out of the remaining 36 BSs, 25 BSs are labeled with a wrong strand annotation but a correct position, and five BSs are proposed to have a correct strand annotation but a wrong position. For six BSs both strand annotation and position are proposed to be wrong.

shift the location of the BS, and for 6 BSs it proposes both a switch of strand orientation and a shift of position. In addition, two BSs are proposed for removal. We present a summary of these results in Table 10.2, and we summarize in Table 10.3 those 13 BSs of the regulator NarL where MotifAdjuster proposes to shift the location of the BS or to remove it from the databases.

To evaluate the accuracy of MotifAdjuster, we check the original literature [Kaiser and Sawers, 1995, Li et al., 1994, Darwin et al., 1997, Golby et al., 1998, Darwin et al., 1996] for each of the 13 questionable BS candidates. Comparing both, we find that in all cases but one (BS of gene *b1224*) the proposed annotation agrees with those in the literature. That is, in 12 of 13 cases signaled by MotifAdjuster as being questionable, the detected error was indeed caused by an inaccurate transfer from the original literature into the gene regulatory databases RegulonDB and CoryneRegNet. Out of those 12 questionable BSs, 10 BSs are correctly proposed to be shifted, and two are correctly proposed to be removed.

Turning to the BS of the gene *b1224*, we find it is published as given in the databases [Li et al., 1994], in contrast to the proposal of MotifAdjuster. However, [Darwin et al., 1996] report that a mutation of this BS has little or no effect on the expression of *b1224*. Hence, the proposal could possibly be correct, and the BS could be shifted or even be deleted.

In addition, MotifAdjuster checks the strand annotation of BSs and proposes strand switches if needed. In order to validate these annotations, we cannot use the annotations from RegulonDB and CoryneRegNet, since these databases contain all BSs in $5' \to 3'$ direction relative to the target gene. Hence, we consult annotation experts at the Center for Biotechnology in Bielefeld to reannotate the strand orientation of the BSs manually, and we compare the results with those of MotifAdjuster. Interestingly, we find that the strand orientations proposed by MotifAdjuster are in perfect (100%) agreement with the manually-curated strand orientations. As an independent test of the efficacy of MotifAdjuster for NarL BSs, we use the manually annotated BSs provided by the PRODORIC database [Münch, 2009]. Remarkably, we find also in this case that the results of MotifAdjuster perfectly agree with the annotations.

Another hint that the proposed adjustments of MotifAdjuster could be reasonable is based on the observation that NarL and NarP homodimers bind to a 7-2-7' BS arrangement [Maris et al., 2005], an inverted repeat structure consisting of a BS on the forward strand, a 2 bp spacer, and a BS on the

ID	**name**	**BS**	**Lit.**	**Occ.**	**Shift**	**Strand**	**Adj. BS**
b0904	*focA*	AATAAAT	[Kaiser and Sawers, 1995]	1	+1	reverse	TATTTAT
b0904	*focA*	ATAATGC	[Kaiser and Sawers, 1995]	1	+1	forward	TAATGCT
b0904	*focA*	ATATCAA	[Kaiser and Sawers, 1995]	1	+1	forward	TATCAAT
b0904	*focA*	CAACTCA	[Kaiser and Sawers, 1995]	1	+1	forward	AACTCAT
b0904	*focA*	CATTAAT	[Kaiser and Sawers, 1995]	1	+1	reverse	TATTAAT
b0904	*focA*	GATCGAT	[Kaiser and Sawers, 1995]	1	+1	reverse	TATCGAT
b0904	*focA*	GTAATTA	[Kaiser and Sawers, 1995]	1	+1	forward	TAATTAT
b0904	*focA*	TATCGGT	[Kaiser and Sawers, 1995]	1	+1	reverse	TACCGAT
b0904	*focA*	TTACTCC	[Kaiser and Sawers, 1995]	1	+1	forward	TACTCCG
b1223	*narK*	CACTGTA	[Li et al., 1994]	0	–	–	–
b1224	*narG*	TAGGAAT	[Li et al., 1994]	1	+1	reverse	AATTCCT
b4070	*nrfA*	TGTGGTT	[Darwin et al., 1997]	1	+1	reverse	TAACCAC
b4123	*dcuB*	ATGTTAT	[Golby et al., 1998]	0	–	–	–

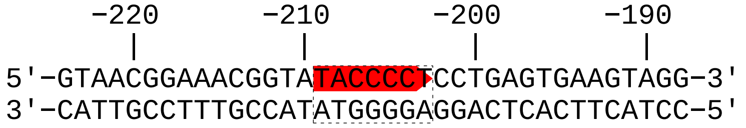
Table 10.3: Annotated NarL BSs where MotifAdjuster proposes either to shift the BS or to remove it from the data set. Columns 1-3 contain gene ID, gene name, and the BS (as stored in the database). Column 4 indicates the original literature related to this BS. The following three columns (5-7) comprise the three possible adjustments suggested by MotifAdjuster, removal, shift, and strand orientation (relative to the target gene). In column 5, a value of 0 indicates that the BS is proposed for removal, and in column 6, a positive (negative) value denotes a shift of the BS to the right (left). Finally, column 8 provides the adjusted BS. Interestingly, we find that the two BSs that are proposed to be removed are not mentioned in the original literature and in 10 out of the 11 cases the shifted BS is consistent to the BS published in the original literature. In addition, MotifAdjuster also proposes to switch the BS strand in six of the 11 cases.

reverse complementary strand. NarP exclusively binds as homodimer to this 7-2-7' structure. While NarL monomers can also bind to a variety of other heptamer arrangements, NarL dimerization at 7-2-7' BSs allows for high-affinity DNA-binding. Instances of this 7-2-7' structure have been reported for four genes, *fdnG*, *napF*, *nirB*, and *nrfA* [Darwin et al., 1997, Maris et al., 2005]. In contrast to this observation, all BSs in CoryneRegNet as well as RegulonDB are annotated to be on the forward strand, including the second half of the inverted repeat. When applied to these four genes, MotifAdjuster proposes all heptamers of the second half of the 7-2-7' structure to be switched to the reverse strand, in agreement with [Darwin et al., 1997, Maris et al., 2005]. In addition, MotifAdjuster proposes six additional BSs with a 7-2-7' BS arrangement, located in the upstream regions of the genes *adhE*, *aspA*, *dcuS*, *frdA*, *hcp*, and *norV*.

10.2.3 Prediction of a novel NarL binding site

After investigating to which degree MotifAdjuster is capable of finding errors in existing gene-regulatory databases, it is interesting to test if MotifAdjuster could be helpful for finding novel BSs. The flexibility of BS arrangements and the low motif conservation complicate the computational and manual prediction of NarL BSs by curation teams. This results in several cases where promoter regions are experimentally verified to be bound by NarL, but where

(a) New NarL BS in *torC* promoter


-220
-210
-200
-190
5'-GTAACGGAAACGGTATACCCCTCCTGAGTGAAGTAGG-3'
3'-CATTGCCTTTGCCATATGGGGAGGACTCACTTCATCC-5'

(b) Histogram of all NarL BS positions relative to the start codon

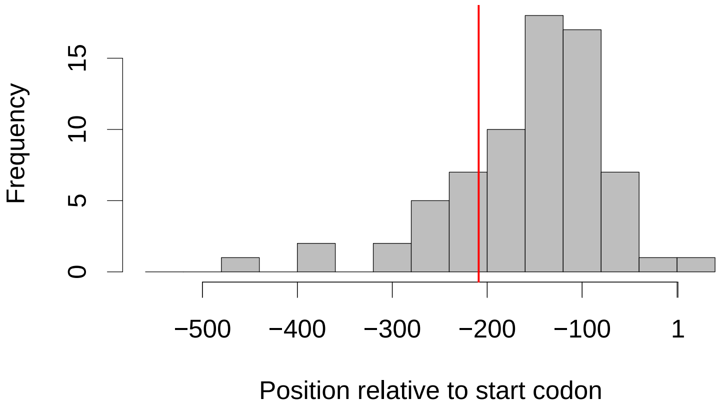


Figure 10.2: The NarL BS `TACCCT` is located on the forward strand with respect to the target operon *torCAD* starting at position −209 bp (red color). All positions are relative to the first nucleotide of the start codon of *torC*. Figure 10.2(a) shows the fragment of the upstream region of the *torCAD* operon containing the NarL BS predicted by the PWM model trained on the adjusted data set. Figure 10.2(b) shows the histogram of all positions of NarL BSs in the database. The red line indicates the position of the predicted BS.

no NarL BS could be detected[Overton et al., 2006, Constantinidou et al., 2006]. Examples of such genes are *caiF* [Eichler et al., 1996], *torC* [Iuchi and Lin, 1987], *nikA* [Rowe et al., 2005], *ubiC* [Kwon et al., 2005], and *fdhF* [Wang and Gunsalus, 2003]. We extract the upstream regions of these genes, where an upstream sequence is defined by CoryneRegNet as the sequence between positions −560 bp and +20 bp relative to the first position of the annotated start codon of the first gene of the target operon. In addition, we extract those upstream regions of *Escherichia coli* that belong to operons not annotated as being regulated by NarL (background data set).

We investigate if we can now detect NarL BSs based on the adjusted data set that could not be detected based on the original data set from CoryneRegNet. For that purpose, we estimate the parameters of the PWM model on the adjusted data set as proposed by MotifAdjuster and the parameters of the homogeneous Markov model on the background data set. From the adjusted PWM, we build a strand model with the same probability for each strand. For the classification of an unknown heptamer $\underline{x}$, we build a simple log-likelihood ratio $r(\underline{x})$ with these parameters. For an upstream region, we compute $r_{\max}$ defined as the highest log-likelihood ratio of any heptamer $\underline{x}$ in this upstream region. We compute the p-value of a potential BS $\underline{x}$ with value $r(\underline{x})$ as fraction of the background sequences whose $r_{\max}$-values exceed $r(\underline{x})$.

Using this classifier, a significant NarL BS can now be detected in the upstream region of *torC*. Figure 10.2a shows the double-stranded DNA fragment with the predicted BS (`TACCCCT`) located on the forward strand starting at −209 bp relative to the start codon, and at −181 bp relative to the annotated TSS [Méjean et al., 1994]. The distance of the predicted BS to the start codon agrees with the distance distribution of previously known NarL BS (Figure 10.2b), providing another additional evidence for the proposed BS. This finding closes the gap between sequence analysis and gene expression studies, as the *torCAD* operon consists of three genes which are essential for the Trimethylamine N-oxide (TMAO) respiratory pathway [Méjean et al., 1994]. TMAO is present as an osmoprotector in tissues of invertebrates and can be used as respiratory electron acceptor by *Escherichia coli*. Transcriptional regulation of this operon by NarL binding to the proposed BS would explain nitrate-dependent repression of TMAO-terminal reductase (TorA) activity under anaerobic conditions [Iuchi and Lin, 1987], thereby linking TMAO and nitrate respiration.

10.3 Conclusions

Gene-regulatory databases, such as AGRIS, AthaMap, CoryneRegNet, CTCFBSDB, JASPAR, ORegAnno, PRODORIC, RegulonDB, SCPD, TRANSFAC, TRED, or TRRD store valuable information about gene-regulatory networks including TFs and their BSs. These BSs are usually manually extracted from the original literature and subsequently stored in databases. The whole pipeline of wet-lab BS identification and annotation, publication, and manual transfer from the scientific literature to data repositories is not just time-consuming but also error-prone, leading to many false annotations currently present in databases.

MotifAdjuster is a software that supports the (re-)annotation process of BSs *in silico*. It can be applied as quality assurance tool for monitoring putative errors in existing BS repositories and for assisting with a manual strand annotation. In contrast to existing de-novo motif discovery algorithms, MotifAdjuster allows the user to specify the probability of finding a BS in a sequence and to specify a non-uniform shift distribution.

We apply MotifAdjuster to seven data sets of BSs for the TFs CpxR, Crp, Fis, Fnr, Fur, Lrp, and NarL with a total of 536 BSs, and we find 51 BSs proposed for removal and 134 BSs proposed for shifts. In total, this results in 34.5% of the BSs being proposed for adjustments. We choose NarL as example to examine the proposed reannotations of MotifAdjuster. Checking the original literature for each of the 13 cases shows that the proposed deletions and shifts of MotifAdjuster are in agreement with the published data. Comparing the strand annotation of MotifAdjuster with independent information indicates that the proposals of MotifAdjuster are in accordance with human expertise. Furthermore, MotifAdjuster enables the detection of a novel BS responsible for the regulation of the *torCAD* operon, finally augmenting experimental evidence of its NarL regulation.

We make MotifAdjuster available for the scientific community as part of the open-source Java library Jstacs [Keilwagen et al., 2008], which allows an easy application, automation, and extension at `http://www.jstacs.de/index.php/MotifAdjuster`.

Chapter 11

Discriminative de-novo motif discovery utilizing positional preference

In the previous chapter, we investigate the quality of TFBS annotation, which is based on the pipeline of wet-lab experiments, scientific publications, and subsequent manual transfer into transcriptional gene-regulatory databases by curation teams. However, the basis of such databases are combinations of different wet-lab experiments that are used to obtain target regions of TFs. The regions identified by these experimental methods are large and not limited to the TFBSs solely, so one challenge in computational biology is the identification of TFBSs in these regions. Typically, de-novo motif discovery tools, which take a set of target regions with unknown binding motif and unknown BSs as input, are used for predicting putative binding motifs and the corresponding putative TFBSs simultaneously.

A wealth of de-novo motif discovery tools has been developed over the last decades including, for example, Gibbs Sampler [Lawrence et al., 1993, Thompson et al., 2003, Thompson et al., 2007], MEME [Bailey and Eklan, 1994], Weeder [Pavesi et al., 2001], Improbizer [Ao et al., 2004], DME [Smith et al., 2005], DEME [Redhead and Bailey, 2007], or A-GLAM [Kim et al., 2008]. These tools differ by the learning principle employed to infer the model parameters and by their capability of learning the position distribution of the BSs from the data.

Many de-novo motif discovery tools including Gibbs Sampler [Lawrence et al., 1993, Thompson et al., 2003, Thompson et al., 2007], MEME [Bailey and Eklan, 1994], Weeder [Pavesi et al., 2001],Improbizer [Ao et al., 2004], and A-GLAM [Kim et al., 2008] use generative learning principles for discovering statistically over-represented motifs from a target data set. However, the discovered motifs often turn out to be over-represented also in the rest of the genome, making the predictions not specific for the target data set under investigation. In order to overcome this limitation, de-novo motif discovery tools using discriminative learning principles such as DME [Smith et al., 2005] and DEME [Redhead and Bailey, 2007] have been developed during the last years. These tools utilize an additional control data set expected to contain no or only few BSs of the motif of interest for discovering differentially abundant motifs.

Many de-novo motif discovery tools including Gibbs Sampler [Lawrence et al., 1993, Thompson et al., 2003, Thompson et al., 2007], MEME [Bailey and Eklan, 1994], Weeder [Pavesi et al., 2001], DME [Smith et al., 2005] and DEME [Redhead and Bailey, 2007]

		position distribution	
		fixed	**learned from data**
learning principle	**generative**	Gibbs Sampler MEME Weeder	Improbizer A-GLAM
	discriminative	DME DEME	

Table 11.1: Characterization of de-novo motif discovery tools. Rows indicate the learning principle, and columns indicate if the position distribution can be learned from the data. Weeder uses a consensus-based representation of the motif, while the other tools use probabilistic models. Interestingly, none of the tools is capable of searching for differentially abundant BSs and learning the positional distribution simultaneously.

use a fixed position distribution, chosen to be a uniform distribution in most cases. Motivated by the observation that TFBSs often occur clustered, i. e., not uniformly distributed along the promoters [Hughes et al., 2000, Thompson et al., 2003, Wray et al., 2003], tools such as Improbizer [Ao et al., 2004] and A-GLAM [Kim et al., 2008] have been developed that are capable of learning the positional distribution from the data.

In Table 11.1, we categorize the above-mentioned tools according to their capability of (i) finding differentially abundant motifs and (ii) learning the position distribution from the data. None of these tools works perfectly [Tompa et al., 2005, Sandve et al., 2007], but typically de-novo motif discovery tools utilizing a discriminative learning principle outperform those utilizing a generative learning principle [Elemento et al., 2007], and de-novo motif discovery tools capable of learning the positional preference of TFBSs typically outperform those with a fixed distribution [Kim et al., 2008]. Interestingly, no algorithm has been developed that combines both features. This chapter is based on [Keilwagen et al., 2010a] where we introduce Dispom, a discriminative de-novo position distribution motif discovery tool that is capable of modeling the positional preference of TFBSs.

Similar to other discriminative tools such as DEME or DME, Dispom utilizes a control data set assumed to contain no or few BSs of interest in addition to the target data set. And similar to Improbizer and A-GLAM, Dispom learns the distribution of binding positions from the data simultaneously with the parameters of the motif model. In addition, Dispom uses a heuristic during parameter learning for adapting the length of the binding motif, which is often unknown in advance, and for compensating phase shifts [Lawrence et al., 1993], which frequently occur in many de-novo motif discovery tools.

The remainder of this chapter is structured as follows. In the next section, we describe Dispom and the data used in the subsequent case studies. In section *Results and Discussion*, we compare the performance of Dispom to that of seven commonly used de-novo motif discovery tools based on 18 benchmark data sets investigating whether these tools are capable of finding motifs with and without positional preference. Subsequently, we test whether Dispom is capable of finding two motif types simultaneously. Finally, we apply Dispom to a data set of promoters of auxin-responsive genes in a cell suspension culture of *Arabidopsis thaliana*. We compare the motif found by Dispom with the canonical auxin-responsive element and test how specific these motifs are at predicting auxin-responsive genes for an independent data set.

11.1 Approach

In this section, we briefly describe Dispom including the probabilistic model, the parameter learning principle, and a heuristic for avoiding an often occurring problem called phase shifts and for the inference of the motif length. Subsequently, we explain how we compare the performance of de-novo motif discovery tools, and we describe the data sets used in the case studies.

Dispom

Dispom is based on the extended zero or one occurrence per sequence model, which we describe in subsection *Extended ZOOPS models* (page 31). This model includes two hidden variables u_1 and u_2 that encode the motif type located in the sequence and the stat position of the BS, respectively. Given that there is a BS of motif type $u_1 > 0$ at position u_2 in sequence $\underline{x}$, this yields the probability

$$\begin{aligned} P^{\mathrm{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(\underline{x}, u_1, u_2 \middle| \underline{\lambda}\right) := {} & P^{\mathrm{eZOOPS}(\underline{\mathcal{M}},\underline{\mathcal{S}},\mathcal{F})}\left(u_1 \middle| \underline{\lambda}\right) \cdot P^{\mathcal{S}_{u_1}}\left(u_2 \middle| \underline{\lambda}^{(\mathcal{S}_{u_1})}\right) \cdot P^{\mathcal{F}}\left(\underline{x}_{1,\ldots,u_2-1} \middle| \underline{\lambda}^{(\mathcal{F})}\right) \\ & \cdot P^{\mathcal{M}_{u_1}}\left(\underline{x}_{u_2,\ldots,u_2+w_{u_1}-1} \middle| \underline{\lambda}^{(\mathcal{M}_{u_1})}\right) \cdot P^{\mathcal{F}}\left(\underline{x}_{u_2+w_{u_1},\ldots,L} \middle| \underline{\lambda}^{(\mathcal{F})}\right) \end{aligned} \tag{11.1}$$

where $\mathcal{F}$ denotes the model for the flanking sequence, $\mathcal{M}_{u_1}$ denotes the model for the motif u_1, and $\mathcal{S}_{u_1}$ denotes the model for the start position of motif u_1.

Similar to other tools, Dispom uses a PWM for each motif model $\mathcal{M}$ for both DNA strands and a homogeneous Markov model of order 0 as flanking sequence model $\mathcal{F}$. In contrast to other tools, Dispom utilizes a mixture of a skew normal and a uniform distribution as position model $\mathcal{S}$. The choice is motivated by the observation that a Gaussian distribution decays quite rapidly, and, hence, BSs further apart from the mean of the Gaussian are often overlooked. Similarly, the choice of the skew normal instead of a Gaussian distribution is inspired by the expectation that if the mean of the Gaussian is close to the TSS there might be a skew of the distribution. Further details about the model can be found in subsection *Extended ZOOPS models* (page 31).

For predicting BSs of motif $u_1 > 0$ in a sequence $\underline{x}$, we compute the probability given in Equation 11.1 for each possible position u_2 of $\underline{x}$. We also compute these probabilities for each possible position in each sequence of the control data set yielding a background distribution of probabilities. We define the p-value of position u_2 being erroneously predicted as a BS as the fraction of the probabilities that exceed the probability at position u_2 according to the background distribution. We finally define a threshold ψ on the p-values, and predict all positions u_2 of a sequence with $pP\left(\underline{x}, u_1, u_2 \middle| \underline{\lambda}\right) < \psi$ as starting positions of a BS.

The goal of de-novo motif discovery is to infer proper parameters of the motif model from a set of target regions and, in case of discriminative approach, an additional set of control regions. While tools like MEME and Improbizer use the generative MAP principle for learning the parameters based on a target data set, DME, DEME, and Dispom use a discriminative learning principle. Specifically, Dispom uses the MSP principle described in section *Learning principles* (page 10) where we use the eZOOPS model for the foreground class, and we follow the proposal of [Redhead and Bailey, 2007] and use a homogeneous Markov model of order 0 for the background class.

When learning the parameters using a Bayesian learning principle as for MAP or MSP, we need a prior density for the parameters. In the cases studies on the benchmark data sets as well as for the auxin data set, we use the prior density described in chapter *Priors* (page 34) with the hyper-parameters listed below. These hyper-parameters are chosen according to a uniform pseudo-data sets.

- The parameters of each PWM model obtain the same hyper-parameters, $\alpha_{\ell,b}^{\mathrm{PWM}_u} = 1$ for $\ell \in [1, w_u]$ and $b \in \Sigma$.
- We expect that the BSs of each motif are located on both strands with equal probability, $\alpha_b^{\mathcal{M}_u} = 2$ for $b \in \{0, 1\}$.
- We expect that the are some sequences that do not contain any BS, $\alpha_0 = 1$, which corresponds to an expectation of 20% and 11% of sequences that do not contain a BS when searching for one or two motifs, respectively.
- We expect that that position distribution of each motif is dominated by the uniform distribution, $\alpha_0^{\mathcal{S}_u} = 3$ and $\alpha_1^{\mathcal{S}_u} = 1$.
- For the skew normal component of each position distribution, we expect that it does not deviate much from a uniform distribution expressed by the following parameters.
  - The mean is located about 250 bp with an standard deviation of 500 bp, $\alpha_{0,0}^{\mathcal{S}_{u,1}} = 250$ and $\alpha_{0,1}^{\mathcal{S}_{u,1}} = 500$.
  - The standard deviation is about 150, $\alpha_{1,0}^{\mathcal{S}_{u,1}} = 0.5$ and $\alpha_{1,1}^{\mathcal{S}_{u,1}} = 0.5 \cdot 150^2$.
  - The skew parameter is about 0 with a standard deviation of 1, $\alpha_{2,0}^{\mathcal{S}_{u,1}} = 0$ and $\alpha_{2,1}^{\mathcal{S}_{u,1}} = 1$.
- Finally, we expect that the parameters of the flanking model do not deviate very much from a uniform distribution. Based on the ESS of the eZOOPS model, this yields the hyper-parameter $\alpha_b^{\mathcal{F}} = L \cdot \frac{\alpha_0 + \sum_u \alpha_{\cdot}^{(\mathrm{PWM}_u)}}{4} - \sum_u \sum_{\ell=1}^{w_u} \alpha_{\ell,b}^{(\mathrm{PWM}_u)}$ for $b \in \Sigma$. For an ESS of the eZOOPS model of 5 and 9 for one and two motif types, respectively, obtain a value of 610 and a value of 1095 for an initial motif length of $w_u = 15$.

We obtain estimates of the parameters of Dispom by numerical maximization [Wallach, 2004] of Equation (3.7a). Since the eZOOPS model implements a non-convex supervised posterior it may get trapped in local optima or saddle points. One prominent type of local optima are so-called phase shifts where the BSs are only covered by a part of the motif model. Besides starting Dispom multiple times, we implement a heuristic that helps reducing this problem and at the same time allows to adjust the motif length. We describe this heuristic in the next subsection.

Phase shift and adjustment of motif length

Similar to other models, the eZOOPS model is prone to phase shifts. We address the problem by allowing the motif model to be shifted, shrunken, or expanded using a heuristic step. We ensure that these heuristic steps do not lead to cycles by keeping a history of performed steps. After each numerical maximization of the parameters, Dispom performs a number of steps to decides whether a heuristic step should be performed or not.

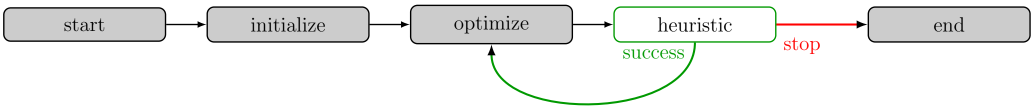
First, we compute the number of foreground sequences B predicted to contain at least one BS. Second, we test each shift s of the motif of at most half of the motif length by shifting the motif model, optimizing the parameters with 10 steps of numerical optimization, and computing the number of foreground sequences B_s predicted to contain at least one BS. Finally, we compare the B and B_s, and determine for both shift directions the number of insignificant positions by finding the shift s with maximal $B_s \geq 0.8 \cdot B$. From these insignificant positions, the heuristic proposes a promising shift or length modification of the motif model described in the following.

The promising modifications are determined by the following rules: Let n_ℓ be the number of insignificant positions on the left side of the motif, and let n_r be the number of insignificant positions on the right side of the motif. If we find no insignificant position on either side of the motif, i. e., $n_\ell = n_r = 0$, we expand the motif to the initial length by appending additional positions to the right side, or if this configuration is already stored in the history, to the left side of the motif. If the initial length is already reached or even exceeded, we expand the motif by one position on both sides, if allowed by the history. Otherwise, i. e., $n_\ell > 0$ or $n_r > 0$, we first try to shift the motif, such that the larger number of insignificant positions is shifted out of the model, i. e., if $n_\ell > n_r$, we shift the model by n_ℓ positions to the right, and vice versa. If the shift operation did not succeed, we shrink the motif by removing n_ℓ positions from the left side and n_r positions from the right side of the motif. We restrict the minimal length of the motif model to 1 to prevent the complete elimination of the motif. Positions that are added to the motif model are initialized with a uniform distribution of nucleotides before we start the numerical optimization. The cycle of heuristic steps and consequent optimization is stopped if none of the promising modifications is still allowed by the history. Figure 11.1 shows the complete flow diagram of Dispom as well as a detailed workflow for the heuristic described above.

It is clear that Dispom can get trapped in local optima or saddle points despite of these heuristic steps. Hence, we start Dispom including the heuristic steps 50 times, and choose that parameter set $\underline{\lambda}$ with the highest supervised posterior for the subsequent prediction of BSs.

Due to these repeated starts of the numerical optimization, the runtime of Dispom is considerable. However, the runtime of a single optimization run highly depends on number and length of the input sequences. Besides this technical limitation, there are also some biological limitations of Dispom. For instance, Dispom shares the limitation of several other tools, as for instance MEME, Weeder, Improbizer, DME, and DEME, to model at most one BS per sequences, since it uses the eZOOPS

(a) Flow diagram for Dispom.



(b) Flow diagram for the heuristic used in Dispom.

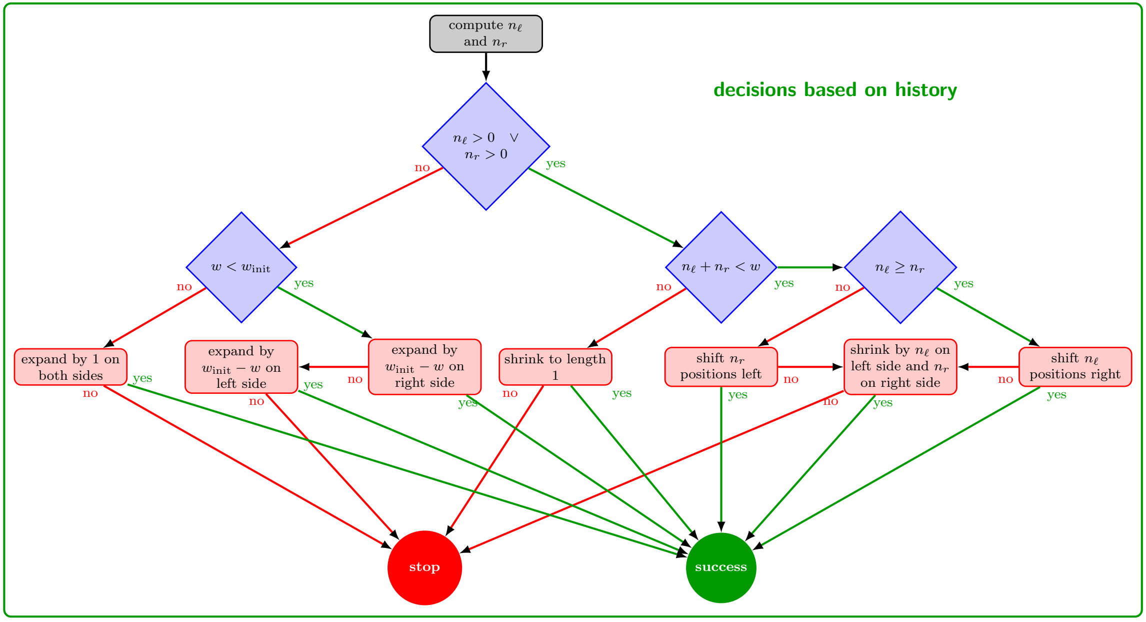

compute n_ℓ and n_r
decisions based on history
$n_\ell > 0 \quad \vee \quad n_r > 0$
no
yes
$w < w_{\text{init}}$
$n_\ell + n_r < w$
$n_\ell \geq n_r$
expand by 1 on both sides
expand by $w_{\text{init}} - w$ on left side
expand by $w_{\text{init}} - w$ on right side
shrink to length 1
shift n_r positions left
shrink by n_ℓ on left side and n_r on right side
shift n_ℓ positions right
stop
success

Figure 11.1: Flow diagram for Dispom. Figure a) shows the flow diagram for Dispom, whereas Figure b) shows the flow diagram of the heuristic that is used to shift, shrink, or expand the motif.

model. In addition, Dispom does only work on sequences of identical length due to the position distribution of the BSs that is learned from the data.

Comparison of de-novo motif discovery tools

Prediction performance of different de-novo motif discovery tools is usually compared using the nucleotide recall (nRs) and the nucleotide precision (nP), which are also referred to as *nucleotide sensitivity* and *nucleotide positive predictive value*, respectively [Tompa et al., 2005]. Let the *true positives* TP be the number of positions correctly predicted to be covered by BSs according to the annotation, let M be the number of positions covered by BSs, and let $\bar{M}$ be the number of positions predicted to be covered by BSs. Then, nR is defined as the fraction of correctly predicted nucleotides out of all nucleotides of all annotated BSs, nR $:= TP/M$, and nP is defined as the fraction of correctly predicted nucleotides out of all nucleotides of all predicted BSs, nP $:= TP/\bar{M}$.

nR and nP depend on parameters of the tools, such as the threshold ψ. For this reason, the values of nP and nR may be very different, and it is hard to compare the performance of different tools using only a single pair of nR and nP. Typically, some tools have high values of nR and low values of nP, while other tools have low values of nR and high values of nP, complicating a one-to-one comparison of their accuracy. Hence, we vary the threshold ψ, which is connected to the number of predictions, and obtain a series of pairs of nR and nP for each tool. Plotting these values of nP against nR yields the *nucleotide precision recall (*nPR*) curve*, which is a natural generalization of the PR curve presented in section *Classification measures* (page 17). For this reason, the nPR curve is more suitable for assessing imbalanced data sets than extension of commonly used ROC curve [Raghavan et al., 1989, Davis and Goadrich, 2006, Sonnenburg et al., 2006, Sonnenburg et al., 2007]. Since most of the tools compared to Dispom have fixed internal thresholds, we can only obtain partial curves for these tools, which still provide more information than single pairs of nP and nR values.

Data sets

First, we describe 18 benchmark data sets with known positions of the BSs of one motif used for comparing the prediction performance of the tools listed in Table 11.1 and Dispom. Second, we describe three benchmark data sets with known positions of the BSs of two motif used for testing the eZOOPS with two motif types. Finally, we describe two data sets of auxin-responsive genes of *Arabidopsis thaliana* [Paponov et al., 2008] used for applying Dispom to a real-life problem where the true BSs are unknown.

Benchmark data sets with known positions of the binding sites of one motif

Several benchmark tests have been used for comparing different de-novo motif discovery tools over the last years. These comparisons are based on annotated BSs [Tompa et al., 2005, Kim et al., 2008] or on binding motifs [Linhart et al., 2008]. For an in-depth comparison of the performance of different de-novo motif discovery tools, a comparison based on BSs is more informative than a comparison based on binding motifs, since comparing binding motifs does not address the prediction accuracy of individual BSs. In [Tompa et al., 2005, Kim et al., 2008] small data sets with long sequences and annotated BSs have been used.

To assess the significance of predictions depending on the amount of data, we estimate the probability to find at least one common subsequence with at most one mismatch in N random sequences. We download Arabidopsis promoter regions from TAIR [Swarbreck et al., 2008] and extract the upstream 2000 bp for each promoter following [Kim et al., 2008]. For varying N, we randomly sample N promoter sequences and determine the length of the longest common subsequence of the N sequences with at most one mismatch. In Figure 11.2, we show the result of repeating this procedure 1000 times where we plot the length of the longest subsequence against the p-value. For N equal to 5, 10, and 100, and a subsequence length of up to 10, 9, and 8 bp, respectively, we find a p-value of 1.[1] Using less conservative approaches as for instance the ZOOPS model, we are not restricted to one mismatch per BS, and we do not require that each sequence contains a BS, making the problem even worse. Hence,

---

[1] We perform the same simulation for human promoters obtaining similar results.

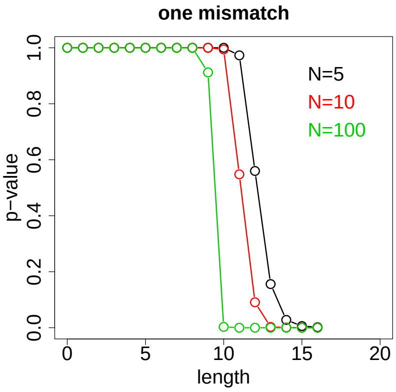


Figure 11.2: *P*-value distribution for the length of common subsequences in promoter regions of *Arabidopsis thaliana.* Illustration of the *p*-value distribution for the length of common subsequences in promoter regions of *Arabidopsis thaliana* for N equals 5, 10, and 100 sequences, respectively, and one allowed mismatch.

finding binding motifs of length 9 bp in data sets with few long sequences is usually insignificant, and we recommend to use benchmark data sets with more or shorter sequences.

Hence, we use the seven largest data sets of known TFBSs from the JASPAR database [Bryne et al., 2008]. These data sets cover TFs of mammals (three data sets: MA0048, MA0052, MA0077), plants (three data sets: MA0001, MA0005, MA0054), and insects (one data set: MA00115). Second, we download the promoters of the corresponding organisms for each of these seven data sets. In case of data set MA0054 from *Petunia x hybrida*, we use promoters of *Arabidopsis thaliana*, since promoters for *Petunia x hybrida* are not available. For each promoter data set, we extract the upstream 500 bp. Third, we create two data sets for each data set of TFBSs and the corresponding promoters by randomly choosing a subset of promoters and subsequently implanting BSs randomly either from a uniform or a Gaussian distribution. For each Gaussian distribution, we draw the mean and the standard deviation uniformly from the intervals $[20, 480]$ and $[20, 80]$, respectively. Each data set created consists of 70% of promoters with one BS on either the forward or the reverse complementary strand and of 30% of promoters with no implanted BS. Finally, for each of these data sets we draw another data set with the same number of promoters not containing any implanted BS, which is used as control data set for discriminative de-novo motif discovery tools.

For testing the discriminative power of the eight de-novo motif discovery tools, we build four additional pairs of data sets. We implant a decoy BS of the data set MA0052 in each sequence in the target and control data sets. Additionally, we implant BSs of the data set MA0048 only into the target data set according to the above procedure. We obtain four pairs of data sets by drawing the positions of the real and the decoy BSs within the promoter either from a uniform or a Gaussian distribution as described above resulting in 18 pairs of data sets in total.

Among these 18 pairs of data sets, we denote the nine data sets with BSs implanted by Gaussian distributions as *Gaussian data sets*, and we denote the remaining nine data sets as *uniform data sets*.

For the assessment of the nPR curves, we use the implanted BS positions except border positions with an information content of less than 0.25 bit in the sequence logo of the true motif.

Benchmark data sets with known positions of the binding sites of two motifs

For testing whether the eZOOPS model is capable of learning more than one motif type, we build three benchmark data sets. We choose the BSs of MA0048 and MA0052 from *homo sapiens* which we have downloaded from JASPAR database. Similar to the procedure described above, we randomly choose 150 promoters and implant the BSs of both types on both strands where each sequence contains at most one BS. We obtain three data sets containing the BSs of MA0048 and MA0052. In the first data set, we implant the BSs of both types using two Gaussian distributions, and similar in the second data set, we implant the BS of both types using a uniform distribution. In addition, we implant the BSs of MA0048 and MA0052 using a uniform and Gaussian distribution, respectively. We use these three data sets as target data sets for Dispom, and we generate a control data set that contains 150 promoters without any implanted BS.

Data sets of auxin-responsive promoters

We use expression data of *Arabidopsis thaliana* from a cell suspension culture, because it is ideal for studying transcriptional responses to different stimuli due to its uniformity and homogeneity. The plant hormone auxin plays a critical role in virtually all aspects of plant growth and development specifically regulating the transcription of many genes [Teale et al., 2006]. Direct target genes of auxin response are known to be regulated quickly, so we select genes with a two-fold increase in gene expression after a short exposure time of 15, 30, or 60 minutes in the cell suspension culture [Paponov et al., 2008]. As an independent set of genes, we select genes up-regulated in seedlings within the same time interval of 60 minutes after treatment [Paponov et al., 2008]. We use the cell suspension data set containing 48 promoters as target data set, and we randomly select 1,000 promoters from the set of all remaining genes on the Affymetrix ATH1 microarray chip as control data set. For testing Dispom, we use the promoters of the seedling data set and of all remaining genes not used during training yielding 113 promoters and 21012 promoters, respectively. For all data sets, we use the promoter region from -500 bp to -1 bp.

11.2 Results and Discussion

In this section, we first compare the performance of the seven de-novo discovery tools A-GLAM, DEME, DME, Gibbs Sampler, Improbizer, MEME, and Weeder with that of Dispom based on 18 benchmark data sets containing experimentally verified BSs of one motif. Second, we test whether Dispom is able to find two different motifs simultaneous in specific benchmark data sets. Finally, we apply Dispom to a situation where neither the true BSs nor their motifs are known. Specifically, we apply Dispom to promoters of genes up-regulated by auxin in a cell suspension culture of *Arabidopsis* *thaliana*, we compare the motif found by Dispom with the canonical auxin-responsive element, and we investigate if the motif is also differentially abundant in the seedling data set compared to all remaining promoters.

11.2.1 Comparison of Dispom with existing tools

For testing the efficacy of Dispom, we compare it with commonly used available methods on the same data sets. First, we consider three different aspects of de-novo motif discovery for all tools. We consider the capability of de-novo motif discovery tools of

1. finding the correct BSs with unknown motif length,
2. recovering a non-uniform position distribution of the BSs in the data sets, and
3. finding differentially abundant motifs in the presence of non-specific but over-represented motifs.

For each of these issues, we consider only one specific example, and we refer to Additional File 2-5 of article [Keilwagen et al., 2010a] for the remaining results. Finally, we provide a comparison of the different de-novo motif discovery tools applied to each benchmark data set.

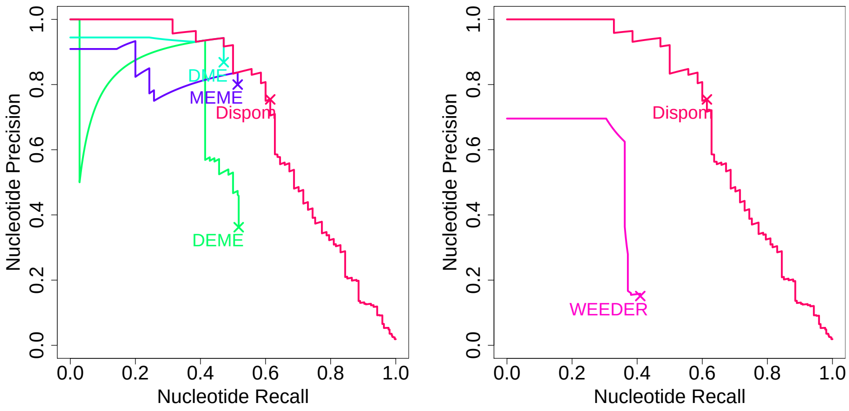
Unknown motif length

First, we consider the aspect of finding the correct motif if the motif length is unknown. In many cases, when de-novo motif discovery tools are used, the user only has a rough idea of the motif length. Hence, the user must test all potential motif lengths and decide which result is of interest, or the tool allows to infer the motif length on its own.

Here, we study the results for different de-novo motif discovery tools for the target data set containing BSs of MA0054 with a Gaussian distribution. In the first experiment, we start all tools with the correct motif length. In the second experiment, we start all tools with an initial length of 15 bp, and allow to adjust the motif length if supported by the tool. In Figure 11.3, we show the results for both cases.

For known correct motif length, we find that DEME, DME, MEME, and Dispom find the implanted motif to a certain degree, showing that these four tools are capable of finding the implanted BSs. Among these four tools, Dispom performs best, and DEME, DME, and MEME perform comparably well. However, in case of unknown motif length, we find that DEME, DME, and MEME are not capable of finding the correct motif. While DEME and DME are not capable of adjusting the motif length, MEME allows searching the motif for a range of possible motif lengths. Nevertheless, all three tools fail to find the motif if the correct motif length is not provided.

In contrast to these findings, Weeder and Dispom are capable of finding the correct motif. Interestingly, Weeder is capable of finding the motif to a certain degree, although it is not capable of finding the motif for the known motif length. Scrutinizing the motif found by Weeder, we find that it is shorter than the true motif (Additional File 5 of article [Keilwagen et al., 2010a]). In contrast, we



(a) With known motif length (b) Without known motif length

Figure 11.3: Comparison of nucleotide precision recall curves of different de-novo motif discovery tools for known and unknown motif length. Figure a) shows the nucleotide precision recall curves for the de-novo motif discovery tools provided with the correct motif length, and Figure b) shows the nucleotide precision recall curves for the de-novo motif discovery tools when the correct motif length is not provided but must be learned by the tools. For reasons of visual clarity, we do not plot the partial nucleotide precision recall curves of those tools with nR and nP below 0.1 for all available thresholds. These curves would be located in the lower left corner of both subfigures.

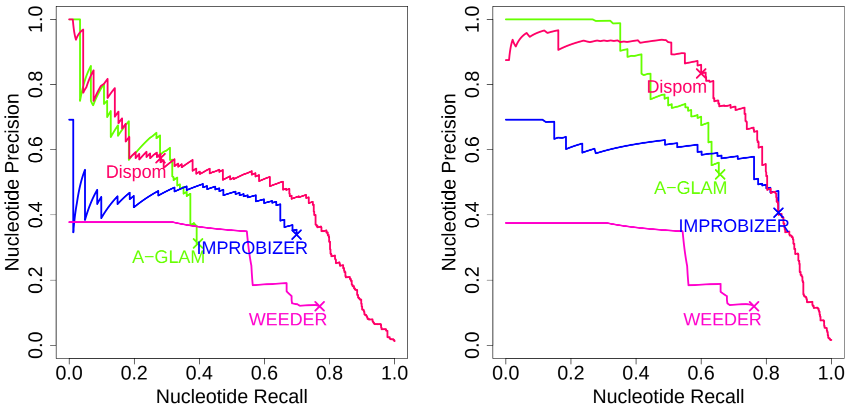
find that the performance of Dispom is very similar to the case of known motif length indicating that Dispom is capable of finding the correct motif including the motif length.

Based on these case studies, we can state that knowing the correct motif length improves de-novo motif discovery. However, in many real-life applications, the correct motif length is unknown, and many de-novo motif discovery tools suffer in this situation. Dispom with its heuristic for shrinking and expanding the motif is capable of learning the correct motif length from the data, and so, outperforms other de-novo motif discovery tools.

Non-uniform position distribution

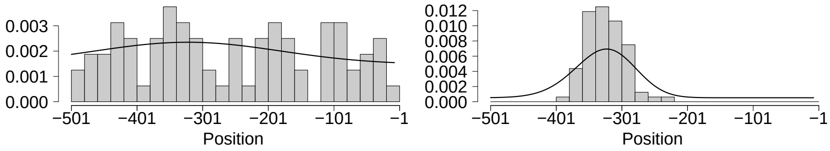
Second, we consider the aspect of recovering a non-uniform position distribution of the BSs in the data set. In many cases, BSs are not uniformly distributed over the entire promoter but rather concentrated with a TF-specific position distribution. To simulate these findings, we use the data sets for MA0015 for which we compare the results of the Gaussian data set to those obtained for the uniform data set. Since both data sets consist of exactly the same BSs and the same promoters, and only differ in the position distribution used to implant the BSs, we are able to measure the effect of modeling a non-uniform position distribution. Figure 11.4 a) and b) show the nucleotide precision recall curves for both position distributions used for implanting the BSs.

For a uniform position distribution we observe that A-GLAM, Improbizer, Weeder, and Dispom find the correct motif. Turning to the case of a Gaussian position distribution, we observe that A-GLAM, Improbizer, and Dispom are able to utilize the positional preference of BSs to substantially



(a) PR-curves for a data set with uniform position distribution

(b) PR-curves for a data set with Gaussian position distribution



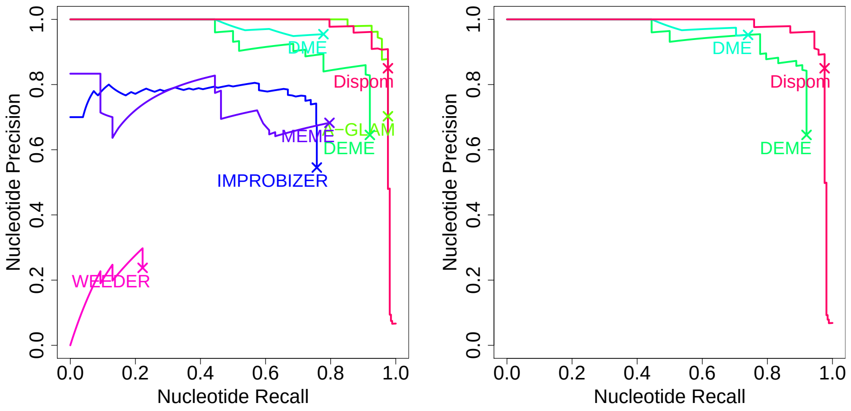
(c) Positional distributions

Figure 11.4: Comparison of nucleotide precision recall curves of different de-novo motif discovery tools for uniform and Gaussian position distribution. Figure a) shows the nucleotide precision recall curves for the de-novo motif discovery tools on the data set with uniformly placed MA0015 BSs, and Figure b) shows the nucleotide precision recall curves for the de-novo motif discovery tools on the data set with Gaussian distributed MA0015 BSs. Figure c) shows for both data sets the real distributions as histograms of start positions of the implanted BSs and the position distributions learned by Dispom. For reasons of visual clarity, we do not plot results located in the lower left corners of subfigures a) and b) (Figure 11.3).

improve their performance. In contrast to these findings, the performance of Weeder does not improve, because it does not model positional dependencies.

We analyze the performance improvements by comparing the distribution used for implanting the BSs with the distribution learned by Dispom. In Figure 11.4c), we show for both cases – the uniform and the Gaussian position distribution – a histogram for the start positions of the implanted BSs and the distribution learned by Dispom. We find that both distributions are in agreement in both cases, indicating that Dispom is capable of learning the position distribution from the data.

Based on these case studies, we can state that recovering the position distribution of the BSs from the data helps in de-novo motif discovery and the subsequent prediction of BSs. Since Dispom is able to learn peaked as well as uniform position distributions from the data, it can be used for in a wide range of applications.



(a) Without decoy motif

(b) With decoy motif

Figure 11.5: Comparison of nucleotide precision recall curves of different de-novo motif discovery tools for a data set with and without decoy motif. Figure a) shows the nucleotide precision recall curves for the de-novo motif discovery tools on the data set without implanted decoy motif, and Figure b) shows the nucleotide precision recall curves for the de-novo motif discovery tools on the data set with implanted decoy motif MA0052. For both subfigures, we do not plot results located in the left lower corner for reasons of clarity (Figure 11.3).

Differentially abundant vs. over-represented motifs

Third, we consider the aspect of distinguishing between over-represented and differentially abundant motifs in the data set. Typically, promoters contain BSs of many different TFs. When applying de-novo motif discovery tools to such sequences, not all of these motifs are equally relevant. For instance, when comparing promoters of differentially and non-differentially expressed genes for a specific condition, we are typically interested in those motifs that differentially abundant in these sets of promoters and not in those motifs that are common to the promoters of both types of genes. Hence, it is beneficial for a de-novo motif discovery tool to distinguish between over-represented and relevant motifs.

Here, we consider the target data set containing BSs of MA0048 with a Gaussian distribution. We compare the results for a data set with a uniformly implanted decoy motif (MA0052) to the same data set without implanted decoy motif. In Figure 11.5, we show the comparison of the nucleotide precision recall curves for known motif length. In case of no decoy motif, we observe that A-GLAM, DEME, DME, Improbizer, MEME, Weeder, and Dispom are capable of finding the correct motif. In a comparison, Dispom performs best, A-GLAM, DEME, DME, and Dispom perform best, Improbizer and MEME perform second best, and Weeder performs third best of these tools.

Considering the data set containing a decoy motif, we observe that A-GLAM, Improbizer, MEME, and Weeder, which are not designed for finding motifs that are differentially abundant in two data sets, are not capable of finding the correct motif. Characteristically, Improbizer, MEME, and Weeder find the unspecific decoy motif (Additional File 4 of article [Keilwagen et al., 2010a]). In contrast, DEME, DME, and Dispom, which are specially designed for finding differentially abundant BSs, are capable of finding the correct motif.

Based on these case studies, we can state that discriminative de-novo motif discovery tools are capable of distinguishing between over-represented and differentially abundant motifs. This property is useful if we like to find motifs that help to discriminate between two data sets. The discriminative de-novo motif discovery tools DEME, DME, and Dispom are capable of finding the correct motif irrespective of the absence or presence of a decoy motif. Hence, they perform very similar in both cases.

Comprehensive comparison

After investigating three aspects of de-novo motif discovery in detail, we now compare all eight tools based on several data sets. To summarize this comparison, we show the nucleotide precision achieved for a nucleotide recall of 10%, 30%, 50%, 70%, and 90%. Based on the partial nucleotide precision recall curves for some tools, we may obtain missing values for some nucleotide recalls of some tools and some data sets. In Figure 11.6, we consider the Gaussian data sets and unknown motif length. Complete and partial nucleotide precision recall curves as well as summaries similar to Figure 11.6 can be found in the Additional File 2-5 of article [Keilwagen et al., 2010a].

For an initial assessment, we first determine for each tool the number of data sets where not exclusively missing values are observed. We find that DME and Gibbs Sampler are unsuccessful in all data sets, while MEME is successful in two data sets, A-GLAM, DEME, and Improbizer in four data sets, Weeder in six data sets, and Dispom in all nine data sets. This initial assessment might be unfair for some tools, since it does not take into account the achieved values of the nucleotide precision. For example, A-GLAM and Improbizer often achieve very high nucleotide precisions, which is not considered in the initial assessment. Hence, we perform a second assessment in which we require a minimum nucleotide precision of 75%. We find that DEME, DME, Gibbs Sampler, and Weeder are unsuccessful in all data sets, while MEME is successful in one data set, Improbizer in two data sets, A-GLAM in three data sets, and Dispom in all nine data sets.

Considering the plant data sets, MA0001, MA0005, and MA0054, we find that most of the tools fail to find the correct motif, while Dispom finds the motif in all three cases. Considering the results for the other data sets and for known motif length (Additional File 2-5 of article [Keilwagen et al., 2010a]), we find similar results for unknown motif length on the uniform data sets and slightly better results for known motif length on both data sets. This indicates that the knowledge of the motif length has a decisive influence on the performance of many of the studied de-novo motif discovery tools. Especially DME, which performs poor in this case study (Figure 11.6), improves if the correct motif length is provided (Additional File 4 of article [Keilwagen et al., 2010a]). Interestingly, Dispom is capable of adapting the motif length from the data, and so it outperforms the other tools.

Dispom utilizes the discriminative MSP principle for learning its parameters. This learning principle is linked to the classification of promoter sequence as belonging to the target or the control data set. Nevertheless, Dispom is capable of finding the motif of interest. Recently,

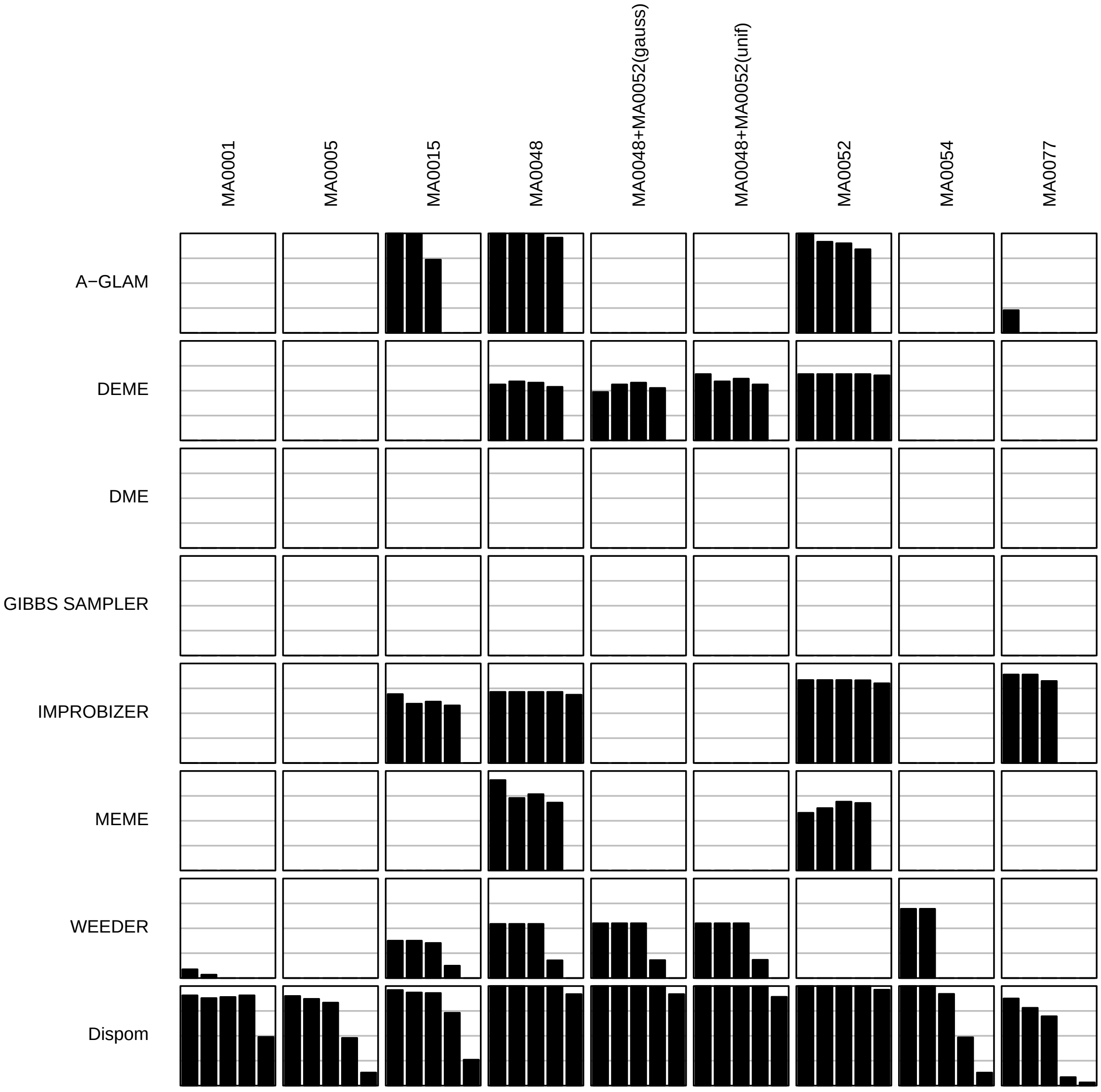


Figure 11.6: Overview of de-novo motif discovery results for Gaussian data sets and unknown motif length. Each column shows the results of one data set, and each row shows the results of one de-novo motif discovery tool. Each subfigure shows five bars that visualize the nucleotide precision for a nucleotide recall of 10%, 30%, 50%, 70%, and 90%, respectively, from left to right. Additionally, each subfigure contains gray horizontal lines for the nucleotide precision of 25%, 50%, and 75%.

KIRMES [Schultheiss et al., 2009] has been proposed, which also aims at an accurate classification of the data sets but is based on a support vector machine, which makes the interpretation of the found motifs a little bit harder. Furthermore, KIRMES utilizes besides a set of promising k-mers also the results of de-novo motif discovery tools or known motifs from databases like TRANSFAC [Matys et al., 2006] or JASPAR [Bryne et al., 2008]. Hence, it is hard to assign KIRMES to exactly one of the following categories: de-novo motif discovery tools, prediction tools like MATCH [Kel et al., 2003], and ensemble methods like MotifVoter [Wijaya et al., 2008] that integrates the output of different de-novo motif discovery tools to obtain a better prediction. However, Dispom with its good performance can be integrated in KIRMES or other ensemble methods for improving their predictions.

11.2.2 Finding two motifs simultaneously

Next, we test whether Dispom is capable of finding two different motifs in a data set simultaneously. For this reason, we use each of the three benchmark data sets containing BSs of MA0048 and MA0052 as target data set and a set of randomly chosen promoters without any implanted BSs as control data set. We start Dispom with the same options as before except the number of motifs, which we increase to two. Specifically, we allow Dispom to learn the length of the motif from that data similar to an application on experimental data. In Figure 11.7, we illustrate the results of Dispom for the three target data sets using the same control data set.

We find for all three cases that Dispom is capable of finding both motifs including their position distributions. Comparing the sequence logos obtained for a p-value of 1.0×10^{-4}, we find that Dispom is capable of finding the correct motif in three out of six cases, and that Dispom finds the core motif in the remaining three cases. Regarding the motif length, we find that Dispom learns the correct motif length. For the three cases where Dispom only finds the core motif the motif has the correct length but is shifted by one position. Comparing the position distribution learned by Dispom with the histogram of the real start positions, we find that Dispom is capable of learning the position distribution with high accuracy. Hence, when combining both – the motif and the position distribution – we find an area under the nucleotide precision recall curve of more than 0.85 in case of finding the motif, and an area under the nucleotide precision recall curve of more than 0.60 in case of finding the core motif.

These results show that Dispom is capable of learning two motifs simultaneously if each sequence contains at most one BS indicating that the extended zero or one occurrence per sequence model is a beneficial extension of the traditional ZOOPS model. However, the extended zero or one occurrence per sequence model does not model multiple BSs of one or multiple types of motifs in one sequence. Yet, it allows to address an interesting subclass of biological problems.

11.2.3 Applying Dispom to promoters of auxin-responsive genes

In the last but one subsection, we compared the performance of Dispom and seven commonly used tools based on 18 data sets, suggesting that Dispom might be useful for finding differentially abundant BSs and their positional preference. In this subsection, we apply Dispom to promoters of auxin-responsive

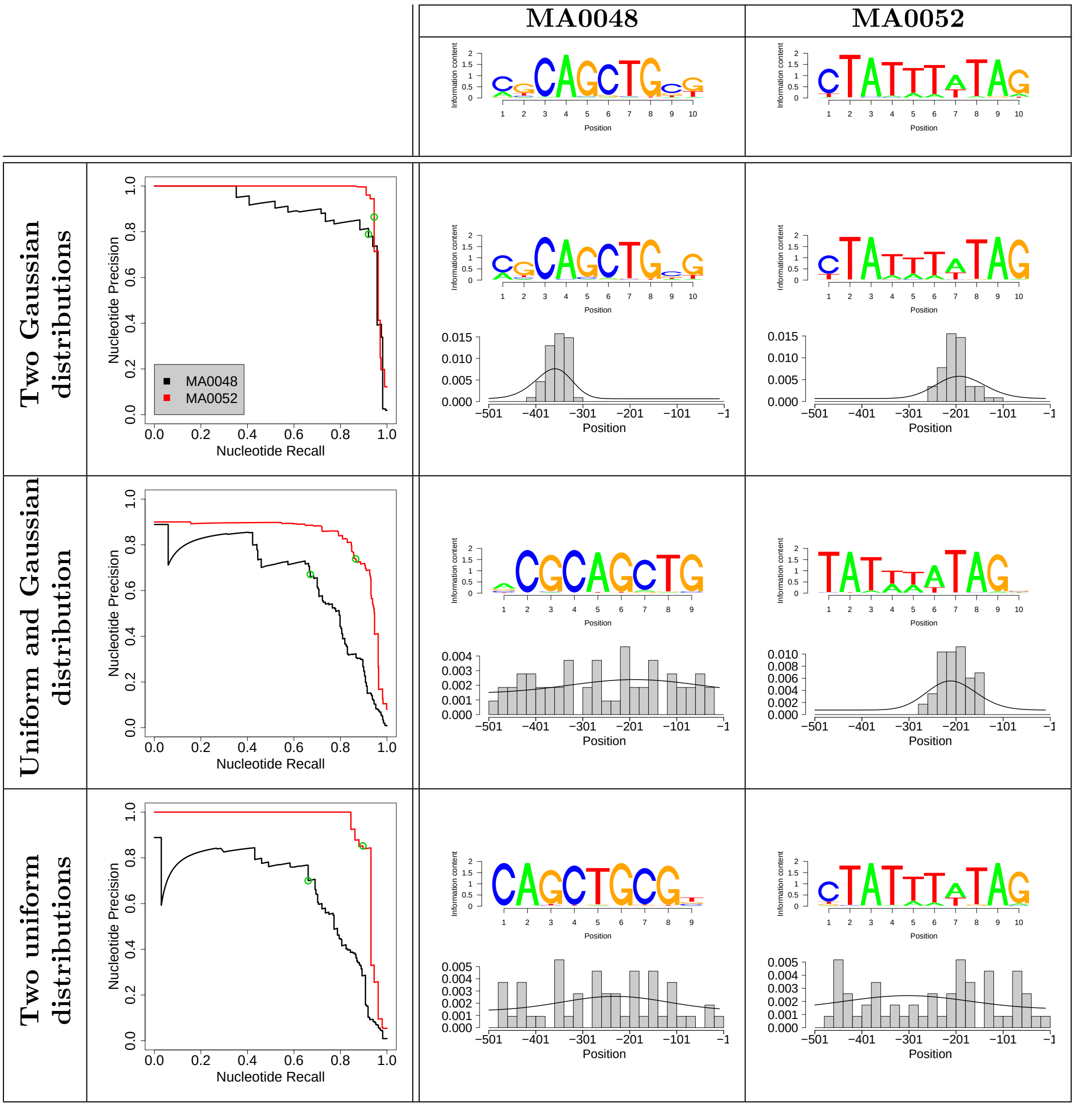
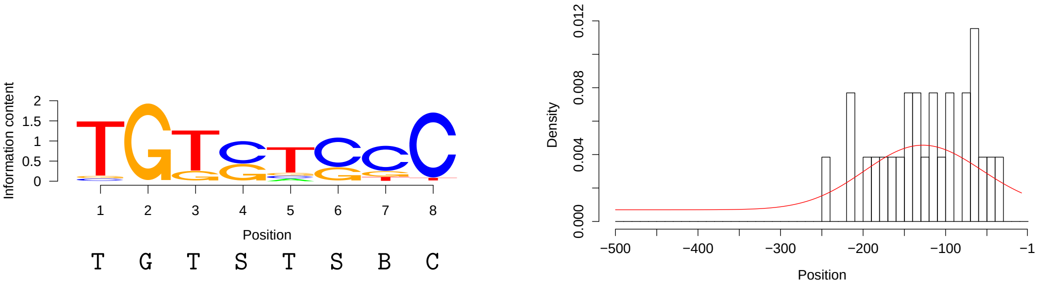


Figure 11.7: Comparison of the performance of Dispom for two motifs and for uniform and Gaussian position distribution. The first row show the sequence logos of the two motifs, MA0048 and MA0052, implanted into the data set. Row two to four show the results for the three target data sets, where row two is based on the target data set using two Gaussian distributions, row three is based on the target data set using a uniform and a Gaussian distribution, and row four is based on the target data set using the uniform distribution in both cases. For these three rows, the first column contains the nucleotide precision recall curves, where the black and the red line show the curves for motif MA0048 and MA0052, respectively. The second and the third column show the sequence logo found by Dispom and the position distribution of the BS. Similar to Figure 11.3, 11.4, and 11.5, the histogram show the distribution of the real start positions and the line visualizes the distribution learned by Dispom. The green circles in the nucleotide precision recall curve indicate a p-value of 1.0×10^{-4} that is used to generate the sequence logos depicted in column two and three.



(a) Sequence logo and consensus sequence

(b) Position distribution

Figure 11.8: Auxin-dependent motif and position distribution found by Dispom. Figure a) shows the sequence logo obtained from the predictions of Dispom and the corresponding consensus sequence, where `S` stands for `C` or `G`, and `B` stands for `C, G,` or `T`. Figure b) shows a histogram of the predicted start positions and the position distribution learned by Dispom (red line).

genes with the goal of finding putative TFBSs.

Auxin-responsive genes are regulated by a set of TFs commonly called auxin-responsive factors (ARF), which bind to auxin-responsive elements (AuxREs) that occur in the promoters of those genes. The canonical AuxRE `TGTCTC` has been identified as a sequence specifically bound by ARF1 using gel mobility shift assays [Ulmasov et al., 1997]. However, the ARF multi-gene family consists of 23 members [Guilfoyle and Hagen, 2007], suggesting that AuxREs might differ for different members of ARFs. Indeed, subsequent analyses of 10 members of the ARF family indicate that only the first four nucleotides `TGTC` are essential for ARF-binding [Ulmasov et al., 1999]. However, these 10 members do not cover all aspects of transcriptional gene regulation by auxin, so the AuxRE is still under discussion.

Analyses of genome-wide expression data are based on the assumption that co-expressed genes are regulated by the same TFs, and so contain the same TFBSs in their promoters. We use expression data sets for searching for a refined AuxRE. We apply Dispom to a set of promoters of genes up-regulated by the plant hormone auxin in *Arabidopsis thaliana* grown in a cell suspension culture [Paponov et al., 2008]. Figure 11.8 visualizes the results of Dispom as a sequence logo [Schneider and Stephens, 1990] and the positional preference corresponding to this motif. We find a motif of length 8 bp predominately positioned in the 250-bp region upstream of the TSS. The core motif can be described as `TGTSTSBC` and can be interpreted as an elongated and modified version of the canonical AuxRE `TGTCTC`.

The presence of the canonical AuxRE `TGTCTC` in the promoters of a gene is often used as an indicator that this gene is auxin-responsive. For avoiding parameter over-fitting, we use an independent test data sets for evaluating the discriminative power of the found consensus sequences. We use the seedling data set described in the section Methods as target test data set, and we use the promoters of all remaining genes that are spotted on the chip as control test data set.

We analyze the discriminative power of the defined consensus sequences for the region [-500,-1].

	[-500,-1]	**[-250,-1]**
`TGTCTC`	1.5×10^{-2}	1.0×10^{-3}
`TGTSTSBC`	2.0×10^{-4}	3.5×10^{-6}

Table 11.2: Rows indicate motif descriptions, while columns indicate the promoter region used for searching the BSs. Each cell contains the p-value obtained from Fisher's exact test using the confusion matrix for the consensus and promoter region specified by row and column, respectively, on the seedling data set and all remaining genes.

For the canonical AuxRE `TGTCTC`, we find that 36 out of 113 promoters in the target test set (32%) contain this motif, whereas only 4741 out of 21012 promoters in the control test set (23%) contain this motif. This increase of enrichment from 23% to 32% is statistically significant, yielding a p-value of 1.5×10^{-2} by Fisher's exact test, stating that the canonical `TGTCTC` motif is significantly enriched in the cell suspension data set compared to the randomly chosen promoters in the control data set. Interestingly, the restriction to the first four nucleotides `TGTC`, considered by some authors to be an improvement over the canonical ARF motif [Ulmasov et al., 1999], decreases rather than increases the specificity (Additional File 6 of article [Keilwagen et al., 2010a]). However, when using the extended motif `TGTSTSBC`, we find that 26 out of 113 promoters in the target test set (23%) contain this motif, whereas only 2305 out of 21012 promoters in the control test set (11%) contain this motif. The difference of these percentages is statistically significant, yielding a p-value of 2.0×10^{-4}. Comparing both p-values, we find a more than 70-fold decrease of the p-value when replacing the canonical ARF motif `TGTCTC` by the extended motif `TGTSTSBC`, stating that the extended motif found by Dispom is significantly more auxin specific than the canonical ARF motif.

Using the positional preference identified by Dispom, we repeat the analysis of the promoters for the interval from -250 to -1 bp with respect to the TSS (Additional File 6 of article [Keilwagen et al., 2010a]). For the canonical AuxRE `TGTCTC`, we find that 26 out of 113 promoters in the target test set (23%) contain this motif, whereas only 2564 out of 21012 promoters in the control test set (12%) contain it, yielding a p-value of 1.0×10^{-3}. Comparing this p-value with the p-value obtained for the region $[-500, -1]$, we find a more than 10-fold decrease, indicating that the positional preference found by Dispom is of high relevance alone. Considering the motif `TGTSTSBC`, we find that 21 out of 113 promoters in the target test set (19%) contain this motif, whereas only 1252 out of 21012 promoters in the control test set (6%) contain it, leading to a p-value of 3.5×10^{-6}. Comparing this p-value with the p-value obtained for the region $[-500, -1]$, we find the same motif is approximately 60-fold more auxin specific in the region $[-250, -1]$. We find that combining the refined motif `TGTSTSBC` and the refined upstream region $[-250, -1]$ yields the lowest p-value of 3.5×10^{-6}, which is more than 4,000-fold lower than the p-value obtained for the traditional combination of the canonical AuxRE `TGTCTC` in the 500-bp upstream region. This observation illustrates the power of combining a discriminative motif finding approach with the approach of simultaneously learning the positional distribution from the data.

In Table 11.2, we summarize these p-values for the canonical AuxRE motif and the `TGTSTSBC` motif for the 500-bp upstream regions and the 250-bp upstream regions. Interestingly, restricting the promoter region to -250 to -1 decreases the p-value strongly, so the capability of Dispom of learning the positional distribution turns out to be essential for finding an auxin-dependent motif.

11.3 Conclusions

Gene regulation and specifically the binding of TFs to their BSs is of fundamental interest in many areas of genome biology. A combination of experimental and computational methods are typically used for finding putative TFBSs. For computational approaches, two fundamental improvements have been proposed in the last years. On the one hand searching for differentially abundant motifs, and on the other hand learning a position distribution have been shown to be promising in several experiments separately. However, up to now there is no tool combining both improvements.

We present Dispom a new computational tool for the de-novo motif discovery that combines the capability of searching for differentially abundant BSs with the capability of learning the BSs. Dispom includes a heuristic for finding motifs of unknown length. We compare the performance of Dispom with seven commonly used de-novo motif discovery tools based on 18 data sets, and we find that Dispom outperforms these tools. Especially in cases where the correct motif length is not provided, the predictions of Dispom are substantially more accurate than those of traditional de-novo discovery tools indicating that the combination of discriminative learning, inferring a position distribution from the data, and utilizing a heuristic for finding the motif length is beneficial for de-novo motif discovery.

In addition, we test whether Dispom is capable of finding two motifs simultaneously, and we find that Dispom still is able to discover the motifs with high accuracy. This indicates that the extension of the ZOOPS model to at most one BS of more than one motif type is feasible allowing to concern a wider range of biological problems than the traditional ZOOPS model.

Finally, we use Dispom on a set of auxin-responsive genes where the true motif is unknown. We find the motif `TGTSTSBC`, which can be interpreted as the elongated AuxRE, predominantly located in the promoter region of -250 to -1. Both the elongated motif as well as the refined promoter region lead to a more than 4,000-fold improvement of the significance of predicting of auxin-responsive genes on genome scale in an independent test data set. Interestingly, the refined promoter region seems to be of high importance for the prediction.

These findings suggest that Dispom might be beneficial for finding differentially abundant BSs and their positional distribution based on high-throughput data. Hence, we make Dispom available for the scientific community as part of the open-source Java library Jstacs [Keilwagen et al., 2008] at `http://www.jstacs.de/index.php/Dispom`.

Chapter 12

Conclusions and Outlook

New experimental techniques including several high-throughput methods have heralded the “Age of Biology.” In this age, bioinformatic analyses of experimental data have become indispensable in many areas of life science. Specifically, the analysis of genetic information, which is encoded in the DNA, is often of fundamental interest, since it supports or confutes different hypotheses and assists in the experimental design.

During the last years several computational methods have been proposed for analyzing experimental data, and it has often been pointed out that discriminative approaches, as for instance support vector machines, outperform probabilistic models in many cases. However, the performance of probabilistic models is highly dependent on the model parameters that are obtained from training data using a specific learning principle. For historical reasons, generative learning principles that aim at an accurate representation of the data in each class are widely used in bioinformatics. In contrast to these generative learning principles, discriminative learning principles that aim at an accurate classification of the data with respect to the class labels have been proposed in the machine-learning community in the last decade. In addition, hybrid learning principles have been demonstrated to benefit from the advantages of generative and discriminative learning principles. Besides the characterization of learning principles with respect to their objective, learning principles can be characterized by the utilization of a-priori knowledge, which often influences learning parameters positively. Bayesian learning principles utilize a-priori knowledge aside from training data to infer the model parameters, whereas non-Bayesian learning principles solely use training data. However, several different learning principles and models have been proposed independently, and there is no common basis that allows to examine their strengths or weaknesses.

In this work, we propose a unified generative-discriminative learning principle that incorporates six established learning principles including generative, discriminative, and hybrid, as well as Bayesian and non-Bayesian learning principles as special cases. In addition, we propose a generalization of the product-Dirichlet prior that can be used for the broad family of Markov random fields allowing to include biological a-priori knowledge more easily. This prior can be used for each of the presented learning principles and provides a common base for their unbiased comparison. We implement these theoretical findings in the open-source Java library Jstacs, which allows a modular combination of different models and learning principles.

Applying both – the proposed learning principle and the proposed prior – together using Jstacs, we show for several data sets that the performance of probabilistic models can be improved. We also find that the choice of an appropriate learning principle is often as important as the choice of an appropriate model. However, due to the discriminative part of the learning principle, it is computational demanding to determine the optimal weights $\underline{\beta}$ for the unified generative-discriminative learning principle. Hence, if we are only interested in classifying sequences, we can confirm the general observation that discriminative learning principles often outperform their generative counterparts.

For the recognition of donor splice sites and of TSSs, we find that the presented tools – using discriminatively trained probabilistic models – perform as good as, or even better than, several state-of-the-art approaches. In both applications, we find that modeling the upstream and the downstream region helps to improve the performance, when using discriminative learning principle. Analyzing TFBS annotations in the gene-regulatory database CoryneRegNet, we find that a simple probabilistic model can help to detect errors that occur during the transfer from the scientific literature into this database. Scrutinizing the predictions, we find that they are in accordance with the scientific literature. Turning to the prediction of putative TFBSs from target promoters, we find that the combination of discriminative learning and learning a position distribution for the start position of the putative TFBSs improves the performance of de-novo motif discovery tools. Applying the corresponding tool to auxin-responsive genes, we find a putative auxin responsive element that is three orders of magnitude more specific than the traditional auxin responsive element on independent test data.

These biological applications show that the presented framework allows to design tools that are competitive with state-of-the-art approaches. The developed tools are modular combinations of different models and learning principles implemented in Jstacs allowing to exploit the computing power of modern multi-core computers. These applications indicate that Jstacs is useful for assessing the benefits of different models or learning principles, and at the same time allows to build sophisticated tools for specific biological applications. For this reason, Jstacs can serve as a common basis for integrating and investigating further biologically relevant features as for instance incorporating phylogeny or handling continuous data.

Glossary

auc	Area under the curve	18
	Area under the PR curve (aucPR)	18
	Area under the ROC curve (aucROC)	18
BS	Binding site	5, 6
cMM	Cyclic Markov model	21, 49
DAG	Directed acyclic graph	24
ESS	Equivalent sample size	37
fpr	False positive rate	18
GTPD	Generalized transformed product-Dirichlet	41
GDT	Generative-discriminative trade-off	11–14
hMM	Homogeneous Markov model	20, 21, 49
IPM	Independent product model	32, 33, 49
iMM	Inhomogeneous Markov model	20, 23, 24
MRF	Markov random field	22, 23, 27, 28, 49
MAP	Maximum a posteriori	11–17
MCL	Maximum conditional likelihood	11–14
ML	Maximum likelihood	10–14, 17
MSP	Maximum supervised posterior	11–16
mBN	Moral Bayesian network	22, 23, 27, 49

nP	Nucleotide precision	95, 96
nR	Nucleotide recall	95, 96
PGDT	Penalized generative-discriminative trade-off	11, 13–15
PWM	Position weight matrix	20, 22–24
ppv	Positive predictive value	18
PR	Precision recall	18
	Nucleotide precision recall (nPR)	96
PPP	Promoter prediction program	72
ROC	Receiver operating characteristic	18
Sn	Sensitivity	18
Sp	Specificity	18
TF	Transcription factor	6
TFBS	Transcription factor binding site	5–7
TSS	Transcription start site	5–7
WAM	Weight array matrix	20, 23, 24
ZOOPS	Zero or one occurrence per sequence	31, 32
	Extended zero or one occurrence per sequence (eZOOPS)	31, 49

List of Figures

3.1 Visualization of learning principles 14
3.2 Illustration of the unified generative-discriminative learning principle 16

4.1 Relation of Bayesian networks and Markov random fields. 23

5.1 Illustration of the GTPD prior for one and two free parameters. 42

6.1 Simplified Jstacs class hierarchy. 48

7.1 Comparison of classification performance for donor and acceptor splice sites based on the ML and the MCL principle. 52
7.2 Comparison of classification performance for donor splice sites based on MAP and MSP principle. 54
7.3 Comparison of classification performance for different sizes of the training data sets using the MAP and the MSP principle. 55
7.4 Mean performance for four TFBSs data sets using the unified generative-discriminative learning principle 58
7.5 Computational costs for different learning principles 60
7.6 The unified generative-discriminative learning principle for different sizes of the training data sets. 62
7.7 Comparison of the MAP and the MSP principle for different ESSs and different sizes of the training data sets. 64

8.1 Foreground model for donor splice sites. 69

9.1 Probabilistic models used in PACT 75
9.2 Comparison of precision recall curves for promoter prediction programs. 77
9.3 Comparison of protocol $2A$ and $2A'$ using PACT 79

10.1 Comparison of binding site conservation for seven transcription factor data sets 87
10.2 Position of the predicted NarL binding site in the upstream region of *torC* 90

11.1 Flow diagram for Dispom. 97
11.2 P-value distribution for the length of common subsequences in promoter regions of *Arabidopsis thaliana* 99
11.3 Comparison of nucleotide precision recall curves of different de-novo motif discovery tools for known and unknown motif length 102
11.4 Comparison of nucleotide precision recall curves of different de-novo motif discovery tools for uniform and Gaussian position distribution 103
11.5 Comparison of nucleotide precision recall curves of different de-novo motif discovery tools for a data set with and without decoy motif 104
11.6 Overview of de-novo motif discovery results for Gaussian data sets and unknown motif length 106
11.7 Comparison of the performance of Dispom for two motifs, and for uniform and Gaussian position distribution 108
11.8 Auxin-dependent motif and position distribution found by Dispom 109

List of Tables

3.1 Characterisation of learning principles 11
3.2 Confusion matrix for two classes. 18

5.1 Established priors for moral Bayesian networks and Markov random fields 36

6.1 Main classes of Jstacs used in this work. 49

7.1 Size of splice site data sets of [Yeo and Burge, 2004]. 51
7.2 Summary of the results of Figure 7.4. 59
7.3 Results of applying the unified generative-discriminative learning principle on donor splice sites 66

8.1 Classification performance for donor splice sites of *Caenorhabditis elegans* 70
8.2 Comparison of area under the curve for donor splice sites of *Caenorhabditis elegans* . . 71

9.1 Statistics of splits for human CAGE data 76
9.2 Performance of different promoter prediction programs. 78

10.1 MotifAdjuster reannotation results 86
10.2 Reannotation statistic for NarL: Binding site shifts and strand switches 88
10.3 NarL binding sites with questionable annotations 89

11.1 Characterization of de-novo motif discovery tools. 93
11.2 Significance for two auxin-dependent motif descriptions 110

Bibliography

[Abeel et al., 2008a] Abeel, T., Saeys, Y., Bonnet, E., Rouz, P., and de Peer, Y. V. (2008a). Generic eukaryotic core promoter prediction using structural features of dna. *Genome Res*, 18(2):310–323.

[Abeel et al., 2008b] Abeel, T., Saeys, Y., Rouze, P., and Van de Peer, Y. (2008b). ProSOM: core promoter prediction based on unsupervised clustering of DNA physical profiles. *Bioinformatics*, 24(13):i24–31.

[Abeel et al., 2009] Abeel, T., Van de Peer, Y., and Saeys, Y. (2009). Toward a gold standard for promoter prediction evaluation. *Bioinformatics*, 25(12):i313–i320.

[Aldrich, 1997] Aldrich, J. (1997). R. A. Fisher and the Making of Maximum Likelihood 1912-1922. *Statistical Science*, 12(3):162–176.

[Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, 215(3):403–410.

[Ao et al., 2004] 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.

[Azzalini, 1985] Azzalini, A. (1985). A class of distributions which includes the normal ones. *Scandinavian journal of statistics*, 12(2):171–178.

[Babu and Teichmann, 2003] Babu, M. M. and Teichmann, S. A. (2003). Evolution of transcription factors and the gene regulatory network in Escherichia coli. *Nucleic Acids Res*, 31(4):1234–1244.

[Baichoo and Helmann, 2002] Baichoo, N. and Helmann, J. D. (2002). Recognition of DNA by Fur: a reinterpretation of the Fur box consensus sequence. *J Bacteriol*, 184(21):5826–5832.

[Bailey and Eklan, 1994] Bailey, T. L. and Eklan, 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.

[Bajic et al., 2004] Bajic, V. B., Tan, S. L., Suzuki, Y., and Sugano, S. (2004). Promoter prediction analysis on the whole human genome. *Nat Biotechnol*, 22(11):1467–1473.

[Baldi et al., 2000] Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., and Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424.

[Bao et al., 2008] Bao, L., Zhou, M., and Cui, Y. (2008). CTCFBSDB: a CTCF-binding site database for characterization of vertebrate genomic insulators. *NAR*, 36(suppl_1):D83–87.

[Barash et al., 2003] Barash, Y., Elidan, G., Friedman, N., and Kaplan, T. (2003). Modeling Dependencies in Protein-DNA Binding Sites. *In proceedings of Seventh Annual International Conference on Computational Molecular Biology*, pages 28–37.

[Barrett et al., 2005] Barrett, J. C., Fry, B., Maller, J., and Daly, M. J. (2005). Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics*, 21(2):263–265.

[Baumbach et al., 2009] Baumbach, J., Wittkop, T., Kleindt, C. K., and Tauch, A. (2009). Integrated analysis and reconstruction of microbial transcriptional gene regulatory networks using CoryneRegNet. *Nat Protoc*, 4(6):992–1005.

[Ben-Gal et al., 2005] 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.

[Bennetzen and Ma, 2003] Bennetzen, J. L. and Ma, J. (2003). The genetic colinearity of rice and other cereals on the basis of genomic sequence analysis. *Curr Opin Plant Biol*, 6(2):128–133.

[Benotmane et al., 1997] Benotmane, A. M., Hoylaerts, M. F., Collen, D., and Belayew, A. (1997). Nonisotopic quantitative analysis of protein-DNA interactions at equilibrium. *Anal Biochem*, 250(2):181–185.

[Berger et al., 1996] Berger, A., Della Pietra, S., and Della Pietra, V. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

[Bernal et al., 2007] Bernal, A., Crammer, K., Hatzigeorgiou, A., and Pereira, F. (2007). Global discriminative learning for higher-accuracy computational gene prediction. *PLoS Comput Biol*, 3(3):e54.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Information Science and Statistics. Springer, New York, 1st edition.

[Bouchard, 2007] Bouchard, G. (2007). Bias-variance tradeoff in hybrid generative-discriminative models. In *ICMLA '07: Proceedings of the Sixth International Conference on Machine Learning and Applications*, pages 124–129, Washington, DC, USA. IEEE Computer Society.

[Bouchard and Triggs, 2004] Bouchard, G. and Triggs, B. (2004). The tradeoff between generative and discriminative classifiers. In *IASC International Symposium on Computational Statistics (COMPSTAT)*, pages 721–728, Prague.

[Boutilier et al., 1996] Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. (1996). Context-specific independence in Bayesian networks. *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence*, pages 115–123.

[Bryne et al., 2008] Bryne, J. C., Valen, E., Tang, M.-H. E., Marstrand, T., Winther, O., da Piedade, I., Krogh, A., Lenhard, B., and Sandelin, A. (2008). JASPAR, the open access database of transcription factor-binding profiles: new content and tools in the 2008 update. *Nucl. Acids Res.*, 36(suppl_1):D102–106.

[Bühlmann, 1997] Bühlmann, P. (1997). Model selection for variable length Markov chains and tuning the context algorithm. Technical Report 82, Statistics, ETH Zentrum, CH-8092 Zuerich, Switzerland.

[Bülow et al., 2009] Bülow, L., Engelmann, S., Schindler, M., and Hehl, R. (2009). AthaMap, integrating transcriptional and post-transcriptional data. *NAR*, 37(suppl_1):D983–986.

[Buntine, 1991] Buntine, W. L. (1991). Theory refinement of Bayesian networks. In *Uncertainty in Artificial Intelligence*, pages 52–62. Morgan Kaufmann.

[Buntine, 1994] Buntine, W. L. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225.

[Burge and Karlin, 1997] Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *J Mol Biol*, 268(1):78–94.

[Burset et al., 2000] Burset, M., Seledtsov, I. A., and Solovyev, V. V. (2000). Analysis of canonical and non-canonical splice sites in mammalian genomes. *Nucleic Acids Res*, 28(21):4364–4375.

[Cai et al., 2000] Cai, D., Delcher, A., Kao, B., and Kasif, S. (2000). Modeling splice sites with Bayes networks . *Bioinformatics*, 16(2):152–158.

[Carninci et al., 2006] Carninci, P., Sandelin, A., Lenhard, B., Katayama, S., Shimokawa, K., Ponjavic, J., Semple, C. A. M., Taylor, M. S., Engstrm, P. G., Frith, M. C., Forrest, A. R. R., Alkema, W. B., Tan, S. L., Plessy, C., Kodzius, R., Ravasi, T., Kasukawa, T., Fukuda, S., Kanamori-Katayama, M., Kitazume, Y., Kawaji, H., Kai, C., Nakamura, M., Konno, H., Nakano, K., Mottagui-Tabar, S., Arner, P., Chesi, A., Gustincich, S., Persichetti, F., Suzuki, H., Grimmond, S. M., Wells, C. A., Orlando, V., Wahlestedt, C., Liu, E. T., Harbers, M., Kawai, J., Bajic, V. B., Hume, D. A., and Hayashizaki, Y. (2006). Genome-wide analysis of mammalian promoter architecture and evolution. *Nat Genet*, 38(6):626–635.

[Carpenter et al., 2006] Carpenter, A., Jones, T., Lamprecht, M., Clarke, C., Kang, I., Friman, O., Guertin, D., Chang, J., Lindquist, R., Moffat, J., Golland, P., and Sabatini, D. (2006). CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10):R100.

[Casella and George, 1992] Casella, G. and George, E. I. (1992). Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174.

[Castelo, 2002] Castelo, R. (2002). *The discrete acyclic digraph Markov model in data mining.* PhD thesis, Faculteit Wiskunde en Informatica, Universiteit Utrecht.

[Castelo and Guigo, 2004] Castelo, R. and Guigo, R. (2004). Splice site identification by idlBNs. *Bioinformatics*, 20(1):i69–76.

[Cerquides and de Mántaras, 2005] Cerquides, J. and de Mántaras, R. L. (2005). Robust Bayesian linear classifier ensembles. In *ECML*, pages 72–83.

[Chen and Rosenfeld, 1999] Chen, S. and Rosenfeld, R. (1999). A gaussion prior for smoothing maximum entropy models. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

[Cochrane and Galperin, 2010] Cochrane, G. R. and Galperin, M. Y. (2010). The 2010 Nucleic Acids Research Database Issue and online Database Collection: a community of data resources. *Nucleic Acids Res*, 38(Database issue):D1–D4.

[Constantinidou et al., 2006] Constantinidou, C., Hobman, J. L., Griffiths, L., Patel, M. D., Penn, C. W., Cole, J. A., and Overton, T. W. (2006). A reassessment of the FNR regulon and transcriptomic analysis of the effects of nitrate, nitrite, NarXL, and NarQP as Escherichia coli K12 adapts from aerobic to anaerobic growth. *J Biol Chem*, 281(8):4802–4815.

[Cooper and Herskovits, 1992] Cooper, G. F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

[Crooks et al., 2004] Crooks, G. E., Hon, G., Chandonia, J.-M., and Brenner, S. E. (2004). WebLogo: a sequence logo generator. *Genome Res*, 14(6):1188–1190.

[Cui et al., 1995] Cui, Y., Wang, Q., Stormo, G., and Calvo, J. (1995). A consensus sequence for binding of Lrp to DNA. *J. Bacteriol.*, 177(17):4872–4880.

[Culotta et al., 2005] Culotta, A., Kulp, D., and McCallum, A. (2005). Gene prediction with conditional random fields. Technical Report Technical Report UM-CS-2005-028, University of Massachusetts, Amherst.

[Darwin et al., 1996] Darwin, A. J., Li, J., and Stewart, V. (1996). Analysis of nitrate regulatory protein NarL-binding sites in the fdnG and narG operon control regions of Escherichia coli K-12. *Mol Microbiol*, 20(3):621–632.

[Darwin et al., 1997] Darwin, A. J., Tyson, K. L., Busby, S. J., and Stewart, V. (1997). Differential regulation by the homologous response regulators NarL and NarP of Escherichia coli K-12 depends on DNA binding site arrangement. *Mol Microbiol*, 25(3):583–595.

[Davis and Goadrich, 2006] 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.

[Davuluri et al., 2001] Davuluri, R. V., Grosse, I., and Zhang, M. Q. (2001). Computational identification of promoters and first exons in the human genome. *Nature Genetics*.

[De Bona et al., 2008] De Bona, F., Ossowski, S., Schneeberger, K., and Rätsch, G. (2008). Optimal spliced alignments of short sequence reads. *Bioinformatics*, 24(16):i174–i180.

[De Wulf et al., 2002] De Wulf, P., McGuire, A. M., Liu, X., and Lin, E. C. C. (2002). Genome-wide profiling of promoter recognition by the two-component response regulator CpxR-P in Escherichia coli. *J Biol Chem*, 277(29):26652–26661.

[Degroeve et al., 2005] Degroeve, S., Saeys, Y., Baets, B. D., Rouz, P., and de Peer, Y. V. (2005). Splicemachine: predicting splice sites from high-dimensional local context representations. *Bioinformatics*, 21(8):1332–1338.

[Down and Hubbard, 2002] Down, T. A. and Hubbard, T. J. P. (2002). Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Res*, 12(3):458–461.

[Eichler et al., 1996] Eichler, K., Buchet, A., Lemke, R., Kleber, H. P., and Mandrand-Berthelot, M. A. (1996). Identification and characterization of the caiF gene encoding a potential transcriptional activator of carnitine metabolism in escherichia coli. *J Bacteriol*, 178(5):1248–1257.

[Elemento et al., 2007] Elemento, O., Slonim, N., and Tavazoie, S. (2007). A universal framework for regulatory element discovery across all genomes and data types. *Molecular Cell*, 28(2):337–350.

[Ellrott et al., 2002] Ellrott, K., Yang, C., Sladek, F. M., and Jiang, T. (2002). Identifying transcription factor binding sites through Markov chain optimization. *In Proceedings of the European Conference on Computational Biology (ECCB 2002)*, pages 100–109.

[Fairbrother et al., 2002] Fairbrother, W. G., Yeh, R.-F., Sharp, P. A., and Burge, C. B. (2002). Predictive identification of exonic splicing enhancers in human genes. *Science*, 297(5583):1007–1013.

[Favorov et al., 2005] Favorov, A. V., Gelfand, M. S., Gerasimova, A. V., Ravcheev, D. A., Mironov, A. A., and Makeev, V. J. (2005). A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length. *Bioinformatics*, 21(10):2240–2245.

[Fawcett, 2004] Fawcett, T. (2004). ROC graphs: Notes and practical considerations for researchers. Technical report, HP Laboratories.

[Feelders and Ivanovs, 2006] Feelders, A. and Ivanovs, J. (2006). Discriminative scoring of Bayesian network classifiers: a comparative study. In *Proceedings of the third European workshop on probabilistic graphical models*, pages 75–82.

[Fickett, 1996] Fickett, J. W. (1996). Finding genes by computer: the state of the art. *Trends Genet*, 12(8):316–320.

[Field et al., 2008] 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.

[Fisher, 1922] Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222:309–368.

[Florea et al., 1998] Florea, L., Hartzell, G., Zhang, Z., Rubin, G. M., and Miller, W. (1998). A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res*, 8(9):967–974.

[Foissac and Schiex, 2005] Foissac, S. and Schiex, T. (2005). Integrating alternative splicing detection into gene prediction. *BMC Bioinformatics*, 6(1):25.

[Frohman et al., 1988] Frohman, M. A., Dush, M. K., and Martin, G. R. (1988). Rapid production of full-length cdnas from rare transcripts: amplification using a single gene-specific oligonucleotide primer. *Proc Natl Acad Sci U S A*, 85(23):8998–9002.

[Galas and Schmitz, 1978] Galas, D. J. and Schmitz, A. (1978). DNAse footprinting: a simple method for the detection of protein-DNA binding specificity. *Nucleic Acids Res*, 5(9):3157–3170.

[Gama-Castro et al., 2008] Gama-Castro, S., Jimnez-Jacinto, V., Peralta-Gil, M., Santos-Zavaleta, A., Pealoza-Spinola, M. I., Contreras-Moreira, B., Segura-Salazar, J., Muiz-Rascado, L., Martnez-Flores, I., Salgado, H., Bonavides-Martnez, C., Abreu-Goodger, C., Rodrguez-Penagos, C., Miranda-Ros, J., Morett, E., Merino, E., Huerta, A. M., Trevio-Quintanilla, L., and Collado-Vides, J. (2008). RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic Acids Res*, 36(Database issue):D120–D124.

[Gelfand et al., 1996] Gelfand, M. S., Mironov, A. A., and Pevzner, P. A. (1996). Gene recognition via spliced sequence alignment. *Proc Natl Acad Sci U S A*, 93(17):9061–9066.

[Geman and Geman, 1984] Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

[Golby et al., 1998] Golby, P., Kelly, D. J., Guest, J. R., and Andrews, S. C. (1998). Transcriptional regulation and organization of the dcuA and dcuB genes, encoding homologous anaerobic C4-dicarboxylate transporters in Escherichia coli. *J Bacteriol*, 180(24):6586–6596.

[Goodman, 2003] Goodman, J. (2003). Exponential priors for maximum entropy models. In *Proceedings of HLTNAACL 2004*.

[Grau et al., 2007] Grau, J., Keilwagen, J., Kel, A., Grosse, I., and Posch, S. (2007). Supervised posteriors for DNA-motif classification. In Falter, C., Schliep, A., Selbig, J., Vingron, M., and Walter, D., editors, *German Conference on Bioinformatics*, volume 115 of *Lecture Notes in Informatics (LNI) - Proceedings*, pages 123–134, Bonn. Gesellschaft für Informatik (GI).

[Greiner et al., 2005] 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.

[Grossman and Domingos, 2004] Grossman, D. and Domingos, P. (2004). Learning Bayesian network classifiers by maximizing conditional likelihood. In *ICML*, pages 361–368. ACM Press.

[Grünwald et al., 2002] 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.

[Guilfoyle and Hagen, 2007] Guilfoyle, T. J. and Hagen, G. (2007). Auxin response factors. *Curr Opin Plant Biol*, 10(5):453–460.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning : data mining, inference, and prediction.* Springer.

[Hatzigeorgiou, 2002] Hatzigeorgiou, A. G. (2002). Translation initiation start prediction in human cDNAs with high accuracy. *Bioinformatics*, 18(2):343–350.

[Heckerman et al., 1995] Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. Technical report, Microsoft Research, Advanced Technology Division, Redmond, WA 98052.

[Hellman and Fried, 2007] Hellman, L. M. and Fried, M. G. (2007). Electrophoretic mobility shift assay (EMSA) for detecting protein-nucleic acid interactions. *Nat Protoc*, 2(8):1849–1861.

[Hofacker, 2003] Hofacker, I. L. (2003). Vienna RNA secondary structure server. *Nucleic Acids Res*, 31(13):3429–3431.

[Holland et al., 2008] Holland, R. C. G., Down, T. A., Pocock, M., Prlic, A., Huen, D., James, K., Foisy, S., Drager, A., Yates, A., Heuer, M., and Schreiber, M. J. (2008). BioJava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–2097.

[Hughes et al., 2000] Hughes, J. D., Estep, P. W., Tavazoie, S., and Church, G. M. (2000). Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae. *J Mol Biol*, 296(5):1205–1214.

[Iuchi and Lin, 1987] Iuchi, S. and Lin, E. C. (1987). The narL gene product activates the nitrate reductase operon and represses the fumarate reductase and trimethylamine N-oxide reductase operons in Escherichia coli. *Proc Natl Acad Sci U S A*, 84(11):3901–3905.

[Jiang et al., 2007] Jiang, C., Xuan, Z., Zhao, F., and Zhang, M. Q. (2007). TRED: a transcriptional regulatory element database, new entries and other development. *NAR*, 35(suppl_1):D137–140.

[Johnson et al., 2007] 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.

[Jordan, 2004] Jordan, M. I. (2004). Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, 19:140–155.

[Kaiser and Sawers, 1995] Kaiser, M. and Sawers, G. (1995). Nitrate repression of the Escherichia coli pfl operon is mediated by the dual sensors NarQ and NarX and the dual regulators NarL and NarP. *J Bacteriol*, 177(13):3647–3655.

[Keilwagen et al., 2009] Keilwagen, J., Baumbach, J., Kohl, T. A., and Grosse, I. (2009). MotifAdjuster: a tool for computational reassessment of transcription factor binding site annotations. *Genome Biol*, 10(5):R46.

[Keilwagen et al., 2008] Keilwagen, J., Grau, J., Gohr, A., Posch, S., and Grosse, I. (2008). A Java framework for statistical analysis and classification of biological sequences. http://www.jstacs.de/.

[Keilwagen et al., 2010a] 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. *PLoS Computational Biology*, page submitted.

[Keilwagen et al., 2007] 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 - Abstraktion*, pages 67–72.

[Keilwagen et al., 2010b] 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 et al., 2010c] Keilwagen, J., Grau, J., Posch, S., Strickert, M., and Grosse, I. (2010c). Unifying generative and discriminative learning principles. *BMC Bioinformatics*, 11(1):98.

[Kel et al., 2003] 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 Res*, 31(13):3576–3579.

[Kent et al., 2002] Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., and Haussler, D. (2002). The human genome browser at UCSC. *Genome Res*, 12(6):996–1006.

[Kim et al., 2008] Kim, N.-K., Tharakaraman, K., Marino-Ramirez, L., and Spouge, J. L. (2008). Finding sequence motifs with Bayesian models incorporating positional information: an application to transcription factor binding sites. *BMC Bioinformatics*, 9:262+.

[Kim et al., 2007] Kim, T. H., Abdullaev, Z. K., Smith, A. D., Ching, K. A., Loukinov, D. I., Green, R. D., Zhang, M. Q., Lobanenkov, V. V., and Ren, B. (2007). Analysis of the vertebrate insulator protein CTCF-binding sites in the human genome. *Cell*, 128(6):1231–1245.

[Klein and Manning, 2003] Klein, D. and Manning, C. (2003). Maxent models, conditional estimation, and optimization. HLT-NAACL 2003 Tutorial.

[Kolchanov et al., 2002] Kolchanov, N. A., Ignatieva, E. V., Ananko, E. A., Podkolodnaya, O. A., Stepanenko, I. L., Merkulova, T. I., Pozdnyakov, M. A., Podkolodny, N. L., Naumochkin, A. N., and Romashchenko, A. G. (2002). Transcription Regulatory Regions Database (TRRD): its status in 2002. *NAR*, 30(1):312–317.

[Körner et al., 2003] Körner, H., Sofia, H. J., and Zumft, W. G. (2003). Phylogeny of the bacterial superfamily of Crp-Fnr transcription regulators: exploiting the metabolic spectrum by controlling alternative gene programs. *FEMS Microbiol Rev*, 27(5):559–592.

[Krek et al., 2005] Krek, A., Grün, 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. *Nat Genet*, 37(5):495–500.

[Kwon et al., 2005] Kwon, O., Druce-Hoffman, M., and Meganathan, R. (2005). Regulation of the ubiquinone (coenzyme q) biosynthetic genes ubica in escherichia coli. *Curr Microbiol*, 50(4):180–189.

[Lasserre et al., 2006] Lasserre, J. A., Bishop, C. M., and Minka, T. P. (2006). Principled hybrids of generative and discriminative models. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 87–94.

[Lawrence et al., 1993] 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 and Reilly, 1990] 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 et al., 2002] Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J.-B., Volkert, T. L., Fraenkel, E., Gifford, D. K., and Young, R. A. (2002). Transcriptional Regulatory Networks in Saccharomyces cerevisiae. *Science*, 298(5594):799–804.

[Leslie et al., 2002] Leslie, C., Eskin, E., and Noble, W. S. (2002). The spectrum kernel: a string kernel for svm protein classification. *Pac Symp Biocomput*, pages 564–575.

[Li et al., 1994] Li, J., Kustu, S., and Stewart, V. (1994). In vitro interaction of nitrate-responsive regulatory protein NarL with DNA target sequences in the fdnG, narG, narK and frdA operon control regions of Escherichia coli K-12. *J Mol Biol*, 241(2):150–165.

[Linhart et al., 2008] Linhart, C., Halperin, Y., and Shamir, R. (2008). Transcription factor and microRNA motif discovery: the Amadeus platform and a compendium of metazoan target sets. *Genome Res*, 18(7):1180–1189.

[Liu, 2002] Liu, J. S. (2002). *Monte Carlo Strategies in Scientific Computing.* Springer.

[MacKay, 1998] MacKay, D. J. C. (1998). Choice of basis for Laplace approximation. *Machine Learning*, 33(1):77–86.

[Maris et al., 2005] Maris, A. E., Kaczor-Grzeskowiak, M., Ma, Z., Kopka, M. L., Gunsalus, R. P., and Dickerson, R. E. (2005). Primary and secondary modes of DNA recognition by the NarL two-component response regulator. *Biochemistry*, 44(44):14538–14552.

[Matys et al., 2006] 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 and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res*, 34(Database issue):D108–D110.

[McGuffin et al., 2000] McGuffin, L. J., Bryson, K., and Jones, D. T. (2000). The PSIPRED protein structure prediction server . *Bioinformatics*, 16(4):404–405.

[Meila-Predoviciu, 1999] Meila-Predoviciu, M. (1999). *Learning with Mixtures of Trees.* PhD thesis, Massachusetts Institute of Technology.

[Méjean et al., 1994] Méjean, V., Iobbi-Nivol, C., Lepelletier, M., Giordano, G., Chippaux, M., and Pascal, M. C. (1994). Tmao anaerobic respiration in escherichia coli: involvement of the tor operon. *Mol Microbiol*, 11(6):1169–1179.

[Metz, 1978] Metz, C. E. (1978). Basic principles of ROC analysis. *Semin Nucl Med*, 8(4):283–298.

[Mönke et al., 2004] 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.

[Montgomery et al., 2006] Montgomery, S. B., Griffith, O. L., Sleumer, M. C., Bergman, C. M., Bilenky, M., Pleasance, E. D., Prychyna, Y., Zhang, X., and Jones, S. J. M. (2006). ORegAnno: an open access database and curation system for literature-derived promoters, transcription factor binding sites and regulatory variation. *Bioinformatics*, 22(5):637–640.

[Münch, 2009] Münch, R. (2009). PRODORIC URL of the PWM of NarL.

[Münch et al., 2003] Münch, R., Hiller, K., Barg, H., Heldt, D., Linz, S., Wingender, E., and Jahn, D. (2003). PRODORIC: prokaryotic database of gene regulation. *Nucleic Acids Res*, 31(1):266–269.

[Münch et al., 2005] Münch, R., Hiller, K., Grote, A., Scheer, M., Klein, J., Schobert, M., and Jahn, D. (2005). Virtual footprint and PRODORIC: an integrative framework for regulon prediction in prokaryotes. *Bioinformatics*, 21(22):4187–4189.

[Naisbitt and Aburdene, 1990] Naisbitt, J. and Aburdene, P. (1990). *Megatrends 2000: Ten new directions for the 1990's.* Morrow, New York.

[Ng and Jordan, 2002] Ng, A. Y. and Jordan, M. I. (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.

[Overton et al., 2006] Overton, T. W., Griffiths, L., Patel, M. D., Hobman, J. L., Penn, C. W., Cole, J. A., and Constantinidou, C. (2006). Microarray analysis of gene regulation by oxygen, nitrate, nitrite, fnr, narl and narp during anaerobic growth of escherichia coli: new insights into microbial physiology. *Biochem Soc Trans*, 34(Pt 1):104–107.

[Pabo and Sauer, 1992] Pabo, C. O. and Sauer, R. T. (1992). Transcription factors: structural families and principles of DNA recognition. *Annu Rev Biochem*, 61:1053–1095.

[Palaniswamy et al., 2006] Palaniswamy, S. K., James, S., Sun, H., Lamb, R. S., Davuluri, R. V., and Grotewold, E. (2006). AGRIS and AtRegNet. A Platform to Link cis-Regulatory Elements and Transcription Factors into Regulatory Networks. *Plant Physiol.*, 140(3):818–829.

[Pan et al., 1996] Pan, C. Q., Johnson, R. C., and Sigman, D. S. (1996). Identification of new Fis binding sites by DNA scission with Fis-1,10-phenanthroline-copper(I) chimeras. *Biochemistry*, 35(14):4326–4333.

[Paponov et al., 2008] 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. *Mol Plant*, 1(2):321–337.

[Pavesi et al., 2001] Pavesi, G., Mauri, G., and Pesole, G. (2001). An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics*, 17:S207–214.

[Peckham et al., 2007] 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 Res*, 17(8):1170–1177.

[Peng, 2008] Peng, H. (2008). Bioimage informatics: a new area of engineering biology. *Bioinformatics*, 24(17):1827–1836.

[Pernkopf and Bilmes, 2005] Pernkopf, F. and Bilmes, J. A. (2005). Discriminative versus generative parameter and structure learning of Bayesian network classifiers. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 657–664.

[R Development Core Team, 2009] R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

[Raghavan et al., 1989] Raghavan, V. V., Jung, G. S., and Bollmann, P. (1989). A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems*, 7:205–229.

[Redhead and Bailey, 2007] Redhead, E. and Bailey, T. L. (2007). Discriminative motif discovery in DNA and protein sequences using the DEME algorithm. *BMC Bioinformatics*, 8:385.

[Rissanen, 1983] Rissanen, J. (1983). A universal data compression system. *IEEE Trans. Inform. Theory*, 29(5):656–664.

[Ron et al., 1996] Ron, D., Singer, Y., and Tishby, N. (1996). The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25:117–149.

[Roos, 2001] Roos, D. S. (2001). Computational biology. Bioinformatics–trying to swim in a sea of data. *Science*, 291(5507):1260–1261.

[Rowe et al., 2005] Rowe, J. L., Starnes, G. L., and Chivers, P. T. (2005). Complex transcriptional control links NikABCDE-dependent nickel transport with hydrogenase expression in Escherichia coli. *J Bacteriol*, 187(18):6317–6323.

[Saeys et al., 2007] Saeys, Y., Abeel, T., Degroeve, S., and Van de Peer, Y. (2007). Translation initiation site prediction on a genomic scale: beauty in simplicity. *Bioinformatics*, 23(13):i418–423.

[Salzberg, 1997a] Salzberg, S. L. (1997a). A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Comput. Appl. Biosci.*, 13(4):365–376.

[Salzberg, 1997b] Salzberg, S.-L. (1997b). On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1:317–328.

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

[Schneider and Stephens, 1990] Schneider, T. D. and Stephens, R. M. (1990). Sequence logos: A new way to display consensus sequences. *NAR*, 18:6097–6100.

[Schultheiss et al., 2009] Schultheiss, S. J., Busch, W., Lohmann, J. U., Kohlbacher, O., and Ratsch, G. (2009). KIRMES: kernel-based identification of regulatory modules in euchromatic sequences. *Bioinformatics*, 25(16):2126–2133.

[Schweikert et al., 2009] Schweikert, G., Behr, J., Zien, A., Zeller, G., Ong, C. S., Sonnenburg, S., and Rätsch, G. (2009). mGene.web: a web service for accurate computational gene finding. *Nucleic Acids Res*, 37(Web Server issue):W312–W316.

[Segal et al., 2006] 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.

[Sethupathy et al., 2006] Sethupathy, P., Megraw, M., and Hatzigeorgiou, A. G. (2006). A guide through present computational approaches for the identification of mammalian microRNA targets. *Nat Methods*, 3(11):881–886.

[Shannon et al., 2003] Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res*, 13(11):2498–2504.

[Sinha et al., 2009] Sinha, R., Nikolajewa, S., Szafranski, K., Hiller, M., Jahn, N., Huse, K., Platzer, M., and Backofen, R. (2009). Accurate prediction of NAGNAG alternative splicing. *Nucl. Acids Res.*, 37(11):3569–3579.

[Smith et al., 2005] 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.

[Sonnenburg et al., 2007] 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 et al., 2006] Sonnenburg, S., Zien, A., and Rätsch, G. (2006). ARTS: accurate recognition of transcription starts in human. *Bioinformatics*, 22(14):e472–e480.

[Staden, 1984] Staden, R. (1984). Computer methods to locate signals in nucleic acid sequences. *NAR*, 12:505–519.

[Stormo et al., 1982] Stormo, G., Schneider, T., Gold, L., and Ehrenfeucht, A. (1982). Use of the 'perceptron' algorithm to distinguish translational initiation sites. *NAR*, 10:2997–3010.

[Stuart et al., 2003] Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003). A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules. *Science*, 302(5643):249–255.

[Sun et al., 2003] 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. *Proc Natl Acad Sci U S A*, 100(16):9428–9433.

[Swarbreck et al., 2008] 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 Res*, 36(Database issue):D1009–D1014.

[Teale et al., 2006] Teale, W. D., Paponov, I. A., and Palme, K. (2006). Auxin in action: signalling, transport and the control of plant growth and development. *Nat Rev Mol Cell Biol*, 7(11):847–859.

[Thijs et al., 2001] Thijs, G., Lescot, M., Marchal, K., Rombauts, S., De Moor, B., Rouze, P., and Moreau, Y. (2001). A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics*, 17(12):1113–1122.

[Thompson et al., 2003] Thompson, W., Rouchka, E. C., and Lawrence, C. E. (2003). Gibbs recursive sampler: finding transcription factor binding sites. *NAR*, 31(13):3580–3585.

[Thompson et al., 2007] Thompson, W. A., Newberg, L. A., Conlan, S., McCue, L. A., and Lawrence, C. E. (2007). The gibbs centroid sampler. *Nucleic Acids Res*, 35(Web Server issue):W232–W237.

[Tompa et al., 2005] 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., Régnier, 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.

[Ulmasov et al., 1997] Ulmasov, T., Hagen, G., and Guilfoyle, T. J. (1997). ARF1, a transcription factor that binds to auxin response elements. *Science*, 276(5320):1865–1868.

[Ulmasov et al., 1999] Ulmasov, T., Hagen, G., and Guilfoyle, T. J. (1999). Dimerization and DNA binding of auxin response factors. *Plant J*, 19(3):309–319.

[Unden and Bongaerts, 1997] Unden, G. and Bongaerts, J. (1997). Alternative respiratory pathways of Escherichia coli: energetics and transcriptional regulation in response to electron acceptors. *Biochim Biophys Acta*, 1320(3):217–234.

[Wallach, 2002] Wallach, H. (2002). Efficient training of conditional random fields. Master's thesis, University of Edinburgh.

[Wallach, 2004] 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 and Gunsalus, 2003] Wang, H. and Gunsalus, R. P. (2003). Coordinate regulation of the Escherichia coli formate dehydrogenase fdnGHI and fdhF genes in response to nitrate, nitrite, and formate: roles for NarL and NarP. *J Bacteriol*, 185(17):5076–5085.

[Wang et al., 2004] Wang, Z., Rolish, M. E., Yeo, G., Tung, V., Mawson, M., and Burge, C. B. (2004). Systematic identification and analysis of exonic splicing silencers. *Cell*, 119(6):831–845.

[Wettig et al., 2002] Wettig, H., Grünwald, P., Roos, T., Myllymäki, P., and Tirri, H. (2002). On supervised learning of Bayesian network parameters. Technical Report HIIT Technical Report 2002-1, Helsinki Institute for Information Technology HIIT.

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

[Wingender et al., 1996] Wingender, E., Dietze, P., Karas, H., and Knuppel, R. (1996). TRANSFAC: A database on transcription factors and their DNA binding sites. *Nucleic Acids Res*, 24:238–241.

[Wray et al., 2003] Wray, G. A., Hahn, M. W., Abouheif, E., Balhoff, J. P., Pizer, M., Rockman, M. V., and Romano, L. A. (2003). The evolution of transcriptional regulation in eukaryotes. *Mol Biol Evol*, 20(9):1377–1419.

[Wu et al., 2006] Wu, J., Smith, L. T., Plass, C., and Huang, T. H.-M. (2006). ChIP-chip comes of age for genome-wide functional analysis. *Cancer Res*, 66(14):6899–6902.

[Yakhnenko et al., 2005] 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.

[Yang, 2007] Yang, Z. (2007). PAML 4: phylogenetic analysis by maximum likelihood. *Mol Biol Evol*, 24(8):1586–1591.

[Yeo and Burge, 2004] 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. PMID: 15285897.

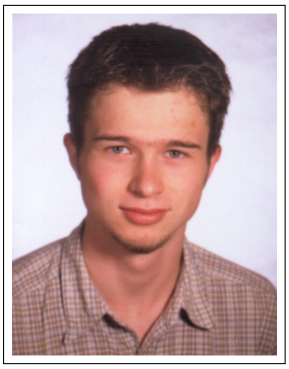
[Zhang and Marr, 1993] Zhang, M. and Marr, T. (1993). A weight array method for splicing signal analysis. *Comput. Appl. Biosci.*, 9(5):499–509.

[Zhao et al., 2005] 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. PMID: 16108724.

[Zhu and Zhang, 1999] Zhu, J. and Zhang, M. (1999). SCPD: a promoter database of the yeast Saccharomyces cerevisiae. *Bioinformatics*, 15(7):607–611.

Curriculum Vitae

Personal Data



Name:	Frank Jens Keilwagen
Academic degree:	Diplom Bioinformatiker
Gender:	male
Date of birth:	2nd of October 1981
Place of birth:	Oschatz
Adress:	Richard-Wagner-Str. 62 38820 Halberstadt
E-Mail:	Jens.Keilwagen@googlemail.com

Education

11/05 - present	Martin Luther University Halle-Wittenberg and IPK Gatersleben Ph.D. studies on bioinformatics supervised by Prof. Dr. Ivo Grosse
10/01 - 10/05	Martin Luther University Halle-Wittenberg Diplom studies on bioinformatics (Grade: 1.0, Georg-Cantor-Preis)
07/00 - 05/01	SKH Hubertusburg civil service at Department of Neurology
01/93 - 06/00	Thomas-Mann-Gymnasium Oschatz (Abitur grade: 1.1, Thomas-Mann-Preis)
09/88 - 12/92	August-Bebel Oberschule Wermsdorf

Research & Work Experience

10/07 - present	Research associate at IPK Gatersleben research group Data Inspection
11/05 - 09/07	Research associate at IPK Gatersleben research group Plant Data Warehouse

List of Publication

Journal papers

[1] Michael Seifert, **Jens Keilwagen**, Marc Strickert, and Ivo Grosse. *Utilizing gene pair orientations for HMM-based analysis of promoter array ChIP-chip data.* Bioinformatics. 2009. 25(16):2118–2125.

[2] **Jens Keilwagen**, Jan Baumbach, Thomas A. Kohl, and Ivo Grosse. *MotifAdjuster: a tool for computational reassessment of transcription factor binding site annotations.* Genome Biology. 2009. 10(5):R46.

[3] **Jens Keilwagen**, Jan Grau, Stefan Posch, and Ivo Grosse. *Apples and oranges: avoiding different priors in Bayesian DNA sequence analysis.* BMC Bioinformatics. 2010. 11(1):149.

[4] **Jens Keilwagen**, Jan Grau, Stefan Posch, Marc Strickert, and Ivo Grosse. *Unifying generative and discriminative learning principles.* BMC Bioinformatics. 2010. 11(1):98.

[5] **Jens Keilwagen**, Jan Grau, Ivan A. Paponov, Stefan Posch, Marc Strickert, and Ivo Grosse. *De-novo discovery of differentially abundant transcription factor binding sites including their positional preference.* PLoS Computational Biology. 2010. submitted.

Conference papers

[1] Jan Grau, **Jens Keilwagen**, Ivo Grosse, and Stefan Posch. *On the relevance of model orders to discriminative learning of Markov models.* LWA: Lernen – Wissen – Adaption, Editor: A. Hinneburg. 2007. 61–66.

[2] **Jens Keilwagen**, Jan Grau, Stefan Posch, and Ivo Grosse. *Recognition of splice sites using maximum conditional likelihood.* LWA: Lernen - Wissen - Adaption, Editor: A. Hinneburg. 2007. 67–72.

[3] Jan Grau, **Jens Keilwagen**, Alexander Kel, Ivo Grosse, and Stefan Posch. *Supervised posteriors for DNA-motif classification.* German Conference on Bioinformatics, Lecture Notes in Informatics (LNI) - Proceedings, Editor: C. Falter, A. Schliep, J. Selbig, M. Vingron, D. Walter. 2007. 115:123–134.

[4] Michael Seifert, **Jens Keilwagen**, Marc Strickert, and Ivo Grosse. *Utilizing promoter pair orientations for HMM-based analysis of ChIP-chip data.* German Conference on Bioinformatics, Lecture Notes in Informatics (LNI) - Proceedings, Editor: A. Beyer and M. Schroeder. 2008. 136:116–127.

[5] Marc Strickert, Petra Schneider, **Jens Keilwagen**, Thomas Villmann, Michael Biehl, and Barbara Hammer. *Discriminatory data mapping by matrix-based supervised learning metrics.* Lecture Notes in Computer Science, Editor: L. Prevost and S. Marinai and F. Schwenker. 2008. 5065:78–89.

[6] Marc Strickert, Katja Witzel, **Jens Keilwagen**, Hans-Peter Mock, Petra Schneider, and Michael Biehl. *Adaptive matrix metrics for attribute dependence analysis in differential high-throughput data.* Proceedings of the fifth international Workshop on Computational Systems Biology (WCSB), TICSP series, Editor: M. Ahdesmäki and K. Strimmer and N. Radde and J. Rahnenführer and K. Klemm and H. Lähdesmäki and O. Yli-Harja. 2008. 41:181–184.

[7] Michael Seifert, and Ali M. Banaei, **Jens Keilwagen**, Michael F. Mette, Andreas Houben, Françios Roudier, Vincent Colot, Ivo Grosse, and Marc Strickert. *Array-based Genome Comparison of Arabidopsis Ecotypes Using Hidden Markov Models.* Proceedings of the 2nd International Conference on Bio-inspired Systems and Signal Processing (Biosignals 2009). 2009. 3–11.

[8] Marc Strickert, **Jens Keilwagen**, Frank-Michael Schleif, Thomas Villmann, and Michael Biehl. *Matrix Metric Adaptation Linear Discriminant Analysis of Biomedical Data.* Bio-Inspired Systems: Computational and Ambient Intelligence, Lecture Notes in Computer Science, Editor: J. Cabestany and F. Sandoval and A. Prieto and J.M. Corchado. 2009. 5517:933–940.

[9] Marc Strickert, Axel J. Soto, **Jens Keilwagen**, and Gustavo E. Vazquez. *Towards matrix-based selection of feature pairs for efficient ADMET prediction.* Proceedings of the 9th Argentine Symposium on Artificial Intelligence (ASAI 2009). 2009. 83–94.

Book chapters

[1] Stefan Posch, Jan Grau, André Gohr, **Jens Keilwagen**, and Ivo Grosse. *Probabilistic Approaches to Transcription Factor Binding Site Prediction.* Methods in Molecular Biology, Humana Press, Editor: S. Ladunga. 2010. Editorially accepted.

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 und Hilfsmittel benutzt. Die den benutzten Werken anderer Autoren wörtlich oder inhaltlich entnommenen Stellen sind als solche kenntlich gemacht worden. Bisher habe ich mich noch nicht um einen Doktorgrad beworben.

Halle (Saale), 01. April 2010 Jens Keilwagen